Pushing the Limits of Wireless Networks

by

Swarun Kumar

M.S., Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 2012
B.Tech., Computer Science, Indian Institute of Technology, Madras, 2010

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

Massachusetts Institute of Technology

February 2016

Author . . . .
Department of Electrical Engineering and Computer Science
December 21, 2015

Certified by . . . .
Dina Katabi
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Students

Pushing the Limits of Wireless Networks

by

Swarun Kumar

Submitted to the Department of Electrical Engineering and Computer Science
on December 21, 2015, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

Wireless networks are everywhere around us and form a big part of our day-to-day lives. In this dissertation, we address the key challenges and opportunities of modern wireless networks. First, perhaps our biggest expectation from modern wireless networks is faster communication speeds. However, state-of-the-art Wi-Fi networks continue to struggle in crowded environments – airports and hotel lobbies. The core reason is interference – Wi-Fi access points today avoid transmitting at the same time on the same frequency, since they would otherwise interfere with each other. This thesis describes OpenRF, a novel system that enables today's Wi-Fi access points to directly combat this interference and demonstrate significantly faster data-rates for real applications. In addition, it presents MoMIMO, which demonstrates how the natural mobility of mobile users can be used to further mitigate interference.

Second, can we use the ubiquitous Wi-Fi infrastructure around us to deliver new services, beyond communication? In particular, this dissertation focuses on indoor positioning, a service that has grabbed the attention of the academia and industry. While GPS has revolutionized outdoor navigation, it does not work indoors. Past work that has explored this problem is either limited in accuracy with errors of several meters, or advocates complete overhaul of the infrastructure with massive antenna-array access points that do not exist on consumer devices. Inspired by radar systems, we present Ubicarse, the first purely-software indoor positioning system for existing Wi-Fi devices that achieves tens of cm in positioning accuracy. Further, we build on this design to develop LTEye, which reveals new insights on how location impacts the performance of commercial AT&T and Verizon LTE cellular networks in the indoor space. Finally, we demonstrate how the tools we develop for indoor positioning open up new connections between wireless networking and robotics, to improve communication and security in multi-robot networks.

Thesis Supervisor: Dina Katabi
Title: Professor of Computer Science and Engineering

To my parents

Acknowledgments

My graduate career at MIT has been an exciting and enriching phase of my life. I am greatly indebted to my advisor, Dina Katabi, who has been amazingly supportive and inspirational. Dina works incredibly hard for her students and is passionate about research. Dina always has the best interests of her students in mind, and is a constant source of encouragement and motivation. I have truly enjoyed the brainstorming sessions through my Ph.D with Dina, which have always led to new and exciting research ideas. I am fortunate to have had such a wonderful mentor guiding me through my graduate career. She remains my role model as I embark on my career as faculty.

I have also had an amazing experience working with Daniela Rus, closely throughout my Ph.D. It was a truly enjoyable experience interacting with Daniela and the DRL group and learning about the world of robotics. I am very grateful for her support during my academic job search and for her valuable feedback on my job talk. I am thankful to Li Erran Li who exposed me to cellular networking. I had a great experience working with him on multiple projects in LTE networking. I thank my thesis committee: Daniela Rus, Li Erran Li and Venkat Padmanabhan, from whom I have learned a great deal. They have been immensely helpful in shaping this thesis.

This thesis builds on several papers [87, 10, 88, 89, 58, 59], which would have not been possible without an amazing team of student collaborators. I have worked with Stephanie Gil on a number of projects in robotics and beyond through my Ph.D career – the topics ranging from autonomous vehicles, multi-robot systems, security and indoor positioning (Ubicarse). It has been an incredibly fun experience working with Steph, who I have learned so much from. Diego Cifuentes was pivotal in spending many sleepless nights implementing the OpenRF project. I recall how Ezz Hamed and I spent many snowy days in the winter outdoors to get the LTEye project ready in time for SIGCOMM. I had an enjoyable time working with Fadel Adib and Omid Aryan on the MoMIMO project. I had great fun flying quadrotors with Mark Mazumder during our cybersecurity project for RSS 2015. I am grateful to Shyam Gollakota who I worked closely with on multiple projects in this thesis: OpenRF and MoMIMO.

I would like to thank my amazing collaborators on many other projects that I have worked on that are not a part of this thesis: Deepak Vasisht, Hariharan Rahul, Lixin Shi, Nabeel Ahmed, Xiuefeng Xie and Xinyu Zhang. I am indebted to Hariharan Rahul, who taught me much about wireless networks on my very first project at MIT: MegaMIMO. I had an exciting time working with Lixin Shi and Nabeel Ahmed on CarSpeak, my Master's thesis project. It was a pleasure working with Deepak Vasisht, who worked incredibly hard on the Chronos project to estimate accurate time-of-flight on commodity Wi-Fi cards. I am fortunate to have collaborated with amazing researchers at the University of Wisconsin Madison: Xiufeng Xie and Prof. Xinyu Zhang on the piStream project to improve video streaming over LTE.

I am grateful to past and present members of the NETMIT group who I have journeyed with through my Ph.D: Hariharan Rahul, Nate Kushman, Szymon Jakubczak, Shyam Gollakota, Jue Wang, Sam Perli, Haitham Hassanieh, Fadel Adib, Lixin Shi, Diego Cifuentes, Nabeel Ahmed, Omid Abari, Badih Ghazi, Ezzeldin Hamed, Zach Kabelac, Deepak Vasisht and Chen-Yu Hsu. I thank Arthur Berger who has provided invaluable feedback on all my papers. I am grateful to our group's admin assistant Mary McDavitt who has been an incredible source of help and support.

MIT is an incredible place with the most talented researchers in any given field. Over my Ph.D, I have had the opportunity to learn from experts in a wide range of fields including networking, theory, graphics, mechanical engineering, astronomy, aeronautics, business and finance, just to name a few. I would like to thank all the professors, students and researchers I have interacted with during my courses and research here. I am also thankful to all my professors and friends at my undergraduate university, IIT Madras, who laid the foundation for my interest in graduate research.

Beyond the lab, I am grateful to all my friends at MIT and the greater Boston area. I thoroughly enjoyed the refreshing company of Anirudh, Sivaraman, Aravind and Siddharth who have been amazing friends. I would also like to thank my relatives in different parts of the U.S. and India who have been very supportive and have always kept in touch. I have thoroughly enjoyed visiting them through the years.

Finally, I am immeasurably indebted to my parents. They have always supported me unconditionally and have guided me through every step of my life. This thesis is dedicated to them.

Previously Published Material

Chapter 3 revises a previous publication [87]: Bringing Cross-Layer MIMO to Today's Wireless LANs, Swarun Kumar, Diego Cifuentes, Shyamnath Gollakota, and Dina Katabi, *Proc. ACM SIGCOMM 2013*, Hong Kong

Chapter 4 revises a previous publication [10]: Interference Alignment by Motion, Fadel Adib (Co-primary), Swarun Kumar (Co-primary), Omid Aryan, Shyamnath Gollakota, and Dina Katabi, *Proc. ACM MOBICOM 2013*, Miami FL

Chapter 6 revises a previous publication [89]: LTE Radio Analytics Made Easy and Accessible, Swarun Kumar, Ezzeldin Hamed, Dina Katabi, and Li Erran Li, *Proc. ACM SIGCOMM 2014*, Chicago IL

Chapter 7 revises a previous publication [88]: Accurate Indoor Localization with Zero Startup Cost, Swarun Kumar, Stephanie Gil, Dina Katabi and Daniela Rus, *Proc. ACM MOBICOM 2014*, Maui HI

Chapter 9 revises a previous publication [58]: Adaptive Communication in Multi-Robot Systems Using Directionality of Signal Strength, Stephanie Gil, Swarun Kumar, Dina Katabi and Daniela Rus, *IJRR 2014*

Chapter 10 revises a previous publication [59]: Guaranteeing Spoof-Resilient Multi-Robot Networks, Stephanie Gil (Co-Primary), Swarun Kumar (Co-Primary), Mark Mazumder, Dina Katabi and Daniela Rus *Proc. RSS 2015*, Rome Italy

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Wireless networks form an important role in our everyday lives. Billions of users worldwide today rely on wireless networks, be it on their laptops, smartphones, wearables or tablets. As a result, pushing the limits of wireless networks will benefit the host of applications that we enjoy today and impact our day to day lives.

Yet, there remain key challenges that limit modern wireless networks. We can summarize these along two broad axes:

- **Combating Interference:** The first axis is a classic problem of every wireless network: wireless interference. Interference is at the core of wireless problems that everyday users face, particularly in crowded hotels and airports. In particular, wireless spectrum is a shared resource. As a result, if wireless nodes transmit at the same time and frequency, their signals interfere. To mitigate this, today's wireless networks adopt a simple strategy – they force users to take turns on the medium, so that each user in the network receives their packet, one at a time [2]. Consequently, users experience increasingly lower speeds, as more and more users join the network [141].
- **New Services:** The second axis is both a challenge and an opportunity – Using the wireless infrastructure around us not just for communication, but also to provide new services. Among these services, a key service that has grabbed the attention of both the academia and industry alike is indoor positioning. While GPS has revolutionized outdoor navigation, it does not work indoors [142]. As a result, there is a massive interest in using wireless technologies like Wi-Fi [188, 82, 142] and cellu-

> lar [5, 109] systems for positioning in the indoor space. Such a service opens up a range of applications: For instance, a user can find her way around in a mall or museum using a mobile phone, just as she can outdoors with GPS today. Businesses like Wal-Mart and retail stores can use it to flash the most relevant offers to customers as they walk along different aisles. Finally, teams of robots can find their way around to pick up different products in a warehouse.

Unfortunately, past work that has that looked into these challenges suffers from one of two major shortcomings: (1) either, it requires complete overhaul of Wi-Fi infrastructure, i.e. replacing all Wi-Fi access points with new hardware that does not even exist on the market; (2) or, it does not provide satisfying performance. For instance, state-of-the-art systems that combat wireless interference [141, 63, 62] require developing new Wi-Fi hardware following new standards that do not exist today. Unfortunately, it will take decades before these standards can be developed and make it to real networks. Consequently, current networks simply force users to take turns on the wireless medium [2].

Indeed, the dilemma remains the same for indoor positioning. State-of-the-art systems require large multi-antenna arrays that are extremely bulky [188, 82]: a row of sixteen antenna elements, each separated by tens of centimeters. Such solutions simply do not exist on consumer devices – both access points, and mobile devices. Yet, alternative solutions that work with today's network infrastructure fail to deliver adequate performance [142, 168, 104]. These require the cumbersome process of collecting signal fingerprints, i.e. exhaustively measuring wireless signals in every room and every floor of the deployment space. To make matters worse, these solutions begin to unravel if the environment changes in any way. For instance, if people move or even if furniture is relocated, they quickly loose their accuracy [111]. As a result, such solutions fail to work reliably in practice.

Ideally, we would like to address both these challenges: combating interference and indoor positioning, while achieving high performance without overhauling our infrastructure. This dissertation addresses this precise need. It presents multiple systems that are designed either as purely-software modifications or simple low-cost hardware that can be readily plugged in to existing wireless infrastructure. Underlying these systems are novel algorithms that can both *measure* and *control* how wireless signals interact with the environment, using simple commodity wireless hardware. Specifically, wireless signals emitted by a transmitter bounce off walls and obstacles spawning multiple copies of the signal. These copies combine at the receiver to either reinforce or cancel each other. This thesis presents novel systems that intelligently control signals at a commodity Wi-Fi transmitter, so that all copies of interfering signals effectively cancel out each other. Further, it demonstrates how measuring and separating these signal copies at the receiver can help identify and track the location of the transmitting device (e.g. a mobile phone). Finally, the ability to measure and control wireless signals opens up new connections between networking and the field of robotics – where wireless communication is innate. Specifically, we discover new ways in which cheap, commodity Wi-Fi radios on robots can emulate virtual "sensors" to enable better navigation, communication and security in multi-robot systems.

The rest of this dissertation describes systems that control and measure wireless signals on commercial wireless hardware to advance three broad goals: 1) Combating wireless interference (Chapters 2-4); 2) Achieving accurate indoor positioning (Chapters 5-7); (3) Opening up new connections between wireless networking and robotics (Chapters 8-10). The following sections briefly introduce these systems and present their core contributions.

■ 1.1 Combating Interference

The first part of this dissertation deals with combating wireless interference in today's wireless LANs. Our work builds on MIMO interference management techniques [63, 99, 141] that rely on multiple antennas on Wi-Fi access points to cancel interference at clients. Unfortunately, prior to this dissertation, deploying these techniques would require complete overhaul of today's Wi-Fi infrastructure: both new hardware and standards that take decades to develop.

In contrast, our work presents novel systems that require neither. At their heart are novel mechanisms that can intelligently control how wireless signals combine over the air at the transmitters, so that any interfering signals cancel out at wireless receivers. These mechanisms are designed to be compatible within the constraints of existing Wi-Fi hardware and standards. We first present a software-only end-to-end architecture to cancel interference on today's multi-antenna Wi-Fi access points running commercial Wi-Fi radios. We further demonstrate that we can combat interference, even for single-antenna devices, by exploiting their natural mobility. Our solutions have been implemented and evaluated in wireless testbeds to demonstrate significant gains in performance compared to state-of-the-art.

Contributions. Our work makes the following key contributions:

- *MIMO interference management on today's wireless LANs:* OpenRF [SIGCOMM '13], the first purely software system to bring MIMO interference cancellation techniques to commercial Wi-Fi devices. At its core are new algorithms that manipulate wireless signals at Wi-Fi transmitters to cancel interference at receivers, within the constraints of today's Wi-Fi hardware. OpenRF demonstrated that such techniques bring massive gains to today's mobile applications for the first time. Fig. 1-1 shows OpenRF in action, where videos sent using OpenRF experience $4\times$ fewer glitches and delays, compared to 802.11n Wi-Fi.

- *Interference alignment by motion:* MoMIMO [MOBICOM '13] brings MIMO interference management (e.g., interference alignment and nulling), hitherto restricted to multi-antenna devices [63, 99], for the first time to single-antenna devices. At its heart is a novel technique that demonstrates how a movement of just an inch or less of a wireless device can achieve interference alignment to dramatically reduce interference. MoMIMO develops new wireless models that understand how signals add up differently across spatial locations. Using these models, MoMIMO shows that by intelligently moving a wireless antenna, one can seek local positions where interference adds up to zero. Such antennas can easily be connected to the external ports of today's Wi-Fi access points. Further, one can leverage the natural mobility of handheld devices themselves, advising users on how best to move them, when they experience interference. MoMIMO was implemented on software radios as well as commodity Wi-Fi devices and demonstrated $1.98\times$ gain in throughput over 802.11n Wi-Fi.

■ 1.2 Indoor Positioning

The second part of this dissertation presents new ways to achieve accurate indoor positioning using Wi-Fi and cellular signals from today's mobile devices. Specifically, it addresses the main culprit that impedes positioning in the indoor space, as opposed to outdoors: the

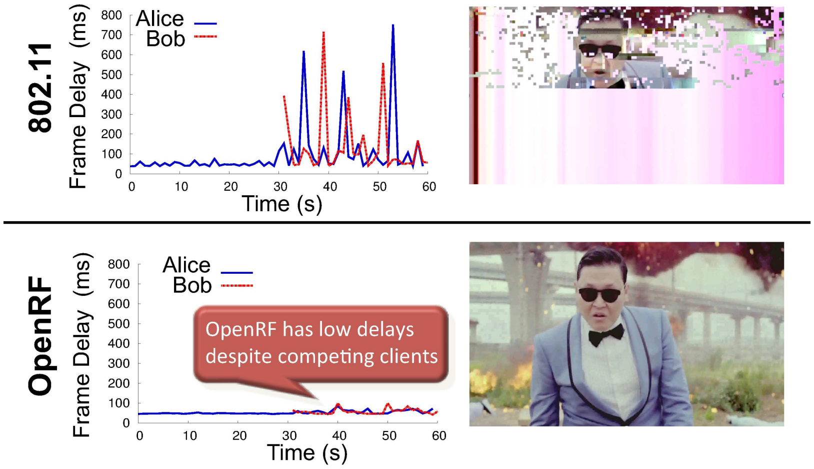


Figure 1-1: OpenRF. OpenRF clients see fewer glitches in videos compared to 802.11 (video screenshots, right). It has shorter frame delays over time (left) even with competing clients under identical conditions and hardware.

fact that wireless signals bounce off walls and objects causing an effect called multipath. Multipath creates ambiguity at the receiver on where the signal originated from, hence hampering its ability to localize the transmitter. Past solutions attempted to resolve this issue in two ways: 1) elaborately fingerprinting the environment to create a database of how the signal may look like for each possible location of the transmitter [142, 104], or 2) expensive infrastructure unavailable on cellphones, like large arrays of antennas [188, 82].

In contrast, this dissertation presents new solutions that require neither (see Fig. 1-2). At the heart of our approach is a novel method to combat multipath by measuring wireless signals vary as a receiving device moves. Specifically, as the receiving device moves, it records different combinations of the wireless signals (i.e., different multipath). Using these different observations, one can invert the effect of multipath and identify the true direction of the source. This approach allows us to transform our commodity wireless devices (e.g. Wi-Fi tablets or access points) into personal radars that perform indoor positioning to within 30 cm.

Specifically, this dissertation presents two systems that tackle the indoor positioning problem from two perspectives: from that of the wireless infrastructure and user device. The first system proposes simple hardware modifications that can be plugged in to existing wireless infrastructure (e.g. Wi-Fi access points or LTE femtocells) to achieve accurate indoor positioning. Our approach requires no modification to user devices and can track them continuously as they move through the deployment space. The second is a software-only localization system that runs on a user device connected to existing unmodified Wi-Fi infrastructure. This system provides an on-demand localization service that a user can invoke, whenever they twist their device. Experiments reveal tens of centimeters accuracy for both systems in indoor environments, without requiring specialized infrastructure or elaborate fingerprinting prior to deployment.

Contributions. Our work delivers two key contributions:

- *LTE radio analytics made easy and accessible:* LTEye is an indoor positioning system that achieves tens of centimeters accuracy in positioning without requiring exhaustive fingerprinting of indoor environments. LTEye is deployed using simple hardware modifications on wireless infrastructure (e.g. Wi-Fi access points or LTE femtocells). In particular, LTEye was deployed on commercial LTE networks, whose performance and interference patterns in the indoor space remains less understood, even by cellular operators given the advent of small cells that users themselves can deploy. Indoor positioning can play a crucial role given that it can link mobile user locations with their cellular performance. By detecting user performance over space, one can quickly identify problems and adopt remedies for impacted users. LTEye is the first open-source system to throw light into the LTE radio layer, without operator support. LTEye contains new mechanisms to infer user performance over space (see Fig. 4), without access to their encrypted data. LTEye was built and deployed on the MIT campus and found serious deficiencies in production AT&T and Verizon networks, including high inter-cell interference and inefficient usage of expensive licensed spectrum.

- *Accurate Wi-Fi indoor positioning with zero startup cost:* Ubicarse[1] is an accurate indoor localization system that runs in software on commodity Wi-Fi devices using existing Wi-Fi infrastructure. Ubicarse is inspired by a radar technique called Synthetic Aperture Radar (SAR). Traditional SAR systems mount receiving antennas on aerial or ground vehicles to resolve multipath and geo-locate transmitters highly accurately. In contrast, Ubicarse transform a handheld device to a personal radar to localize Wi-Fi signals. Indeed, doing so is quite natural; users can readily move their

[1] The Spanish word Ubicarse derives from the Latin root *ubi* and signifies to locate or situate oneself with precision.

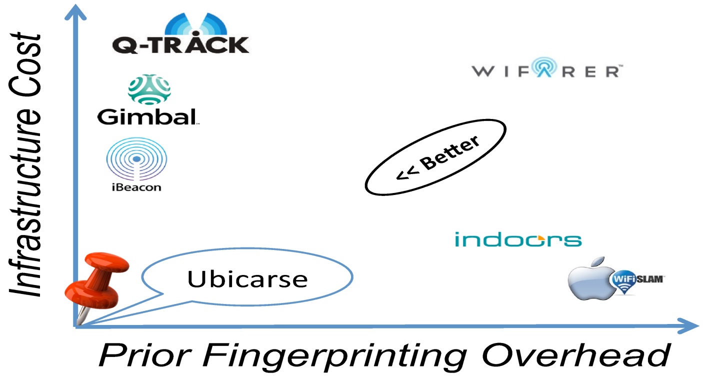


Figure 1-2: Ubicarse vs. Current Indoor Positioning Systems. Current indoor positioning systems (logos) need new infrastructure or fingerprinting. Ubicarse needs neither

devices, much like how vehicles move radars in SAR. However, bringing SAR from complex radars to handheld devices is not simple. SAR vehicles move antennas along highly controlled trajectories like perfect lines or circles. It is impossible to ask users to move their devices with such precision. Ubicarse's core contribution is a new formulation of SAR that localizes handheld devices even as users move their mobile devices along unknown paths. Ubicarse is not limited to localizing RF devices; it combines RF localization with stereo-vision algorithms to localize common objects with no RF source attached to them. We implement Ubicarse on a HP SplitX2 tablet and empirically demonstrate a median error of 39 cm in 3-D device localization and 17 cm in object geotagging in complex indoor settings.

1.3 New connections between wireless and robotics

The final part of this dissertation explores new connections between wireless communication and robotics. Wireless networks and robotics have a long history. Any multi-robot system requires wireless networks to communicate, be it for manufacturing, agriculture, mining or search-and-rescue. Yet, wireless communication is traditionally viewed as a blackbox in robotics, simply to exchange messages.

This dissertation reveals the benefits of cracking the blackbox open – it addresses the important problems pertaining to navigation and security in robotics, simply using off-the-shelf Wi-Fi radios as sensors. Specifically, we present algorithms that actively measure how wireless signals from a transmitter reflect around obstacles before they reach the receiver.

These algorithms form the basis for two novel systems. First, we make the observation that robots can learn how to avoid obstacles by sensing how radio waves reflect around them. This enables a novel system where robots that can navigate to improve signal quality by following the wireless signals emitted by their targets, akin to how moths navigate towards a lamp, following visible light. Second, we observe that signals from a wireless transmitter carry unique information that is dependent both on the transmitter's hardware, its location and its environment (e.g. the location of obstacles around it). As a result, they can be used to extract fingerprints that remain unique to specific transmitters. We demonstrate how this primitive enables a natural authentication mechanism for robots to secure against the Sybil attack, where a malicious robot can disrupt a multi-robot network by pretending to be a large number of fake clients.

Contributions. We present the following contributions to advance network throughput and security in multi-robot systems:

- *Multi-Robot Communication:* Multi-robot swarms are teams of robots collaborating on a common objective, e.g. to survey the most area after an earthquake, or score maximum goals in RoboCup soccer. While doing so, robots must maintain connectivity to each other. Traditionally, robotic networks used simplistic models to do this. For instance, the popular disk model [33, 80] assumes that two robots will always be connected if they are within a specific distance. This model is known to be impractical, both within the networking and robotics communities [113]. However, robotic systems retain the model since it involves simple constraints that maintain the closed-form solutions of robotic optimizations. In other words, past work trades-off performance for simplicity. In contrast, we develop a new approach achieves the best of both worlds. At its heart are new data-driven models based on the physics of wireless signals. These models calculate a mapping between a robot's current position and the signal strength that it receives along *each spatial direction*, for its wireless links to every other robot. We show that this information can be used to design a positional controller that adapts to wireless signals in real-world environments, yet retains a simple mathematical structure. Experiments show that our method outperforms state-of-the-art approaches [33, 80, 95, 159] both in convergence time (3.4× faster) and variability of performance (4× smaller variance).

- *Multi-Robot Security:* Wireless communication enables wide-ranging services in multi-robot tasks such as aerial surveillance and unmanned delivery. However, effective coordination between multiple robots requires trust, making them particularly vulnerable to cyber-attacks. Specifically, such networks can be gravely disrupted by the Sybil attack, where even a single malicious robot can spoof a large number of fake clients. We propose a new solution to defend against the Sybil attack, without requiring expensive cryptographic key-distribution. Our core contribution is a novel algorithm implemented on commercial Wi-Fi radios that can "sense" spoofers using the physics of wireless signals. We derive theoretical guarantees on how this algorithm bounds the impact of the Sybil Attack on a broad class of multi-robot coordination problems. We experimentally validate our claims using a team of AscTec quadrotor server robots and iRobot Create ground clients, and demonstrate spoofer detection rates over 96%.

■ 1.4 Organization

The rest of this dissertation is organized as follows. Chapter 2 presents an overview of our systems to combat interference. In Chapter 3, we describe OpenRF in greater detail and how it brings multi-antenna interference management to commodity Wi-Fi radios. The chapter presents the first study of the benefits of these techniques on today's network protocols and real applications. Chapter 4 describes how MoMIMO enables interference management techniques like interference alignment and nulling by motion. In doing so, it brings multi-antenna interference management, for the first time, to single-antenna devices.

Next, Chapter 5 is an overview of our systems to achieve accurate indoor positioning. Chapter 6 describes LTEye, the first open platform to monitor and analyze LTE radio performance at a fine temporal and spatial granularity. The chapter presents empirical insights and analytics from commercial AT&T and Verizon base stations. Chapter 7 presents Ubicarse and describes how it achieves indoor localization with tens of centimeters of accuracy on commodity mobile devices, with no specialized infrastructure or fingerprinting.

Chapter 8 is an overview of systems that open up new connections between wireless networking and robotics. In Chapter 9, we describe a simple and efficient approach for robots in a multi-robot Wi-Fi network to distribute themselves to maintain high communication quality. In Chapter 10, we describe our system to secure multi-robot networks against the Sybil attack, and provide analytical bounds on the effect of adversaries, without requiring expensive cryptographic protocols. We perform a detailed experimental evaluation on a swarm of aerial and ground robots, equipped with off-the-shelf Wi-Fi radios.

CHAPTER 2
Combating interference

It is well known that wireless spectrum is a limited and valuable resource. In the last U.S. spectrum auction, a mere 10 MHz of spectrum was sold for $6 Billion. The main reason for this is interference. Specifically, multiple users cannot transmit concurrently on the same spectrum because their signals will interfere and receivers cannot decode them. Hence, users need to share spectrum between them, making it a scarce resource. Interference is therefore the fundamental challenge of wireless networking. Traditionally, the wireless community designed protocols that avoid interference. They made users take turns in the air, slowing traffic to send only one packet at any time. Only recently did the community realize how a better understanding of wireless signals can limit interference, without losing network speeds. Unfortunately, in doing so, they require a complete overhaul of wireless infrastructure – both new hardware and new Wi-Fi standards.

This dissertation elevates this approach by developing new mechanisms that actively control how signal copies combine in the air to eliminate interference altogether. These mechanisms are designed to be implemented on commodity Wi-Fi radios employing existing standards, either purely in software or with simple hardware modifications.

We begin with a system that can be integrated purely in software to combat interference from multi-antenna access points. This system actively controls interfering signals transmitted by each antenna of commercial Wi-Fi radios in access points to cancel out at receiver. Further, we present a system that can cancel interference even from single-antenna devices. Our key idea is to exploit mobility of wireless devices to control the signals they emit. We demonstrate how moving these devices just by an inch or less can dramatically reduce the interference they cause at wireless receivers. The following sections elaborate the key challenges that each of these systems address, their high-level ideas and relation to prior work.

■ 2.1 Bringing Interference Management to Today's WLANs

Despite demonstrating significant throughput gains, multi-antenna interference management has not made it into real networks. Deploying these innovations requires adoption from Wi-Fi chip manufacturers. Yet, manufacturers hesitate to undertake major investments without a better understanding of how these designs interact with real networks and applications.

In chapter 3, we present OpenRF [87] that aims to breaks this stalemate: It proposes the first system to demonstrate multi-antenna interference management in today's networks with commodity Wi-Fi cards and actual applications. OpenRF enables access points to cancel their interference at each other's clients, while beamforming their signal to their own clients. OpenRF draws from software defined networking [116, 23] to separate PHY-layer actions at the data plane from quality-of-service requirements at the control plane. Such a design enables the key benefit of self-configurability: network administrators can deploy it without needing to understand MIMO or physical layer techniques. OpenRF's preliminary results demonstrate significant performance gains in TCP throughput as well as reduced delays and stalls for higher-layer video applications.

OpenRF's design. OpenRF's key contribution is a self-configuring design to enable multi-antenna (MIMO) interference management on commodity access points. Architecturally, OpenRF borrows from a software-defined networking design of wired networks (e.g. OpenFlow [116]), in that it separates the control plane from the data plane and exposes functions that have traditionally been deeply hidden in the network stack, to higher layers. As illustrated in Fig. 2-1, the data plane is controlled by the *OpenRF interface*, which resides on access points and shields other components from how signal processing techniques are implemented at the device level. Analogous to the OpenFlow interface [116], the OpenRF interface operates over a table of (FlowID, Actions) tuples. In contrast to OpenFlow entries, where an action may identify which port to transmit the flow on, here, an action

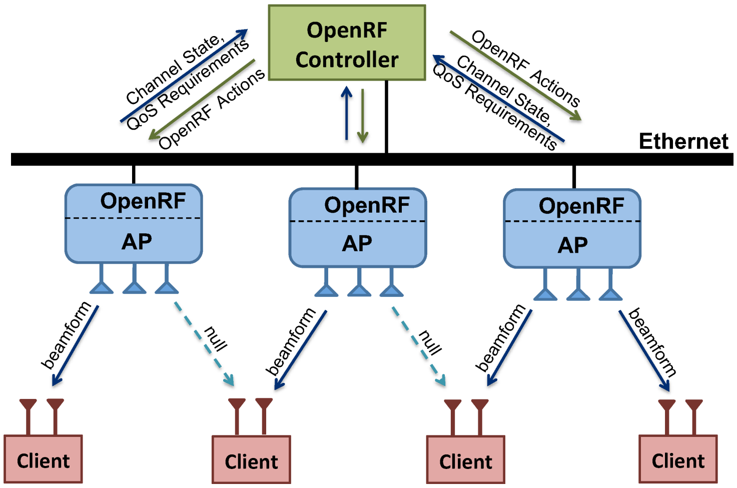


Figure 2-1: Architecture of OpenRF. OpenRF provides an interface to physical-layer MIMO signal processing, e.g., interference nulling, interference alignment, and beamforming. The OpenRF Controller coordinates network devices through this interface.

specifies the relative power used to transmit the flow on each of the access point's antennas. This is typically referred to in MIMO terminology as the pre-coding vector of the flow. Just as forwarding steers a flow's packets toward a particular route in the wired network, pre-coding steers the wireless signal and creates a beam that propagates along a particular spatial direction, allowing the system to null interference at an unwanted receiver and focus the power on the desired receiver.

The control plane in our design is managed by the *OpenRF Controller.* It is configured with per-flow quality of service requirements (e.g., a desired rate), as well as which PHY actions are supported by the OpenRF interface. It also periodically takes as input channel measurements from the access points. Using this information, the controller maps the quality-of-service requirements into signal-layer actions like interference nulling [63], beamforming [166], or alignment [99]. The controller then fills up the OpenRF table with these actions mapped to various flows, so that the needs of each flow can be satisfied.

Chapter 3 delves into this design in greater detail. Further, it describes how OpenRF ensures that the whole network stack operates reliably, while performing MIMO interference management. Specifically, it details how OpenRF accounts for TCP and application burstiness and the resulting dynamism in the interference patterns. Finally, it implements OpenRF by modifying the iwlwifi driver for the Intel 5300 Wi-Fi card, and demonstrates performance gains over 802.11n Wi-Fi [2] in a 20-node testbed.

■ 2.2 Interference Management by Motion

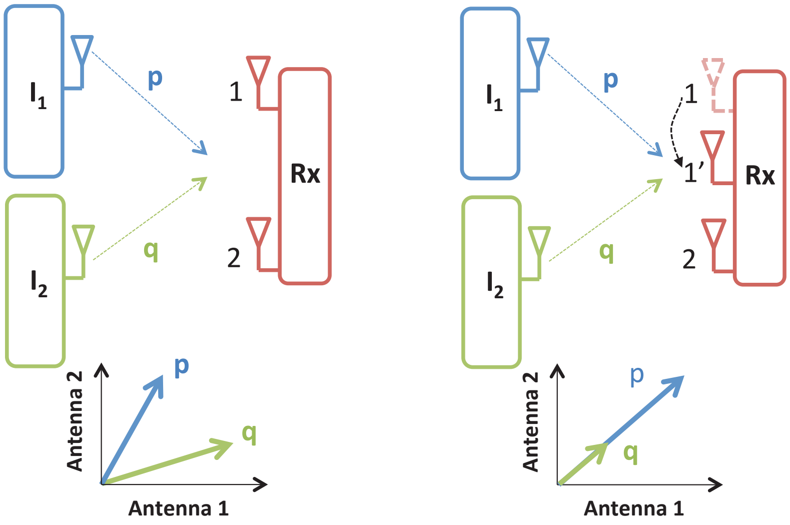
Traditional MIMO interference management techniques leverage multiple antennas on wireless devices to improve throughput. Unfortunately, several Wi-Fi devices: wearables, webcams, sensors, etc., have single-antennas, and are therefore left out from these benefits.

In chapter 4, we present MoMIMO [10] that aims to bring these MIMO benefits to single-antenna devices. Specifically, MoMIMO shows that MIMO techniques can be applied simply by moving a single antenna by about an inch. In other words, a device can cancel unwanted signals from an interferer just by intelligently sliding its receive antenna. Our approach recognizes that since indoor settings are rich in multipath, even a small perturbation in a device's location, can significantly alter the received wireless channel. Its key contribution is a novel algorithm that studies how the wireless channel varies as the antenna moves, to achieve interference management goals. Preliminary results demonstrate that our approach provides a 2× improvement in throughput over traditional Wi-Fi for both single and multi-antenna devices.

MoMIMO's key idea . We investigate whether a receiver can perform interference alignment, a MIMO interference management technique, by simply adjusting the position of one of its antennas. Our intuition is that this may be possible due to two reasons. First, indoor settings are rich in multipath. Hence, at any point in space, the perceived channel is a combination of many channels that correspond to the various paths that the signal traverses. Even a small shift in the antenna location would cause some of these paths to become shorter and others longer, leading to a significant change in the resulting channel. Second, channels are continuous functions over space.[1] This means that the receiver can gradually slide its antenna, measure the channels, and keep moving the antenna in the direction that increases the alignment between the senders.

To test this intuition, we used USRP radios to experiment with the scenario in Fig. 2-2. In this setup, we have two single-antenna interferers and a two-antenna receiver. We mount one of the receive antennas on an iRobot Create robot to emulate a sliding antenna. We measure the channels and gradually adjust the position of one of the receive antennas,

[1] The channel is a linear combination of multiple continuous waveforms [172], and, hence, is continuous over space and time.



(a) **Before Sliding** (b) **After Sliding**

Figure 2-2: Interference Alignment by motion. Consider two single-antenna interferers I_1 and I_2 and a two-antenna receiver Rx. Initially, the two interferers are not aligned in the antenna space of Rx. Then, Rx slides one of its receive antennas so that I_1 and I_2 are aligned in the antenna space.

moving it in the spatial direction that increases the alignment between the interferers, as in Fig. 2-2(b). The experiments have confirmed our intuition: alignment can indeed be performed by sliding the receiver's antenna. What is perhaps even more surprising is that the required antenna displacement was typically an inch or less. Multiple experiments in two campus buildings in both line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios showed that in 84% of the settings, the required antenna displacement was less than one inch. The implications of our results are two-fold: First, they show for the first time that interference alignment can be achieved via motion. Second, the fact that the required displacement is of the order of one inch, means that motion-based interference alignment can be readily incorporated into existing wireless devices outfitted with the recent sliding antennas available on the market.[2]

Chapter 4 validates, through signal propagation models and experimental results that interference alignment by motion can be performed general indoor environments. It shows how the required antenna displacement for interference displacement is only an inch or two, given the significant variation in wireless channels (from combining constructively to combining destructively) due to the presence of reflecting surfaces in the environment (e.g., walls and furniture). The chapter describes how the process of sliding antennas to

[2] Recent USB Wi-Fi adapters have sliding antennas whose positions can be manually adjusted to improve the SNR [174].

seek positions of interference alignment, and related techniques like interference nulling, can be automated using a Stochastic Hill Climbing algorithm. Finally, it implements MoMIMO on USRP N210 software radios and Atheros Wi-Fi cards outfitted with movable antennas, and provides experimental results that demonstrate its performance gains over 802.11n Wi-Fi.

2.3 Relation to Prior Work

Recent years have seen multiple cross-layer designs [193, 176, 134] to combat interference. A large number of these systems perform MIMO interference management, e.g., Interference Alignment and Cancellation [63], Beamforming [14], SAM [166], and others [141, 61, 98, 29]. These systems design interference management techniques that focus on specific topologies or traffic patterns. Further, they have been demonstrated on software radios and require modifications to Wi-Fi hardware and standards. In addition, they require multi-antenna senders.

Our work builds on these systems but differs on two dimensions:

- First, OpenRF is the first system that deploys physical-layer MIMO techniques (i.e., nulling, alignment, beamforming) on commodity Wi-Fi cards. In doing so, it is the first cross-layer design that is demonstrated using a fully operational network stack with real applications. OpenRF is a general architecture that applies the right set of MIMO techniques to any topology or traffic pattern. OpenRF leverages a modified 802.11n physical layer and medium-access protocol that runs on today's wireless LANs.
- Second, MoMIMO is the first system to demonstrate that interference alignment and nulling can be performed purely by moving the receive antennas by just about one or two inches. Consequently, MoMIMO can provide the throughput benefits of MIMO interference management to single-antenna devices.

Our work also relates multiple other systems that improve wireless throughput using other strategies that complement our own.

- OpenRF also relates to solutions to improve quality-of-service in enterprise wireless LANs through: frequency management [143, 24, 123, 124, 198], load balancing [39, 154, 196, 75, 164] and leveraging custom data-plane hardware [108, 17]. OpenRF complements this work by adding new MIMO cross-layer techniques to the toolbox available for managing enterprise WLANs.

- MoMIMO is also related to prior work demonstrated that mobility improves the performance of wireless network along different axes: network capacity [66], channel quality in robotic networks [13, 73, 103], beamforming [26] and energy consumption [190]. MoMIMO builds on this past work but differs from it by being the first to show that motion enables MIMO-type interference alignment and nulling.

Chapter 3
MIMO Interference Management on Today's Wireless LANs

Recent years have witnessed the boom of cross-layer wireless designs like SAM [166], Jello [193], MegaMIMO [141], WhiteRate [134], TIMO [61], SoftPHY [176], and many others [98, 167, 29, 63]. Instead of treating the physical layer as a black box, these systems jointly optimize network protocols and physical-layer signal processing. Collectively, they have created a rich literature of cross-layer designs that are implemented in software radios and have shown large throughput gains. Unfortunately, hardly any of these ideas have made it into Wi-Fi chips or real networks. Indeed, a form of stalemate exists: The research community is waiting for cross-layer innovations to be implemented in Wi-Fi hardware so that they may be used in operational networks. Yet, Wi-Fi chip manufacturers cannot make expensive hardware investments without better understanding how these designs interact with real applications and real networks. The situation evokes a similar picture from 10 years ago, when Internet protocols were developed and demonstrated in small testbeds and the ns simulator, but were not adopted by switch manufacturers. Wired networks broke this cycle by building innovations in commodity hardware and directly deploying them in operational networks. OpenFlow, Ethane, and the body of work that led eventually to software defined networks (SDNs) have pioneered this path. We believe that, similarly, cross-layer research needs to start targeting commodity Wi-Fi cards, actual applications, and today's networks.

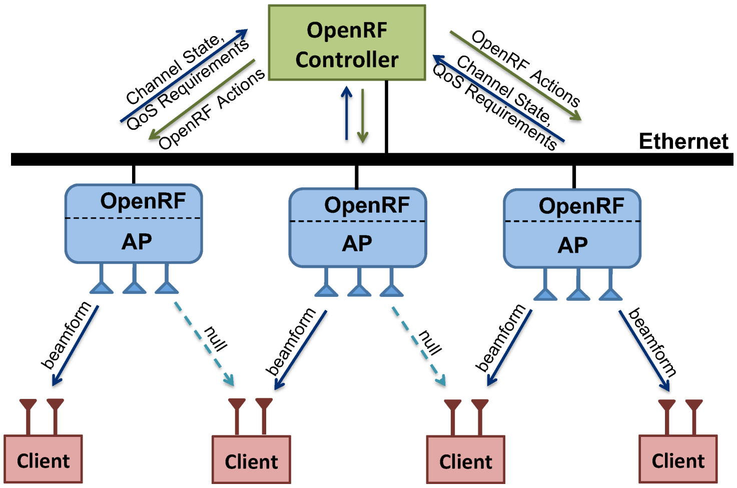


Figure 3-1: Architecture of OpenRF. OpenRF provides an interface to physical-layer MIMO signal processing, e.g., interference nulling, interference alignment, and beamforming. The OpenRF Controller coordinates network devices through this interface.

In this chapter, we take the first step towards this goal. We present OpenRF, a cross-layer architecture for MIMO interference management. OpenRF resides on Access Points (APs) and enables them to control MIMO signal processing at the physical layer. Specifically, OpenRF provides the following capabilities:

- It enables commodity Wi-Fi APs to perform three MIMO interference management techniques: interference nulling [63], coherent beamforming[141] and interference alignment[98].
- It is self-configuring. Network administrators need not understand MIMO signal processing and when to apply a particular interference management technique. OpenRF automatically infers the interference layout of the network, and dynamically applies the right combination of interference nulling, alignment, or beamforming, wherever they are beneficial.
- It translates high-level quality of service requirements of downlink traffic to low-level physical-layer MIMO techniques. For instance, an administrator may request OpenRF to guarantee a minimum rate to real-time applications in the network (e.g., remote desktop or VOIP). OpenRF employs MIMO signal processing to control interference across APs and deliver the desired rate.
- Finally, OpenRF is fully compatible with commodity 802.11n cards that implement transmit beamforming, an 802.11n optional feature. We have built OpenRF as a patch

> to the iwlwifi driver for Intel 5300 cards, providing researchers and network administrators the ability to deploy it in their networks to experience the benefits of cross-layer MIMO, or experiment with new physical-layer MIMO designs.

Architecturally, OpenRF borrows from the SDN design, in that it separates the control plane from the data plane and exposes functions that have traditionally been deeply hidden in the network stack, to higher layers. As illustrated in Fig. 3-1, the data plane is controlled by the *OpenRF interface*, which resides on access points and shields other components from how signal processing techniques are implemented at the device level. Analogous to the OpenFlow interface, the OpenRF interface operates over a table of (FlowID, Actions) tuples. In contrast to OpenFlow entries, where an action may identify which port to transmit the flow on, here, an action specifies the relative power used to transmit the flow on each of the AP's antennas. This is typically referred to in MIMO terms as the pre-coding vector of the flow. Just as forwarding steers a flow's packets toward a particular route in the wired network, pre-coding steers the PHY signal and creates a beam that propagates along a particular spatial direction, allowing the system to null interference at an unwanted receiver and focus the power on the desired receiver.

The control plane in our design is managed by the *OpenRF Controller.* It is configured with per-flow quality of service (QoS) requirements (e.g., a desired rate), as well as which PHY actions are supported by the OpenRF interface. It also periodically takes as input channel measurements from the APs. Using this information, the controller maps the QoS requirements into PHY actions like nulling, beamforming, or alignment. The controller then fills up the OpenRF table with these actions mapped to various flows, so that the needs of each flow can be satisfied.

A key challenge in OpenRF is the need to ensure the whole network stack operates reliably, while performing MIMO interference management. In other words, OpenRF has to account for TCP and application burstiness and the resulting dynamism in the interference patterns. Another challenge stems from the interaction between physical layer techniques and the 802.11 protocol. In particular, PHY techniques achieve gains by enabling concurrent transmissions using intelligent interference management. However, the 802.11 protocol is designed under the assumption that transmitters in the same interference region should not transmit concurrently. This assumption manifests itself in its handling of end-to-end reliability and related functions such as carrier sense, acknowledgments and retransmissions. OpenRF has to ensure concurrent transmissions at the PHY-layer without compromising 802.11's reliability. In §3.3 and §3.4 we explain these challenges in detail and we describe how OpenRF addresses them.

We have built a prototype of OpenRF on Intel 5300 Wi-Fi adapters. We patched the iwlwifi driver to enable nulling, alignment, and beamforming, separately and combined. We deployed a 20-node network, where six of the nodes act as APs and the rest are clients. We compared the performance of the network with and without OpenRF. Our results show the following.

- In a network of 20-nodes, and for a random setup of TCP and UDP flows, the aggregate performance gain in terms of throughput of UDP and TCP flows are 1.7× and 1.6× respectively.
- We also evaluate the impact of OpenRF on the quality of real-time applications. In particular, we evaluate rich applications (multimedia) over Remote Desktop, which is increasingly common in the enterprise. Our results show that OpenRF reduces the percentage of screen glitches in a VNC Remote Desktop client by 6×. It also reduces the 90th percentile delay for VNC by 4×.
- Commodity 802.11n cards can perform beamforming to improve the average SNR at the receiver by 3 dB. Beamforming also leads to a flatter receiver SNR profile across OFDM bins. These reasons cause the bit-rate adaptation algorithm to jump to the next rate 80% of the time, thereby enhancing throughput. The cards can also perform interference nulling and interference alignment to reduce the average interference-to-noise ratio (INR) at the receiver by 12 dB and 11 dB respectively. This enables concurrent transmissions provided the interference is below 10-15 dB.

Contributions: OpenRF is the first system that deploys physical-layer MIMO techniques (i.e., nulling, alignment, beamforming) on commodity Wi-Fi cards. OpenRF dynamically applies the right set of these MIMO techniques to suit any topology or traffic pattern. It is also the first cross-layer design that is demonstrated using a fully operational network stack with real applications. As such, it takes cross-layer design all the way from the physical layer to the application layer, and opens up an opportunity for these technologies to make impact on today's networks.

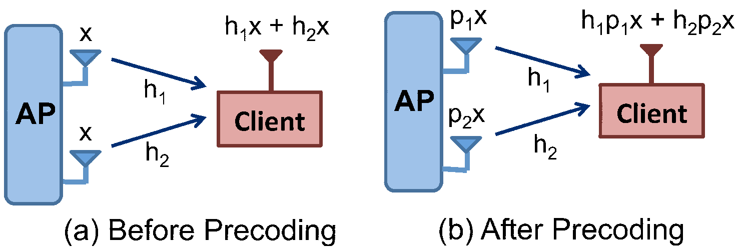


Figure 3-2: Example Topology. We show how a 2-antenna AP can precode its signal to a 1-antenna client.

3.1 MIMO Primer

In this section, we provide a brief introduction to basic MIMO interference management techniques. To understand these techniques, it is important to know the following fundamental properties of MIMO transmissions [98, 199]:

- *An n-antenna node receives signals in n-dimensional space.* For example, a 1-antenna client receives a signal at only one antenna; so it receives signals in one-dimension. Similarly, a client with two antennas receives a signal on both of its antennas. So, the received signal is a vector in a 2-D space.
- *An n-antenna node transmits signals in n-dimensional space.* For e.g., a 3-antenna AP transmits a 3-D vector.
- *n-antenna receivers can decode up to n concurrent signals.*
- *A transmitter can use precoding to change how its signal is received at a particular node.* To do so, it multiplies the transmitted signal by a pre-coding matrix P. The pre-coding matrix can be chosen to null (i.e., cancel) the signal at a particular receiver, beamform the signal, or align it along some space. Below, we explain how these three MIMO techniques leverage precoding.

3.1.1 Interference Nulling

Interference nulling allows a transmitter to completely cancel (i.e., null) its signal at a receiver. For example, suppose a 2-antenna AP wants to null its signal x at a 1-antenna receiver as shown in Fig. 3-2(a). Say the AP sends x on both of its antennas. Let the channels from the two transmitting antennas to the receiver's antenna be h_1 and h_2. The receiver obtains the signal $h_1x + h_2x$. So, how can the AP precode its signal so that the received signal is zero?

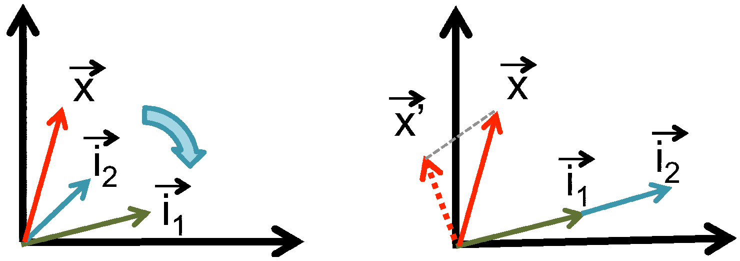


Figure 3-3: Interference Alignment at a 2-Antenna Receiver. The AP uses interference alignment to rotate the interfering signal i_2 along i_1, making the two interferers seem as if there were one.

Say, the AP pre-multiplies the signal at its first antenna by a constant p_1 and second antenna by another constant p_2. The new received signal is now $h_1 p_1 x + h_2 p_2 x$, which needs to be 0. Clearly, this can be solved easily, for example, by setting $p_1 = -h_2$ and $p_2 = h_1$.[1] Thus, the AP can now safely transmit without causing any interference at this client.

We can generalize nulling for a multi-antenna client by using a precoding matrix P such that $HP = 0$, where H is the channel matrix from transmitter to receiver [98, 63].

3.1.2 Coherent Beamforming

Coherent beamforming helps an AP maximize its signal at a receiver, i.e., increase the SNR. Let us revisit the example in Fig. 3-2, where this time, the 2-antenna AP wants to beamform its signal at the 1-antenna receiver. As before, the received signal is $h_1 x + h_2 x$. So, can the AP further increase the power of the received signal?

Fortunately, the AP can indeed do so by applying precoding once again, this time so that the signals from the two antennas add up constructively. In other words, received signal, $h_1 p_1 x + h_2 p_2 x$, needs to be maximized. Since $h_1 p_1$ and $h_2 p_2$ are complex numbers, to maximize their sum we need to choose p_1 and p_2 to align the two complex numbers so that they have identical phases. This can be done by choosing $(p_1, p_2) = (h_1^*, h_2^*)$.

In general, a multi-antenna AP can beamform its streams coherently at a single-antenna client using a precoding matrix $H^*/||H^*||$, where H^* is the conjugate transpose of the channel matrix. [141, 199].

[1] One still needs to normalize to ensure the power after pre-coding sums up to the transmit power budget. For clarity however we ignore normalization, assuming that the transmitter always normalizes its signal before transmission.

3.1.3 Interference Alignment

Suppose a 2-antenna receiver receives two signals - a desired signal along $\vec{x}$ and an interfering signal along $\vec{i}_1$. As mentioned before, these signals can be represented as vectors in a 2-D space, as in Fig. 3-3. Since a 2-antenna receiver can decode two concurrent signals, it can discard the interfering signal, to obtain its desired signal. However, if another interferer joins the network, the receiver can no longer decode, since the 2-D space has a third vector $\vec{i}_2$, as in Fig. 3-3. How can the transmitter of $\vec{i}_2$ avoid interfering at the receiver?

Interestingly, the transmitter of $\vec{i}_2$ can leverage *interference alignment* and precode its transmission to rotate the vector $\vec{i}_2$ and align it along the same direction as $\vec{i}_1$. In effect, the receiver now obtains only two vectors, the desired signal along vector $\vec{x}$, and the sum of interferences $\vec{i}_1 + \vec{i}_2$ along a different direction, as shown in Fig. 3-3. Thus, it can easily decode by treating the unwanted interference $\vec{i}_1 + \vec{i}_2$ as one signal from a single antenna, and projecting $\vec{x}$ orthogonal to the interference.

In general, a multi-antenna AP can align its signals along the vector space V using a precoding matrix P, such that, $V^{\perp}HP = 0$, where $(.)^{\perp}$ denotes the orthogonal vector space [98, 63].

Note that since all the above techniques leverage MIMO precoding, they can be combined in different ways by a transmitter that has a sufficient number of antennas.

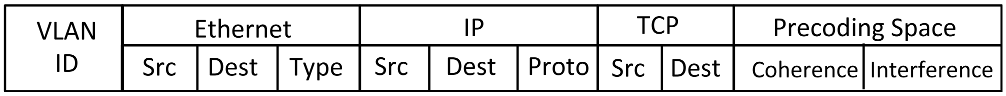
3.2 OpenRF's Design Principles

OpenRF provides a general framework for applying MIMO PHY techniques in an Enterprise WLAN. It resides at the APs and can perform interference nulling and coherent beamforming without modifications to the clients. It can also perform interference alignment by patching the client's driver as explained in §3.4.3.

OpenRF is compatible with commodity 802.11n cards that implement transmit beamforming, an 802.11n optional feature.[2].

OpenRF's design is based on the realization that cross-layer interference management techniques can be decoupled into two classes: *coherence* techniques and *interference* techniques. Coherence techniques aim to maximize the signal strength at a receiver, e.g., co-

[2] The Transmit Beamforming feature allows OpenRF to set precoding vectors (or matrices) on 802.11n cards, with certain restrictions, as explained in §3.4.3.



VLAN ID	Ethernet			IP			TCP		Precoding Space	
	Src	Dest	Type	Src	Dest	Proto	Src	Dest	Coherence	Interference

Figure 3-4: OpenRF Flow Table. OpenRF defines flows based on packet headers, similar to OpenFlow.

herent beamforming (See §3.1.2). In contrast, interference techniques allow APs to transmit additional concurrent streams, provided these streams do not interfere with existing parallel transmissions. Interference techniques include nulling, and alignment (See §3.1.1 and §3.1.3). Interestingly, coherence techniques can be performed completely locally, as they only involve a single AP transmitting to its own clients. In contrast, interference techniques need APs to synchronize with other APs, so that they do not interfere at each other's clients, while they transmit concurrently on the shared wireless medium. Thus, OpenRF's first policy is: *only interference techniques need to be coordinated (i.e., scheduled) by the central controller.* This rule reduces coordination complexity.

Second, just like OpenFlow steers a flow's packets toward a particular route, OpenRF steers the signal of each flow along a particular spatial direction. This is done by precoding the signal before transmission. However, unlike OpenFlow, OpenRF does not directly assign a precoding vector (or precoding matrix) to each flow. This is because MIMO APs need to combine transmitting flows *coherently* to their clients, along with canceling *interference* at other APs' clients. Fixing the precoding vector for a particular flow would also fix the set of flows that can be concurrently transmitted with that flow. However, due to traffic burstiness, the set of flows which need to be combined together changes frequently depending on which flows have pending packets in the queue. Thus, instead of assigning a precoding vector per flow, OpenRF assigns to each flow an *interference vector* and a *coherence vector*. Now suppose the APs consider a set of flows for concurrent transmission depending on which flows currently have pending packets in the queue. At each AP, OpenRF can now combine, in real time, the relevant coherence and interference vectors, to produce a precoding vector that enables transmitting the desired flow coherently, while nulling any interference that may affect other flows. Hence, OpenRF's second policy is to perform *late binding of interference and coherence decisions*.

In the following section, we explain the above policies and their implementation in greater detail.

3.3 OpenRF's Architecture

OpenRF is architecturally similar to SDN in that it separates the control and data planes, and exposes functions that have traditionally been deeply hidden in the network stack to higher layers. The data plane is controlled by an open interface that is managed in software by a controller. In this section, we describe the architecture of both the OpenRF Interface and the OpenRF controller.

3.3.1 OpenRF Interface

The OpenRF Interface operates over a table indexed by flows, which describes how the AP handles different flows in the network. Flows are identified based on fields in the packet header, such as destination IP address, port number etc. (Fig. 3-4). Additionally, OpenRF maintains for each flow, a coherence vector, which specifies the direction along which the signal is received coherently at the client, and an interference vector which specifies the direction that we need to be orthogonal to in order to avoid interference at this client.

The concept of an interference vector and a coherence vector (or more generally an interference matrix and a coherence matrix) is best explained via an example. Consider the scenario in Fig. 3-5, where a 3-antenna AP wishes to beamform its signal to its client, Alice, while nulling interference at Bob, the client of a different nearby AP. Let the channels to Alice be h_{a1}, h_{a2}, and h_{a3}, and the channel to Bob be h_{b1}, h_{b2}, and h_{b3}, as shown in the figure. We can express these channels as two 3-dimensional vectors, $\vec{h}_a$ and $\vec{h}_b$.

Now, any pre-coding vector that the AP computes must satisfy the following two conditions:

- **Interference Condition:** To null to Bob, the AP has to pre-code its signal such that it falls in the pink-colored plane orthogonal to Bob's channel vector, as shown in Fig. 3-5. Specifically, let $\vec{p}$ be the pre-coding vector that the AP applies to its signal. Then to null to Bob, we need $\vec{h}_b \cdot \vec{p} = 0$.

- **Coherence Condition:** The AP also wants to beamform its signal to Alice. In the absence of the nulling constraint, beamforming requires the AP to align its transmission with Alice's channels, i.e., $\vec{p} = \vec{h}_a^*$. However, this pre-coding vector creates interference at Bob as it does not satisfy the nulling condition $\vec{h}_b \cdot \vec{p} = 0$, thereby making concurrent transmissions to Alice and Bob infeasible. Thus, the AP's best option

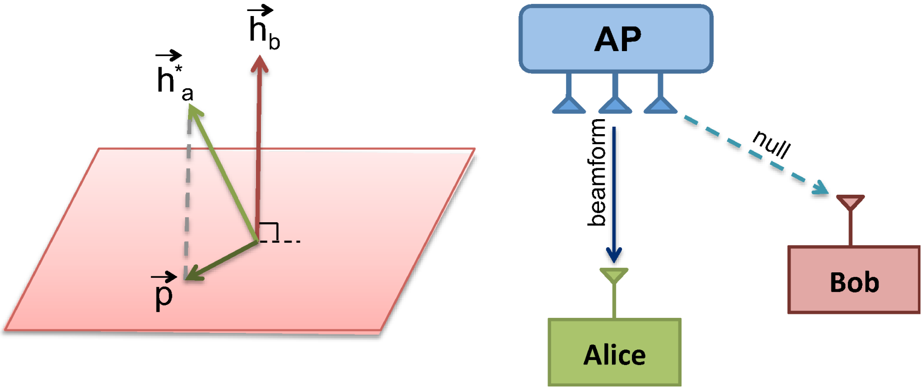


Figure 3-5: Computing the precoding vector. Suppose an AP needs to beamform its signal to Alice, while nulling interference at Bob. Then the precoding vector $\vec{p}$ is the projection of coherence vector $\vec{h}^*_a$ onto the plane orthogonal to interference vector $\vec{h}_b$.

is to pick a pre-coding vector that satisfies the nulling condition but is as close as possible to $\vec{h}^*_a$. To do so, the AP projects $\vec{h}^*_a$ on the plane that nulls the signal to Bob, as shown in Fig. 3-5.

In the above example, Bob's interference vector is his channel $\vec{h}_b$, whereas Alice's coherence vector is the conjugate transpose of her channel $\vec{h}^*_a$. The AP can keep these vectors in the OpenRF table along with the entries of Alice and Bob. Whenever it wants to null to Bob and beamform to Alice, it uses these vectors to compute the required precoding vector $\vec{p}$ and applies it to the transmitted signal. The AP may also combine nulling to Bob with beamforming to a client other than Alice, say, Charlie. To do so, it only needs to combine Bob's interference vector with Charlie's coherence vector. These decisions can be made in real-time depending on which clients have packets pending at the AP.

While the above example deals with single-antenna receivers, it can readily be generalized to typical multi-antenna MIMO systems that leverage MIMO multiplexing. In the following section, we formalize our definitions of the coherence and interference vectors and the computation of the precoding vector. Note that as we generalize to multi-antenna MIMO clients, the vectors become matrices.

3.3.2 Formalizing the Precoding Computation

Suppose an n-antenna AP needs to transmit k independent MIMO streams $X_{k\times 1}$ to an m-antenna client, where $k \leq m$. Let $H_{m\times n}$ be the channel matrix, where $m \leq n$. The AP applies a precoding matrix $P_{n\times k}$ so that the received signal: $Y = HPX$.

The AP can choose the precoding matrix P to satisfy certain interference and coherence conditions. We define:

- **Coherence Matrix:** The coherence matrix C_i for flow i identifies the space along which the signal should be transmitted to increase the SNR at the client. For coherent beamforming to a single antenna client, we define $C_i = H_i^*/||H||$, where H is the channel matrix to the client. More generally, we define $C_i = Eig_k(H^*H)$, where $Eig_k(M)$ denotes the k eigen vectors of M with the largest eigen values.
- **Interference Matrix:** The interference matrix I_i for flow i identifies the space to which the signal should be orthogonal in order to avoid interference at the client. For interference nulling $I_i = H_i$. For interference alignment where the client is aligned along the direction (or, more generally, space) specified by the vector (or, matrix) V_i, $I_i = V_i^{\perp} H_i$.

Note that the above matrices allow for beamforming, and nulling/alignment to MIMO clients that receive multiple streams.

Computing the Precoding Matrix: Now that we have defined coherence and interference matrices, we need to explain how OpenRF computes the precoding matrix, in real-time, for an arbitrary set of concurrent flows. Assume that the OpenRF controller (described in §3.3.3) has already populated the interference and coherence matrices for a set of flows in the OpenRF table. Suppose the controller decides that the AP needs to transmit a flow 1, while concurrently canceling any interference caused to concurrent flows: $2, 3, \ldots, n$.

Let C_1 be the coherence matrix of flow 1 in the OpenRF table, and $I_2, \ldots, I_n$ be the interference matrices of flows $2, \ldots, n$. OpenRF first computes the combined interference matrix I by concatenating these interference matrices, i.e. $I = [I_2 \ \ldots \ I_n]^T$. The matrix I denotes the space that the signal should be orthogonal, to avoid interference at all of flows: 2, …, n. Intuitively, OpenRF now needs to project its coherence space C_1, orthogonal to the interference space I. This can be interpreted as a standard geometric problem, similar to Fig. 3-5, but generalized to n-dimensional space. OpenRF computes the solution as the precoding matrix: $P = I^{\perp}(I^{\perp})^* C_1$, where $I^{\perp}$ denotes the null-space of matrix I.

■ 3.3.3 OpenRF Controller

The OpenRF controller has two components: a central component that coordinates all APs, and a local component residing on each AP that adapts to real-time changes to channels and traffic patterns. We refer to this local component as the *local agent*. As argued in §3.2, OpenRF's key design principle is that the central controller manages only interference spaces across APs, and delegates managing the coherence space to the local component at each AP.

To schedule concurrent transmissions across APs, we divide time into slots. Having short slots corresponding to the size of a packet can cause excessive overhead. Hence, OpenRF leverages 802.11n's aggregate frames, which combine multiple MAC-layer packets into a single PHY-layer frame from the perspective of medium access. This allows us to set the slot size corresponding to the size of an aggregate frame, which in our system defaults to 5 ms.

The central controller designates some slots for scheduling edge clients where two or more APs interfere.[3] Other slots are scheduled locally, by the local agent at each AP. This limits coordination overhead across APs to only these centrally scheduled slots, which we call the *interference slots*.

Once the central controller assigns the interference slots, the local agents on the APs abide by the following contract: (1) Any flow that suffers interference from other APs (i.e. flows to edge clients) can only be transmitted in its own interference slot. (2) Interfering APs that wish to concurrently transmit during this slot *must* perform the interference management technique recommended by the central controller, i.e., interference alignment or nulling. The local agents, however continue to have the flexibility to schedule any flow locally based on the dynamic traffic patterns, in any slot, provided the flow does not suffer from interference.

To illustrate the above rules, let us consider the simple example in Fig. 3-6. Here, AP-1 has two clients Alice and Bob, AP-2 has two clients Charlie and Dave. The pale blue circles show the interference range of each AP. For simplicity, we assume that all channels support the same rate and that there are no QoS requirements. Suppose the controller assigns every alternate slot as interference slots for edge client Charlie. During such a

[3] Edge clients can be identified by checking if the SNR based on the channel matrices of APs is above a nominal threshold.

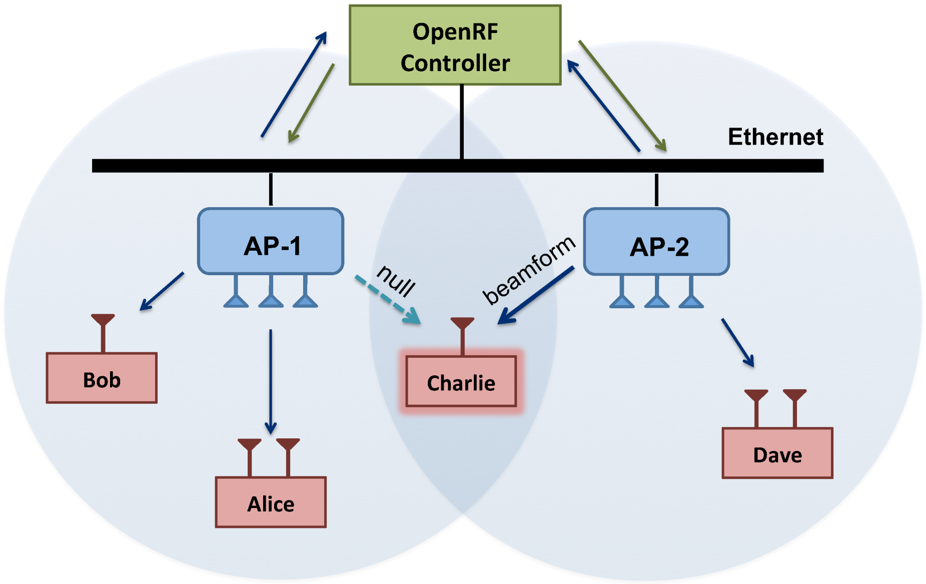


Figure 3-6: Illustrative Example. In this example, there are two 3-antenna APs: AP-1 and AP-2, and four clients.

slot AP-1 and AP-2 follow these policies: (1) AP-1′s contract is to null its signal to Charlie during all of his interference slots. However AP-1 is free to transmit concurrently to either Alice or Bob, depending on who has traffic. (2) AP-2′s contract is that it can transmit to Charlie concurrently to AP-1 only in its designated slots (We explain how to selectively enable concurrent transmissions in 802.11 in §3.4). However, AP-2 is free to transmit to David, in either slot, concurrently with AP-1.

In the following paragraphs, we explain how the central controller and local agent function.

Local Agent. The goal of the local agent is to dynamically schedule which flow an AP must transmit to during each time slot. The local agent takes as input the list of interference slots, and the QoS requirements of different flows. The controller then employs the following algorithm described in Alg. 1, inspired by deficit round robin scheduling. At a high level the algorithm maintains a *credit counter*, c_i, for each flow i. In general, the credit counter of a flow is large, if it has not been scheduled for an extended period. Now at any time slot, the local agent picks the flow f whose credit counter is the highest. It then measures the number of bytes b it could send for this flow in this slot. Finally, it updates the credit counters, by reducing the credit of this flow, based on b, and redistributing this equally among all other flows. In effect, this ensures that the medium is shared fairly between all flows.

However, to support OpenRF, we need the following additional modifications: First, consider an interference slot, where other APs null interference for flow f, belonging to this AP. In such a slot, the AP always picks flow f, if f has pending packets in the queue.

Second, the local agent enforces two kind of QoS requirements: a proportional allocation of throughput, and fixed reservation of throughput. These requirements are configured by the network manager, for flows identified by their packet headers. Note that we do not consider delays in our current system. Proportional allocation requires the flows of an AP to achieve throughputs proportional to some set of weights: $\{w_i, \text{ for all flows } i\}$. This can naturally be incorporated into the local agent's algorithm. Specifically, instead of redistributing credit equally between flows, we redistribute it proportionally based on weights $\{w_i\}$ (Alg. 1, Line 13). This ensures that flows are allocated precisely according to these weights.

Now that we know how to allocate throughput proportionally, how do we enforce fixed reservations? In this scenario, we have two kinds of flows: Quality of Service (QoS) flows, which have fixed throughput requirements, and Best Effort (BE) flows, must have equal (or more generally, proportional) throughput between them to ensure fairness. Interestingly, the local agent can translate the problem of fixed reservations to proportional allocation with dynamic weights. In particular, the agent computes e_i, the expected number of bytes that it hopes to send during time span t for a flow. For QoS flows, e_i is simply $qos_i * t$, where qos_i is the throughput requirement. In contrast, all BE flows must achieve equal throughput in this time, so e_i is the mean of the throughput achieved for all BE flows during time span t. (Alg. 1, Line 11) The agent now applies proportional allocation by resetting the weights w_i, in every time slot, to e_i/b_i, where b_i is the number of bytes sent so far for flow i. In effect, this biases the weights to ensure that b_i is as close as possible to e_i, hence satisfying the QoS requirements.

Central Controller. The central controller assigns interference slots that serve flows to edge clients in the network. The controller takes as input the network state, i.e. channel state information, the list of clients and their flows. Similar to the local agent above, the central controller uses credit counters to decide the number and list of interference slots,

1 Pseudo-code for the Local Agent Algorithm

```
function LocalAgent($t$, $qos$, $slots$)
    ▷ $t$: Slot, $qos$: QoS needs, $slots$: Interference Slots
    if $t \in slots$ then                                   ▷ $t$ an interference slot
        $f =$ Flow for slot $t$ (if flow has packets to send)
    else $f = arg[max_i(d_i)]$                              ▷ $f$: Flow with most deficit
    end if
    $d_f = d_f - b$                                         ▷ $d_f$: deficit, $b$: no. of bytes sent
    $b_f = b_f + b$                                         ▷ $b_f$: no. of bytes sent so far by $f$
    for each flow $i$ do
        ▷ $e_i$: Expected no. of bytes for QoS or BE flows
        $e_i = qos_i * t$, if QoS (or) $avg_{k \in BE}(b_k)$, if BE
        $w_i = e_i / b_i$                                   ▷ $w_i$: weight for flow $i$
        $d_i = d_i + b * w_i / \sum_k w_k$                  ▷ Update deficits
    end for
    return $f$
end function
```

while accounting for QoS, BE or proportional throughput requirements for different flows in the network.

The controller applies the following heuristic algorithm for each AP i in the network: First, it uses credit counters, similar to the local agents, to pick flow f_i for AP i. After choosing f_i, the controller iterates over all other APs (besides AP i), and checks if any of these APs may potentially interfere with f_i. For such APs, the controller decides how they should manage interference so that they can concurrently transmit with f_i. In particular, it prescribes interference nulling if f_i is to a single-antenna client, and interference alignment, otherwise.[4] Next, these APs are informed that the present slot is an interference slot allotted to f_i.

At this point, the controller must update the credit counters based on the throughput for the set of flows$\{f_i\}$. But recall that unlike the local agent, the central controller cannot obtain the number of bytes that have been sent in a slot. In other words, the controller needs to determine the throughput that each flow will achieve. Fortunately, this can be done fairly accurately by calculating the effective channel of each client, which is the product of the client's channel and the AP's precoding matrix, and then estimate the throughput using the ESNR algorithm [69].

Finally, the Central controller, unlike the local agent, can leverage its global view of

[4] Sometimes, the AP may not have enough antennas to simultaneously null/align to multiple flows. In such cases, this AP is not permitted to transmit to any of its clients during this interference slot.

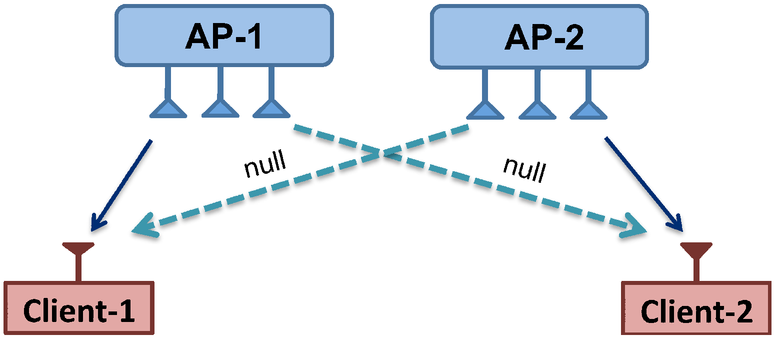


Figure 3-7: Interference Nulling. Here, AP-1 and AP-2 transmit concurrently to their respective clients: client-1, and client-2, by nulling interference at each other's clients.

the network to ascertain whether a flow's fixed throughput requirement can be satisfied. Whenever a new QoS flow joins the network, the controller performs the following *admission control algorithm*: It runs the above central controller algorithm with the new throughput requirement factored in. It then admits this flow as a QoS flow only if the AP achieves the requisite throughput requirement of this flow, at the end of the algorithm.

■ 3.4 Integration with 802.11 protocol

To implement OpenRF on commodity Wi-Fi cards, it must be integrated with the 802.11 protocol. Unfortunately, the 802.11 protocol was not designed with MIMO interference management techniques in mind. This causes subtle interactions between PHY-layer interference management and 802.11's MAC, as described below.

■ 3.4.1 Ensuring Reliability

End to end reliability is essential for the correct operation of TCP and most applications. WLANs achieve reliability using 802.11's carrier sense, acknowledgments and retransmissions, which together provide an abstraction of a reliable communication channel to higher layers of the network stack. Unfortunately, these three reliability mechanisms do not lend themselves naturally to concurrent transmissions in the same interference region, which are essential for most PHY-aware techniques to achieve throughput gains.

(a) Carrier Sense: Consider the example in Fig. 3-7, where we have two APs, each transmitting streams to their own clients, while nulling interference at each other's clients. If carrier sense is disabled on the two APs, OpenRF will enable these APs to transmit con-

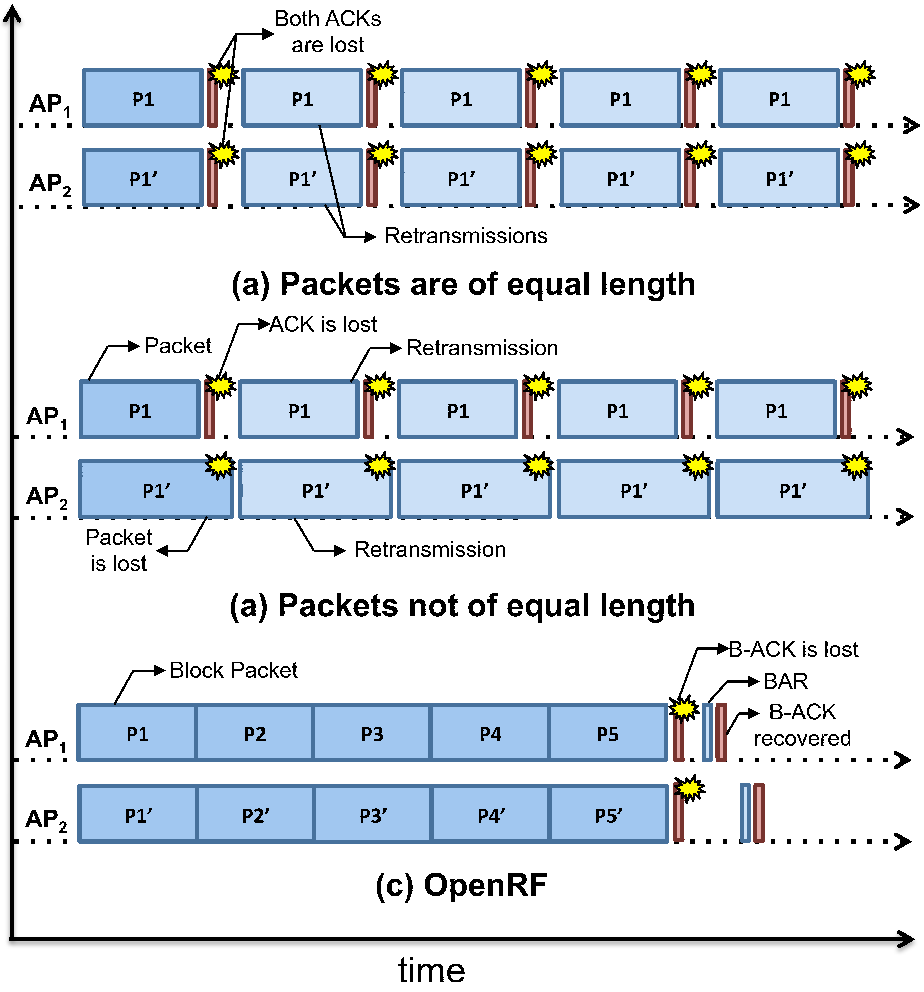


Figure 3-8: Collision of MAC-Layer ACKs. In Case (a), packets are of equal length, causing ACKs to repeatedly collide with each other. In Case (b), packets are of unequal lengths causing ACKs repeatedly to collide with packets. In contrast, OpenRF ensures that the system recovers from any MAC-layer ACK collisions.

currently in the same interference region and correctly deliver packets to their own clients (as described in the previous section). However, in the presence of carrier sense, one of the APs will begin transmitting before the other, causing the other AP to carrier sense the signal and abstain from concurrently transmitting. Thus, to provide concurrent transmissions at the physical layer, we need to make the two APs carrier sense only when appropriate. Of course, one option is to deactivate carrier sense altogether; but that would lead to severe collisions with uplink traffic.

To ensure that the two APs in Fig 3-7 transmit concurrently, despite carrier sense, OpenRF synchronizes their packets so that they start precisely at the same point and therefore effectively do not carrier sense each other. But how do we ensure that the two APs start exactly at the same time? Recall that 802.11 nodes decide at what time to transmit as follows: If the node senses the medium as idle, it picks a random slot between 0 and CW_{min}, and transmits starting from that slot. Hence, if we convince two nodes to pick exactly the same slot, they will never sense each other and will transmit concurrently starting from that slot. One way to achieve this is to set CW_{min} to zero, causing both APs to always transmit concurrently at the 0-slot. However, this alone is not enough, as always picking the zero slot gives the APs unfair access to the medium compared to other nodes in the network.

To recover fair medium access, we leverage the AIFS parameter in 802.11e/n standards. Instead of applying the same DIFS period to all types of traffic, 802.11e/n allows different traffic classes to replace the standard DIFS waiting period by a different waiting time referred to as the arbitrary inter frame spacing (AIFS). This enables different classes of traffic to have varying levels of priority. In particular, we leverage AIFS as follows: Whenever two APs have to transmit concurrently, we set their AIFS period to a fixed value of $DIFS + CW_{min}^{def}/2$, where CW_{min}^{def} is the default value for CW_{min}. In addition, we set CW_{min} to zero, so that both APs always send concurrently, precisely after AIFS. Together, these two mechanisms ensure that the two APs transmit concurrently in slot zero, but as their AIFS is longer than DIFS, it will look to other nodes as if they picked the middle slot $CW_{min}^{def}/2$, which is precisely what we need for average fairness. Note that APs can send flows that are not meant to be transmitted concurrently with other APs, independently using standard AIFS and contention windows.

(b) MAC ACKs and Transmission: The above solution allows us to synchronize the two access points and make them transmit their data packets concurrently. However, since 802.11's acknowledgments are transmitted immediately after a packet, they will invariably cause collisions. Specifically there are two scenarios as shown in Fig. 3-8: (1) The concurrent packets span the same length on the wireless medium, in which case, their ACKs will collide as shown in Fig. 3-8(a). (2) The concurrent packets span different lengths on the medium, which causes ACKs to collide with data packets as shown in Fig. 3-8(b). In either case, APs do not receive acknowledgments for their data packets, and therefore they will end up retransmitting the packets again and again, eliminating any potential gain.

There are a variety of solutions to mitigate this, which are not compatible with the 802.11 standard. For example, one may propose that since the APs have multiple antennas, they may jointly decode the colliding ACKs. Unfortunately, joint decoding is not possible with the current 802.11 standard. Alternatively, one may propose applying interference nulling to ACKs on the uplink, similar to the downlink data packets. However, coordination between disparate clients is relatively difficult compared to the APs, since the clients are not connected to a wired backend.

To address the above problem, we leverage 802.11n block aggregation. Specifically, block aggregation allows you to bond multiple MAC-layer packets (MPDUs) to the same destination that span equal time durations of up to 5 ms on the medium as shown in Fig. 3-8(c). Note that packets may have different lengths and may be sent at different rates leading to blocks of sizes below 5 ms. So, we ensure blocks span precisely 5 ms by suffixing dummy MPDUs at the end of the block if necessary. Using block aggregation greatly reduces the number of ACKs in the system, since the client issues only a single *block acknowledgment* for all MPDUs in a block. However, the two block ACKs of concurrently transmitted blocks will continue to collide. Fortunately, when a block ACK is lost, the access point does not retransmit the entire block once again. Instead, the 802.11 standard specifies that the AP issues a short block acknowledgment request for the block ACK[2]. By sending this block acknowledgment request using standard best effort with default AIFS and contention windows, OpenRF can ensure that their is no concurrency of the block requests, therefore the block acknowledgments do not collide as shown in Fig. 3-8(c).

In summary, the above mechanisms ensures that packets that are transmitted concurrently are still acknowledged correctly, providing the expected level of reliability to TCP and higher-layer applications. Furthermore, the only overhead we incur to ensure reliability is the loss of up to one block ACK at most every 5 ms, which is relatively low, compared to the gains from concurrency.

■ 3.4.2 Working with Unmodified Clients

PHY-aware techniques such as interference nulling, interference alignment and beamforming, all require knowledge of the channels from APs to their clients. To this end, one possible approach is to require clients to measure their channels and provide this information as feedback to the APs. However, this approach has two problems: First, it incurs high overhead. Second, many clients may not have the capability to measure these channels in the first place. For example, a single antenna client cannot measure the chan-

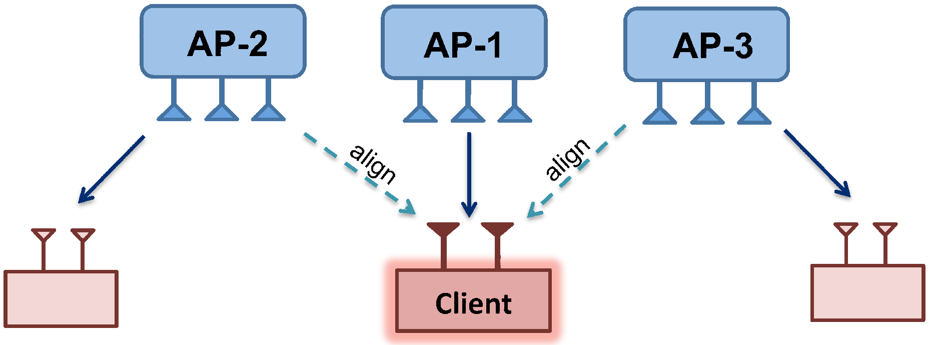


Figure 3-9: Interference Alignment. In this example, we have three AP-client pairs. AP-1 and AP-3 align their interfering signal to enable the 2-antenna client to decode AP-2's signal.

nels simultaneously from a 2-antenna AP. One might wonder if it suffices for the AP to transmit two individual packets: the first packet on its first antenna and the second on its second antenna. While this would allow the client to measure the channel from each transmit antenna separately, such measurements will be separated by at least hundreds of microseconds. However channels required for performing nulling or beamforming must be measured at nearly the same time, and can at most be separated by a few OFDM symbols [141].[5] So how can we perform nulling, alignment or beamforming at APs, without requiring clients to send their channel state information?

To address this problem we use the principle of channel reciprocity[63]. Reciprocity states that the channel in the forward direction is the same as the channel in the reverse direction, up to some calibration constant. In effect, APs can now use packets sent from the clients to measure the reverse channel, and then use those measurements to infer the forward channel needed for interference management. However, the AP still needs to know the calibration constant to derive the forward channel from the reverse channel.

Fortunately, this calibration constant only depends on the transmitter. It can be calibrated a priori by the having each AP measure the forward and reverse channels to any other AP and thereby infer the calibration constant.[6]

[5] If not, the channels would be severely distorted by an accumulated phase difference owing to the frequency offset between the clocks of the AP and client, as well as phase noise [141].

[6] The reason the calibration constant depends only on the transmitter, is that all interference management techniques rely only on the ratio of the channels from the transmit antennas, and not their absolute values [61]. As a result constant factors that depend only on the receivers are canceled out.

■ 3.4.3 Overcoming Limitations of the 802.11n standard

Interference Alignment with today's MIMO clients. Implementing interference alignment with today's 802.11 nodes is quite challenging. In particular, consider the example in Fig. 3-9, where we have a 2-antenna client receiving its desired signal from AP-1. Two other access points, AP-2 and AP-3 align their transmissions together at this client to collapse their interfering signals into one stream. In principle, the client in Fig. 3-9 can decode his desired signal by projecting the signal it received from AP-1 orthogonal to the common direction of the two interferers (See Fig. 3-3). This is the standard approach for decoding multiple streams, and, in particular, interference alignment. The client should now ignore the interfering signal, since it does not need it, and in fact, cannot decode it. This mechanism has been used in several past designs implementing alignment on software radios [63, 98]. Unfortunately, current 802.11 standards enforce fate sharing between the different MIMO streams received by a MIMO receiver. Specifically, the client in Fig. 3-9 is not designed to ignore the interfering stream and simply accept the desired stream that it was able to decode correctly (802.11 applies a single convolutional code across all streams. So, errors in one stream percolate to all streams). So how can OpenRF support interference alignment despite this major limitation?

Our solution to this challenge is counter-intuitive. The only way to implement interference alignment on off-the-shelf MIMO nodes is to convince the receiver that one of his antennas is not functional! Specifically, the driver can issue a command to deactivate one of the client's antennas. Suppose we now align the two interfering streams along antenna 1, by setting the direction of alignment V_i (see §3.3.2) as the unit vector along the axis corresponding to antenna 1 . We then ask the client to deactivate antenna 1 (Fig. 3-10).[7] Then, effectively, the client will not see the interfering stream altogether, and it will only observe its desired stream, which can now be decoded. This effectively tricks the client into performing interference alignment by ignoring the fate sharing between MIMO streams. Therefore, our approach allows AP-2 to transmit to its own client, even while AP-1 and AP-3 are concurrently transmitting to their own clients, thereby increasing concurrency in the network.

It has to be Unitary! Past literature, as well as our discussions so far, assumes that we

[7] More generally, for better performance, the client dynamically deactivates the antenna with poorest channel to the desired AP.

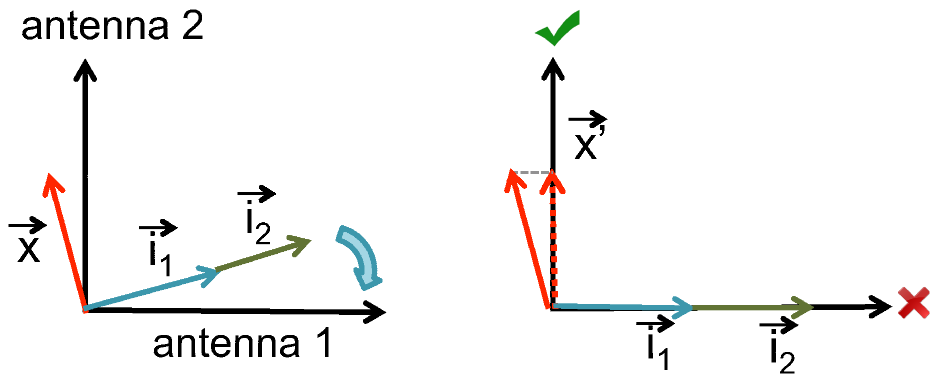


Figure 3-10: Enabling Alignment for Today's clients. OpenRF implements alignment for the client by aligning all interfering signals: i_1 and i_2, along antenna 1. The client disables antenna 1 to decode desired signal x on antenna 2.

can set the precoding vectors to arbitrary values according to the desired PHY-technique. However, the 802.11n compressed transmit beamforming feature requires the matrix of precoding vectors to be unitary. For example, an AP with 3 antennas, applies 3 precoding vectors which forms a 3×3 matrix. This matrix has to be unitary. To ensure this, after computing precoding vectors, OpenRF uses geometric transforms to express the ultimate precoding matrix, used by the device, as a unitary matrix.

3.5 Implementation

We implemented OpenRF by modifying the iwlwifi driver for Intel Wi-Fi cards on Ubuntu 10.04 LTS. We built our solution over the University of Washington's 802.11 CSI tool [67]. Our APs also use the transmit beamforming feature of the Intel 5300 Wi-Fi cards.[8]

To support the OpenRF Interface, we made the following modifications to the driver: First, we implemented the OpenRF table (Sec. §3.3.1) as a debugfs file that interfaces between user and kernel modes. Second, OpenRF uses the 802.11 CSI tool to calculate the reciprocal channels from the clients, which are used to maintain the coherence and interference spaces in the OpenRF table for various flows at this AP. Finally, we employ the transmit beamforming feature to set precoding vectors for various flows in real-time, using the coherence and interference spaces as in §3.3.2.

To support the OpenRF Controller at each AP, we first employ block aggregation to

[8] While we chose to implement OpenRF on Intel cards, OpenRF's design is compatible with any 802.11n card that supports the transmit beamforming feature[2].

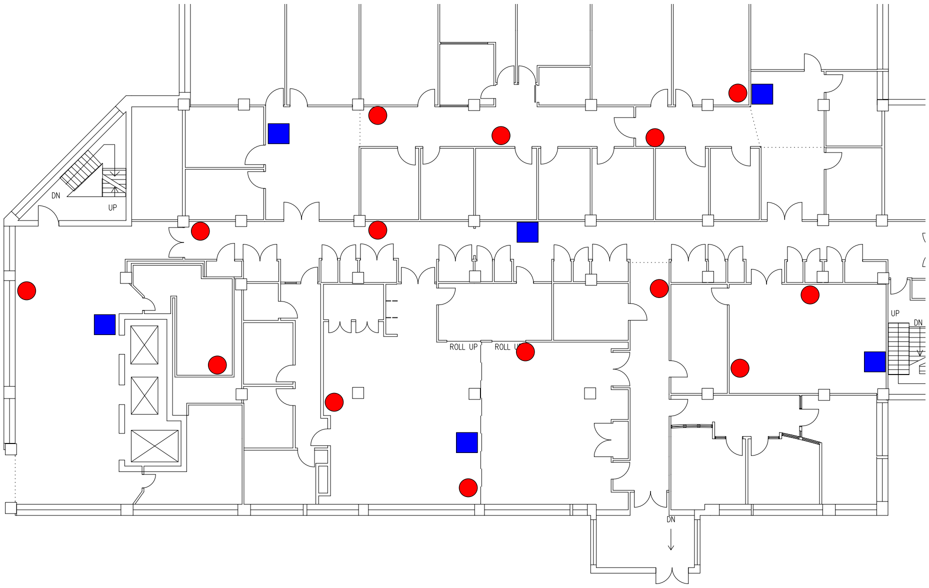
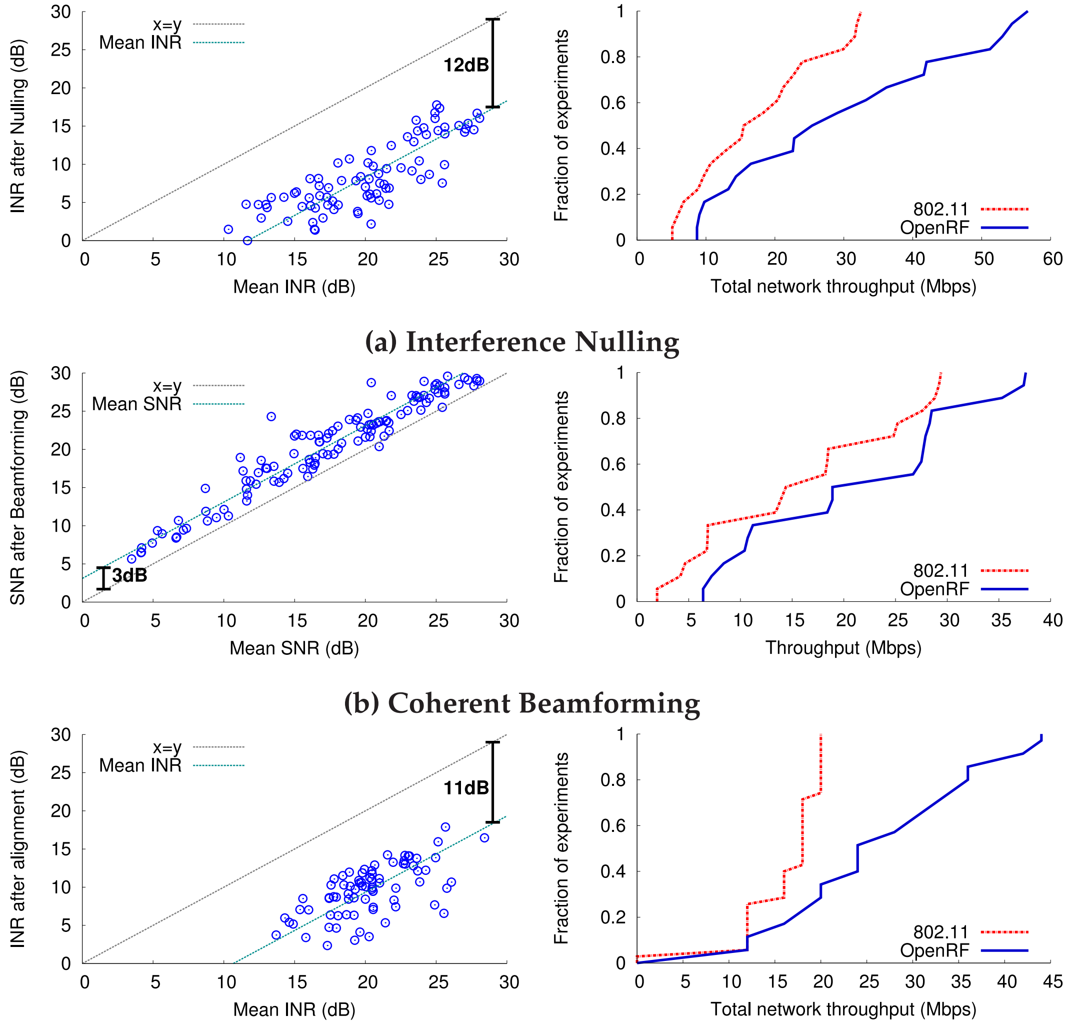


Figure 3-11: Floor map for our testbed. Our 20-node testbed includes 6 APs (blue squares) and 14 clients (red circles).

implement time slots spanning 5 ms. We use NTP over the wired Ethernet backbone to synchronize the block aggregation slots across APs in the network. We observed that this was sufficient to synchronize blocks spanning 5 ms each, within tens of microseconds. We intercept packets from the higher layers into per-flow queues at the driver and apply the local controller algorithm as described in §3.3.3. Our controller and scheduler algorithms are implemented using high resolution timers to minimize waste of airtime. The central controller is implemented on a Linux workstation, and coordinates APs in the network over the Ethernet backbone using the algorithm described in §3.3.3.

Finally, our implementation fully complies with 802.11n and handles concurrent transmissions, carrier sense, and ACKs as in §3.4.

We deployed OpenRF on a network of 20 nodes, each containing an Intel Wireless-N series card connected to a PC or laptop. Our network has six 3-antenna APs, spread out on a single floor of a large building as shown in Fig. 3-11. Each AP is a Linux PC running hostapd, and implementing our patched iwlwifi driver. We deploy 14 clients, including seven single-antenna, five 2-antenna and two 3-antenna wireless nodes. We randomize the locations of the clients on the floor across different experiments (a candidate set of client locations is shown in Fig. 3-11). All APs in the network share the same wireless channel, potentially causing interference at some clients, that are at the edge of their AP's communication range.



(a) Interference Nulling

(b) Coherent Beamforming

(c) Interference Alignment

Figure 3-12: MIMO Interference techniques. We plot the SNR or INR of the client and CDF of total throughput for TCP flows with and without (a) Interference Nulling. (b) Coherent Beamforming. (c) Interference Alignment.

■ 3.6 Results

We compare OpenRF in various settings with a standard 802.11n baseline that uses block-aggregation but does not perform interference nulling, coherent beamforming or interference alignment.

■ 3.6.1 PHY-aware techniques on Commodity Cards

First, we evaluate how effectively commodity Wi-Fi cards can perform common PHY-aware techniques, like, interference nulling, coherent beamforming and interference alignment. We evaluate how these techniques impact TCP flows generated using iperf [170] in our 20-node network in Fig. 3-11. We use a mix of both long and short lived TCP flows. Each client has one long lived flow. Additionally, short flows are generated according to a Poisson arrival process, with mean inter-arrival time of 1s, and have a Pareto file size with mean of 125KB and shape parameter of 1.5 [133, 36].

We use the OpenRF Controller to identify scenarios corresponding to different MIMO interference management techniques, applied in this network. We repeat the experiment across randomly chosen client locations. In each case, we compare our system's performance with 802.11, by replaying identical traffic patterns.

Experiment 1: (Interference Nulling) We consider scenarios reported by the controller where two single-antenna clients obtain signals from two APs as shown in Fig. 3-7. To mitigate interference, the controller applies interference nulling on the APs, so that they null any unwanted interference at neighboring clients. We measure the interference-to-noise ratio (INR) at each client from its unwanted AP, with and without OpenRF's nulling, across experiments. We also measure the total throughput of the TCP flows per client, with OpenRF's interference nulling and standard 802.11.

Results 1: The plot in Fig. 3-12(a) shows that the INR reduction from Interference Nulling is an average of 12 dB. This is a substantial reduction in INR, and it is sufficient for enabling concurrent transmissions in common wireless networks. This is because the operational SNR of 802.11 is in the range of 5 - 25 dB [141]. Furthermore, clients that experience interference are often near the boundaries of the communication range of two Access Points. Such clients receive low SNR from their interfering APs, which still severely disrupts signals from their own AP. In fact, our results in Fig. 3-12(a), demonstrate that performing Interference nulling for such clients leads to a considerable gain in total throughput of TCP flows per client by a factor of $1.7\times$.

One might wonder why the reduction in INR in commodity cards using Interference nulling is below that of typical software radios [98, 63]. This is because, the channel state information reported by 802.11 is only provided for 30 OFDM sub-carriers, and each 3×3

transmit beamforming matrix is allotted only 24 bits [2]. While these settings reduce the overhead from explicit feedback, they also limit the INR reduction possible. However, as shown above, this reduction suffices to enable big gains for real WLANs.

Experiment 2: (Coherent Beamforming) In this experiment, we consider scenarios where the controller performs coherent beamforming from an AP to a single-antenna client. For each set of client locations, we measure the increase in SNR at the client, with 802.11 and OpenRF's beamforming. We also measure the total throughput of TCP flows to this client with OpenRF and standard 802.11.

Results 2: The plot in Fig. 3-12(b) shows the SNR from AP to client, with and without coherent beamforming. We observe that coherent beamforming provides a mean increase of 3 dB in the SNR of the desired signal. This is very significant, since we observed this was sufficient to enable 802.11's rate adaptation algorithm[69] to adapt to a higher rate compared to standard 802.11, under identical circumstances, in 80% of our experiments. In fact, our results for TCP flows, shown in Fig. 3-12(b) demonstrate a mean gain of 1.4× in the throughput of OpenRF's beamforming, compared to 802.11.

Besides the gain in SNR, the gain in throughput is facilitated by another interesting phenomenon. Quite often, some OFDM sub-carriers in the 802.11 channel profile of a single antenna experience low SNR due to fading. However, coherent beamforming from all three transmit antennas across sub-carriers ensures that fading of any single OFDM bin is extremely unlikely. Thus, channel profiles using diversity, are significantly more flat, when compared to standard 802.11. This is especially beneficial, since 802.11 uses the same modulation and coding scheme across OFDM bins.

Experiment 3: (Interference Alignment) In this experiment, we consider scenarios reported by the controller where a 2-antenna client is receiving its desired signal from its AP. However, the client also receives undesired interference from two other 3-antenna APs (Fig. 3-9). To mitigate this, we perform interference alignment, so that the two interfering APs align their signals along one of the client's antennas. This ensures that any residual interference on the client's other antenna is minimized. The client now decodes its desired signal, interference-free, from its other antenna (See §3.4.3).

For each client, we then measure the total INR received from both antennas, as well as how much residual INR leaked into the second antenna meant for the desired signal. We also evaluate the gain in the total throughput of the concurrent TCP flows made possible due to OpenRF's alignment, compared to standard 802.11.

Results 3: Fig. 3-12(c) plots the residual INR leaking into the direction of the desired signal, against the total INR. Our results show that the residual INR is 11 dB below the total INR. This is a significant reduction, as clients receiving signals at the boundary of three APs are likely to obtain low INRs from the interfering APs. Yet, these INRs are still comparable to the SNR of the desired signal. In fact, the total throughput of TCP flows for the client under OpenRF's Interference Alignment experiences a $1.5\times$ gain compared to standard 802.11.

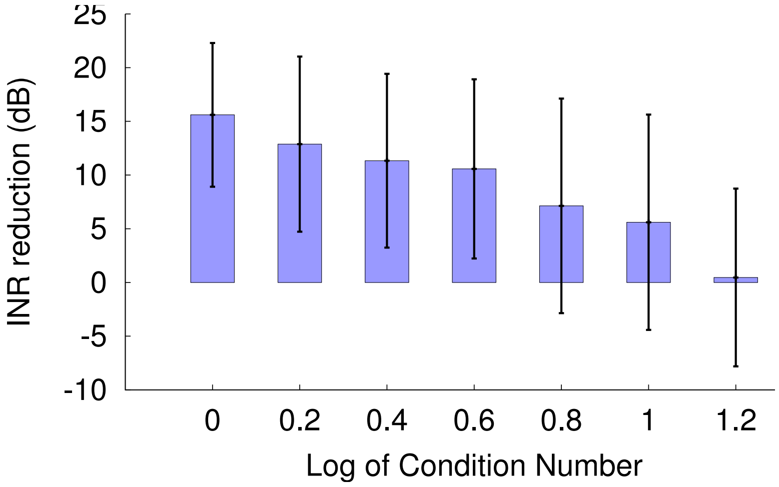


Figure 3-13: Effect of Channel on Interference Alignment. The gain of interference alignment is highly dependent on the condition number of the channel matrix. OpenRF only employs interference alignment when the log of the condition number is below 0.6.

Interestingly, OpenRF's controller algorithm does not apply Interference Alignment for multi-antenna clients in all scenarios resembling Fig. 3-9. In some cases, the controller may choose not to apply alignment, as the channel matrix from an interfering AP to the client was ill-conditioned. In fact, for any MIMO system to provide multiplexing gains, it is imperative that the channel matrix is well-conditioned [199]. This constraint naturally extends to interference alignment. Specifically, Fig. 3-13 plots the ratio of total and residual INR for different condition numbers of the channel. Note that when the log of the condition number exceeds 1.2, the total and residual INRs are nearly equal, providing virtually no gain. As OpenRF's controller has channel state information, it accounts for this by applying alignment only when channel matrices are well-conditioned, i.e. the log of the condition numbers are below 0.6.

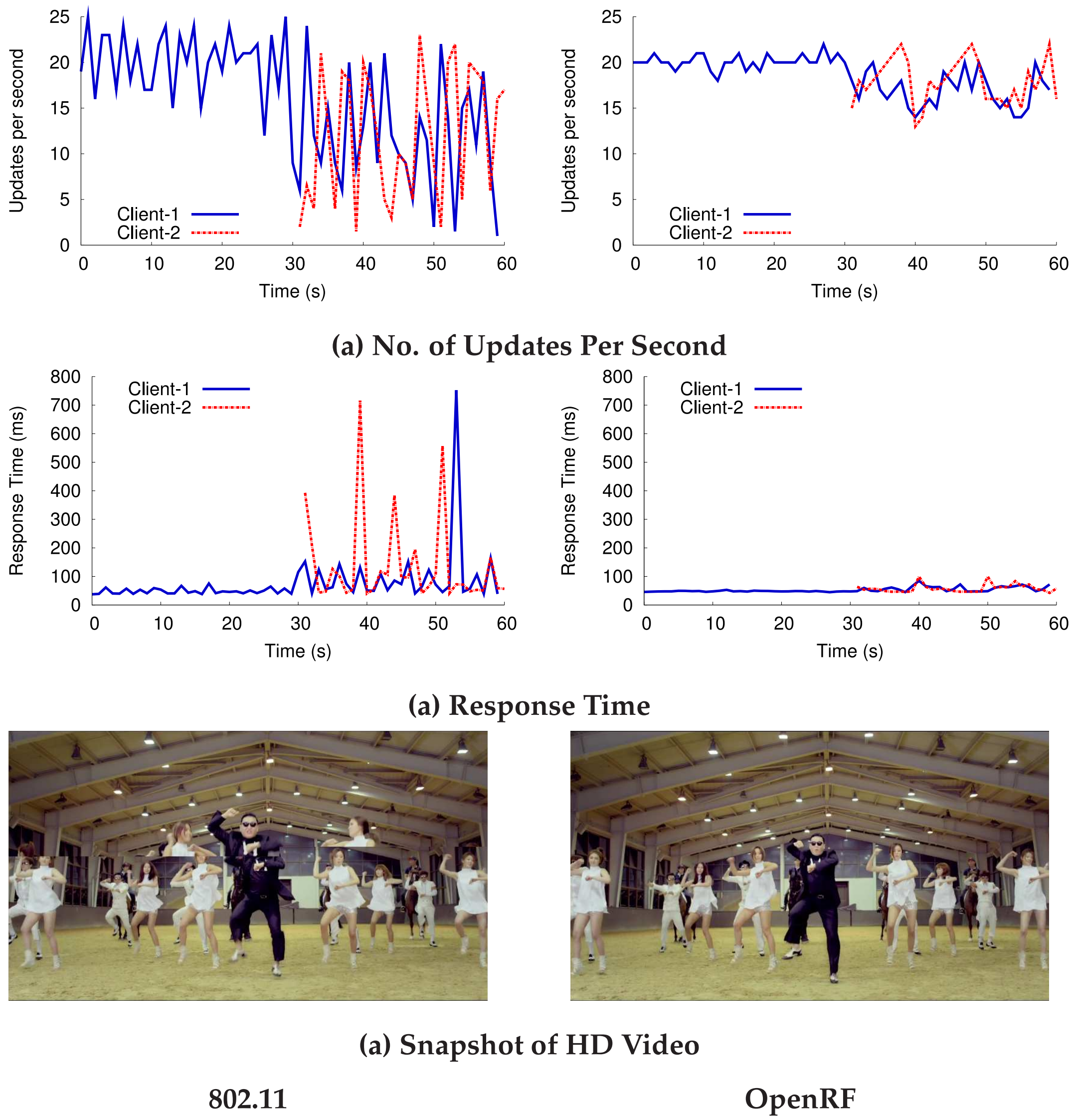
3.6.2 Application performance

In this experiment, we evaluate the impact of OpenRF on application performance. Specifically, we consider the problem of accessing rich multimedia applications over remote desktop, which is increasingly common in the enterprise.

Experiment. Once again, we consider scenarios reported by the controller where two single-antenna clients obtain signals from two APs as shown in Fig. 3-7. In this experiment, our clients run a remote desktop session over their wireless links, using VNC [84], a commonly used open-source platform for remote desktop. In particular we play a 1080p HD-video over remote desktop from the remote AP. We implement the VNC server on the access points to eliminate any potential delays or loss of throughput over the wired link. Our experiment proceeds as follows: We start a VNC session from client-1 to AP-1 at $t = 0$. We then start a parallel VNC session from client-2 to AP-2 at $t = 30$ seconds, and gather traces for both sessions until $t = 60$ seconds. We repeat this experiment under identical conditions for 802.11 and OpenRF. We measure the following quantities:

- *Updates per second*: The VNC protocol works by clients requesting updates for a certain number of pixels for a frame. The server responds with any changes to these pixels. In our experiment, we play a clip of Psy's Gangnam style, a rich and dynamic HD-video requiring frequent updates. We note that updates at a frequency of 15-20 per second are sufficient to avoid perceivable glitches.
- *Response delay*: Measures the mean delay between a request and its corresponding response. If the response time is high (over 200 ms), the video has visible outages.

Results. Fig. 3-14 plots the traces for the two clients for both OpenRF and 802.11. First, we notice that with 802.11, the VNC flow ceases to provide a comfortable remote desktop experience once the two APs transmit concurrently. In particular, due to heavy contention between the two large VNC flows, we observe that the number of updates per second, varies dramatically in the range of 2-22 updates per second for each flow, with a mean of 12.2. This means that the user experiences several glitches in the video. More importantly, we observe on several occasions that the response delay is over 200 ms, with the 90th percentile response delay being 270 ms. Thus, the user experiences frequent and visible outages in the remote desktop application.



(a) No. of Updates Per Second

(a) Response Time

(a) Snapshot of HD Video

802.11

OpenRF

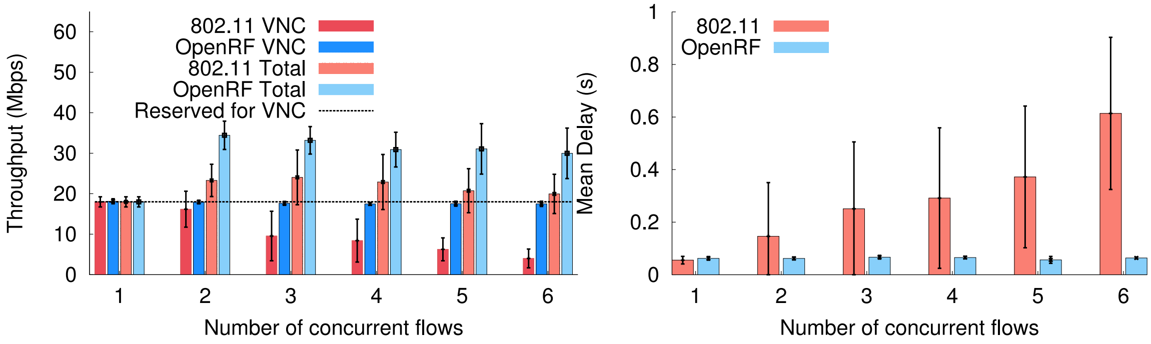
Figure 3-14: Performance of Video over Remote Desktop. Trace for 802.11 and OpenRF of: Row (1) - number of updates per second; Row (2) - response time(ms); Row (3) - snapshot of the HD video at the same frame.

In contrast, OpenRF provides updates for both clients in the range of 14-22 updates per second, with a mean of 17.6 for the clients, even when they are transmitting concurrently. This is because OpenRF benefits from PHY-aware cross-layer techniques which enable concurrent transmissions and therefore eliminate the need for contention for medium-sharing between the access points. As a consequence, OpenRF enhances user experience by ensuring fewer glitches[9] in the video by a factor of 6×, compared to standard 802.11. In addition, the 90th percentile delay of OpenRF is 67 ms, which is well below the required 200 ms, and is 4× lower than that of standard 802.11.

■ 3.6.3 Reservation

We study how OpenRF's reservation policy enhances application-level quality of experience.

Experiment. We consider scenarios reported by the OpenRF controller where a single-antenna client is at the edge of the communication range of the two APs, and therefore requires interference nulling from one of the APs. We reserve a throughput of 17 Mbps for a large VNC flow destined to this client from its APs. We then begin transmitting concurrent long-lived TCP flows, from either AP to up to five other clients in the network and report the VNC performance as a function of the number of concurrent flows.



(a) Throughput Reservation

(b) Response Delay

Figure 3-15: Reservation with VNC. Plots the mean and standard deviation of the following quantities, for OpenRF and 802.11, across number of concurrent flows: (a) Throughput of VNC flow and total network throughput (b) Response Delay of the VNC flow.

Results. The plot in Fig. 3-15(a) shows the throughput of the VNC flows, as well as the total network throughput, across the number of concurrent flows for both 802.11 and OpenRF. We observe that OpenRF reserves throughput at a mean value of 17.2 Mbps for the VNC flow, as required by the flow. More importantly, it does this while continuing to provide the same gains in total network throughput by exploiting PHY-aware cross layer

[9] We quantify glitches based on the number of times there were fewer than a fixed threshold of 14 updates per second.

techniques to enable concurrent transmissions of flows between the two APs.

In contrast, with 802.11, the performance of the VNC flow decays with an increasing number of contending flows. Furthermore, the network throughput is lower, as the APs do not manage interference, and therefore must share the medium between them.

In addition, Fig. 3-15(b) depicts the delay of the VNC flows, with 802.11 and OpenRF. As expected, OpenRF maintains the required user quality of experience by keeping VNC response delays at a low mean of 63 ms, despite concurrent transmissions, while with 802.11, the delays progressively deteriorate with an increasing number of concurrent flows.

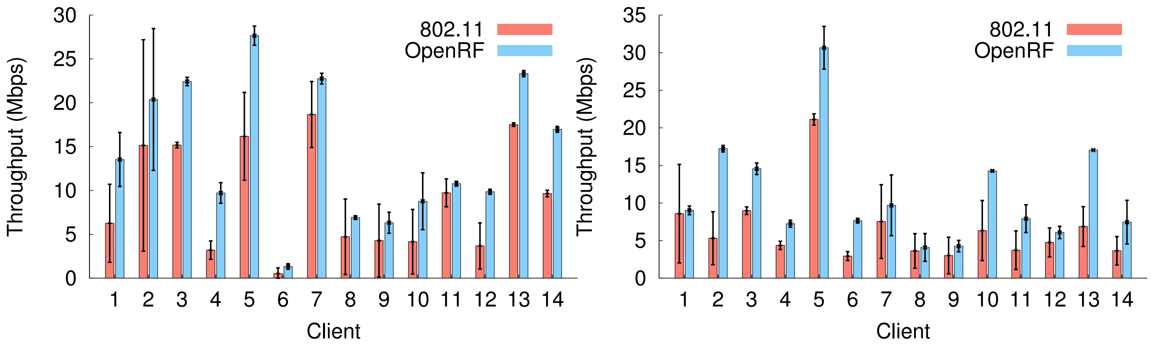
■ 3.6.4 Large scale measurement

Finally, we study how OpenRF performs under dynamic traffic patterns, dynamic channels, flow arrivals and departures for the full 20-node network.

Experiment. We place our AP and client nodes in randomly chosen locations (Fig. 3-11). Our clients are a combination of seven single-antenna, five 2-antenna and two 3-antenna nodes. We conduct our experiments with a variety of TCP and UDP best effort flows generated using iperf. In particular, our network allows a combination of TCP ($\approx$ 60 %) and UDP ($\approx$ 20 %) flows as well as uplink TCP and UDP traffic ($\approx$ 20 %).[10] UDP flows are long lived flows. TCP flows are generated using a mix of both long and short lived TCP flows. Each client has one long lived flow. Additionally, short flows are generated according to a Poisson arrival process, with mean inter-arrival time of 1s, and have a Pareto file size with mean of 125KB and shape parameter of 1.5. We assign TCP flows between uplink and downlink, according to the desired traffic ratios. We compare our system with standard 802.11 by replaying identical traffic patterns.

Results. Figure 3-16 plots the total throughput of all TCP and UDP flows per client, for 802.11 and OpenRF. In particular, we obtain a mean gain of $1.6\times$ for all TCP flows and $1.7\times$ for all UDP flows in the network. More importantly, we note that every client in our system achieves higher throughput on average, compared to 802.11. In fact, our results demonstrate that the OpenRF controller is truly self-configuring and correctly employs the right combination of PHY-aware techniques to suit dynamic network topologies and traffic patterns in real-time.

[10] These ratios are based on the traffic mix on our local network.



(a) TCP

(b) UDP

Figure 3-16: Performance. Plots the mean and standard deviation of throughput of TCP and UDP flows for all clients in the network

■ 3.7 Related Work

Related work falls into three categories:

Cross-Layer Designs: OpenRF is related to past work on the cross-layer wireless designs [193, 176, 134], particularly to systems that perform MIMO interference management, e.g., Interference Alignment and Cancellation [63], Beamforming [14], SAM [166], and others [141, 61, 98, 29]. However, these systems use techniques that focus on specific topologies or traffic patterns. In contrast, OpenRF is a general architecture that applies the right set of MIMO techniques to any topology or traffic pattern. Furthermore, unlike past work implemented on software radios, OpenRF leverages a modified 802.11n MAC, which employs carrier sense over interference and coherence slots to support MIMO techniques in today's wireless LANs. Thus, OpenRF tackles new challenges such as interference management in the presence of bursty TCP traffic, cooperative MIMO techniques across APs in an enterprise, compliance and integration with 802.11 standards, and implementation using commodity Wi-Fi cards.

Software Defined Networks: Our work is inspired by the large body of work designing software defined networks for wired LANs, including OpenFlow [116], and Ethane [23]. While we borrow the design principle of separating the data and control plane, unlike OpenFlow which remains at Layer-2, OpenRF provides MIMO interference management techniques that impact the physical layer. Such techniques cannot be built simply using OpenFlow.

Recently, there have been several proposals bringing software defined networking to wireless LANs. For e.g., OpenRoads [196], OpenWiFi [75] and Odin [164] abstract the higher layers of the network stack to provide applications such as seamless mobility, load balancing, etc. Our work is complementary to this work, and further enhances the control plane through MIMO interference management techniques.

Perhaps the closest related work to our system is OpenRadio[17], which provides a programmable data plane for wireless base stations supporting multiple technologies such as LTE, WiFi, or WiMAX. OpenRadio deals purely with managing the data plane on a single wireless device, by re-using physical layer blocks to implement different technologies. Our system complements OpenRadio by coordinating multiple wireless APs in a network to manage interference from the control plane.

Enterprise WLAN Architectures: DenseAP [123], Dyson [124], CENTAUR [154], and DIRAC [198] have proposed architectures to better manage enterprise wireless LANs. OpenRF complements this work by adding new MIMO cross-layer techniques to the toolbox available for managing enterprise WLANs.

Recent systems have proposed improving network management by budgeting frequency channels [143, 24], time [39], or leveraging antenna arrays [108]. In contrast, our system enables multiplexing shared spectrum across interfering APs through MIMO interference nulling and alignment. This complements the above schemes, enabling more efficient use of spectrum per channel. In addition, OpenRF can accommodate networks moving towards channel bonding [38] leading to larger chunks of channels shared across APs.

Finally, Trinity [155] proposes profiling users based on mobility and channel coherence for enabling distributed MIMO and diversity. However, unlike OpenRF, it uses custom hardware (WARP boards) and does not study how these PHY layer techniques interact with real-life traffic patterns and application-layer requirements.

■ 3.8 Discussion

This chapter introduces OpenRF, the first system that enables the deployment of physical-layer MIMO techniques on commodity Wi-Fi cards. It is also the first cross-layer design that is demonstrated using a fully operational network stack with real applications. While our current implementation of OpenRF demonstrates interference nulling, interference alignment and beamforming, we believe OpenRF can be extended to support other PHY-aware techniques such as multi-user MIMO and frequency diversity schemes [167], when future wireless cards support them.

Chapter 4
Interference Alignment By Motion

Recent advances in MIMO technology have demonstrated the practicality of interference alignment and its throughput gains [63, 166, 99, 87]. These systems leverage the MIMO capability of the transmitter to precode the signal and align it along a particular spatial direction at the receiver. In contrast, this chapter investigates a simple question: Can we perform interference alignment with single-antenna transmitters? Furthermore, can interference alignment be performed purely by the receiver without cooperation from the senders? A positive answer to these questions could extend the gains of interference alignment to scenarios with single antenna systems, such as sensors, hence improving the overall throughput of these networks. Furthermore, the ability to do alignment purely at the receiver eliminates the need to send feedback from the receiver to the senders to inform them about the direction along which they should align their signals.

Motivated by the above questions, we investigate whether a receiver can perform interference alignment by simply adjusting the position of one of its antennas. Our intuition is that this may be possible due to two reasons. First, indoor settings are rich in multipath. Hence, at any point in space, the perceived channel is a combination of many channels that correspond to the various paths that the signal traverses. Even a small shift in the antenna location would cause some of these paths to become shorter and others longer, leading to a significant change in the resulting channel. Second, channels are continuous functions over space.[1] This means that the receiver can gradually slide its antenna, measure

[1] The channel is a linear combination of multiple continuous waveforms [172], and, hence, is continuous

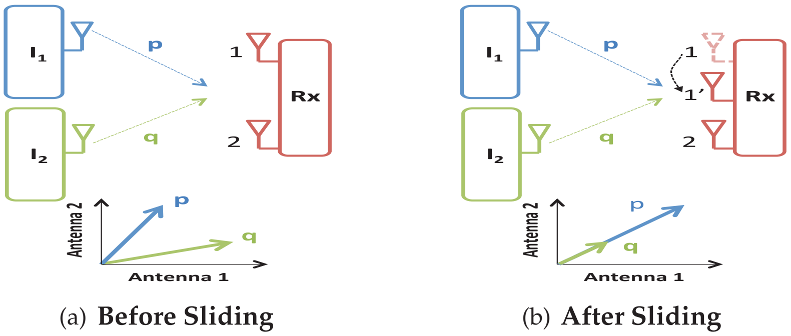


Figure 4-1: Interference Alignment by motion. Consider two single-antenna interferers I_1 and I_2 and a two-antenna receiver Rx. Initially, the two interferers are not aligned in the antenna space of Rx. Then, Rx slides one of its receive antennas so that I_1 and I_2 are aligned in the antenna space.

the channels, and keep moving the antenna in the direction that increases the alignment between the senders.

To test this intuition, we used USRP radios to experiment with the scenario in Fig. 4-1. In this setup, we have two single-antenna interferers and a two-antenna receiver. We mount one of the receive antennas on an iRobot Create robot to emulate a sliding antenna. We measure the channels and gradually adjust the position of one of the receive antennas, moving it in the spatial direction that increases the alignment between the interferers, as in Fig. 4-1(b). The experiments have confirmed our intuition: alignment can indeed be performed by sliding the receiver's antenna. What is perhaps even more surprising is that the required antenna displacement was typically an inch or less. Multiple experiments in two campus buildings in both line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios showed that in 84% of the settings, the required antenna displacement was less than one inch. The implications of our results are two-fold: First, they show for the first time that interference alignment can be achieved via motion. Second, the fact that the required displacement is of the order of one inch, means that motion-based interference alignment can be incorporated into recent sliding antennas available on the market.[2]

To gain insight into these empirical results, we have extended past signal propagation models to incorporate the effects of multipath on interference alignment. Our results reveal that the presence of a reflector near the receiver creates significant spatial diversity, causing the channel to change dramatically (from combining constructively to combining

over space and time.

[2] Recent USB Wi-Fi adapters have sliding antennas whose positions can be manually adjusted to improve the SNR [174].

destructively) within an inch or two. Since the receiver is typically situated on some platform which serves as a reflecting surface (e.g., a table, the floor, etc.), it is likely that it experiences reflections from at least one nearby reflector, enabling motion based interference alignment.

We introduce MoMIMO, a technique that enables MIMO interference management through mobility. Besides motion-based interference alignment, our study of MoMIMO reveals several other interesting capabilities:

- First, MoMIMO also provides interference nulling. Specifically, a single-antenna receiver can slide its antenna position to cancel an unwanted interference signal from a single-antenna transmitter, hence enabling single-antenna nodes to exploit MIMO nulling.
- Second, the process of sliding antennas to seek positions of interference alignment or nulling can be automated using a Stochastic Hill Climbing algorithm, as described in §4.5.
- Third, both uplink and downlink traffic benefit from motion-based MIMO techniques. This stems from the reciprocity of wireless channels, which ensures that motion-based alignment or nulling on the downlink, causes interference alignment or nulling on the uplink (see §4.6).

We implemented MoMIMO on an indoor network of USRP N210 software radios. To emulate a sliding antenna, we mounted the antennas on an iRobot Create robot. We conducted our experiments at a bandwidth of 20 MHz in the 2.4 GHz Wi-Fi band, both in line-of-sight and non-line-of-sight scenarios. Our results show the following:

- MoMIMO reduces the interference-to-noise ratio (INR) of undesired signal by an average of 22 dB for interference alignment and 15 dB for interference nulling.
- In a network of multiple transmitter-receiver pairs that use a combination of alignment and nulling, MoMIMO achieves an average throughput gain of $1.98\times$ over 802.11n in networks with both single-antenna and multi-antenna nodes.
- MoMIMO's Stochastic Hill Climbing algorithm requires a mean displacement of the receive antenna of 0.44 inches for interference alignment and 1.17 inches for nulling.[3]

[3] Alignment is easier to satisfy because many options exist for the direction along which two senders may be aligned, which provides extra flexibility for the choice of channels.

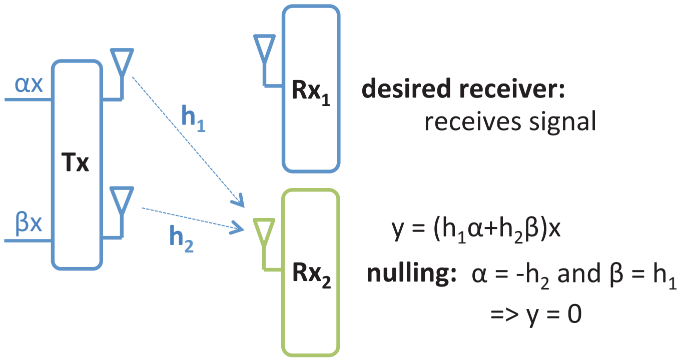


Figure 4-2: Interference Nulling. The transmitter cancels its signal at Rx_2 by precoding it, while Rx_1 can still receive its signal x.

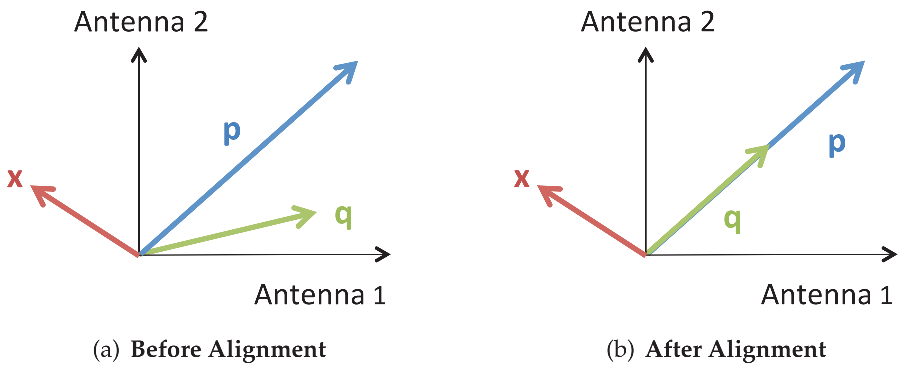
- Due to reciprocity, MoMIMO's interference alignment and nulling on the downlink also reduces the INR on the uplink to below the noise floor (i.e., the mean INR on the uplink is below 0 dB).
- MoMIMO is robust to sudden channel changes in otherwise static environments. It adopts a fail safe mechanism that falls back to 802.11n if the channel changes significantly, and recovers its gains once the channel is stable.

Contributions. This chapter presents three main contributions: (1) It is the first to demonstrate the feasibility of interference alignment and nulling purely via motion. Furthermore, it reveals that the required displacement is about one inch, and hence can potentially be supported by sliding antennas similar to those used in recent products [174]. (2) It presents a stochastic hill climbing algorithm that automatically repositions the antenna to achieve interference alignment or nulling. (3) We implements our design and demonstrate large throughput gains over real wireless channels.

■ 4.1 Background

We briefly introduce interference nulling and alignment.

(a) Interference Nulling: Interference nulling enables a MIMO node to transmit signals, and at the same time cancel them out at particular receiver (other than its own). Consider the example in Fig. 4-2; it consists of a two-antenna transmitter Tx and two single-antenna receivers Rx_1 and Rx_2. Let h_1 and h_2 denote the wireless channels from the two transmit antennas of Tx to the single antenna of Rx_2. Suppose the transmitter wants to send a signal



(a) **Before Alignment** (b) **After Alignment**

Figure 4-3: Interference Alignment. q's transmitter performs precoding in order to align q along the same direction as p in the antenna space of the receiver.

x to Rx_1 without interfering at Rx_2. The transmitter can then send αx on its first antenna and βx on its second antenna. Then, Rx_2 would receive: $y = (\alpha h_1 + \beta h_2)x$. Hence, the transmitter can null its signal to this receiver, simply by picking $\alpha = -h_2$ and $\beta = h_1$. This process of encoding the transmitted signal across multiple antennas (i.e. multiplying by α and β) is referred to as *precoding*, and is used in both nulling and alignment.

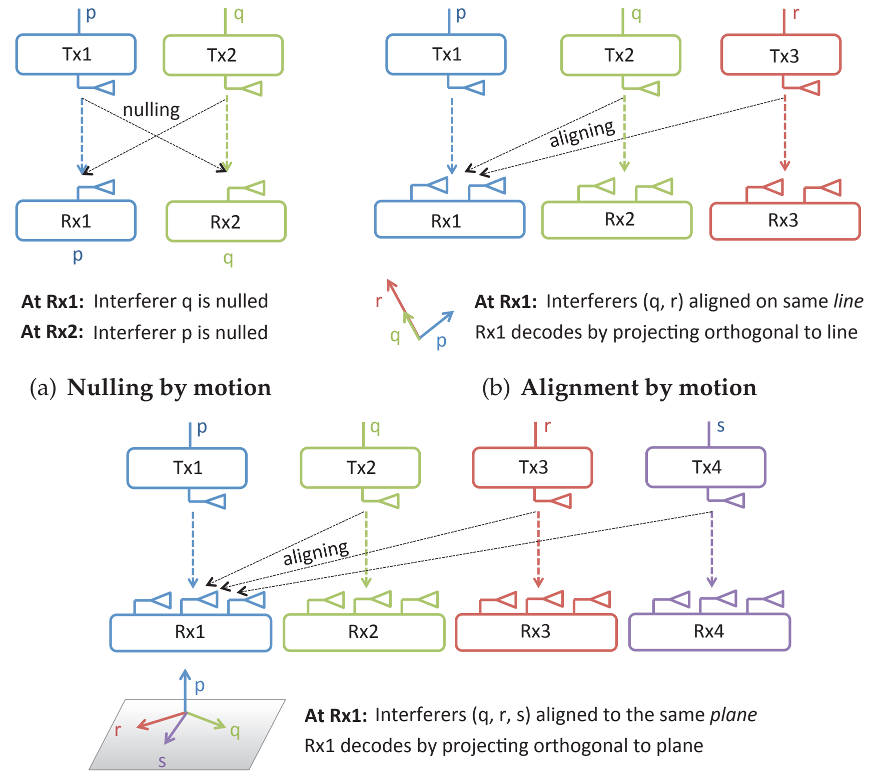
(b) Interference Alignment: Recall the following basic properties of MIMO systems [172, 99]:

- An N-antenna receiver receives signals on each of its N antennas. These signals can be represented as one vector in an N dimensional space.
- An N-antenna receiver can decode up to N independent signals.

Suppose a 2-antenna receiver receives two signals – a desired signal x and an interfering signal p. As mentioned above, these signals can be represented as vectors in a 2-D space, and the MIMO receiver can decode them. However, let's say another interferer joins the network, so that the 2-D space has a third vector q, as in Fig. 4-3(a). Unfortunately, the receiver can now no longer decode. So, how can the transmitter of q avoid interfering at the receiver?

The transmitter of q can precode its transmission to rotate the vector q and align it along the same direction as p; this technique is called *interference alignment*.[4] In effect, the receiver now obtains only two vectors, one vector is the desired signal x, and the other is the sum of interferences p and q, as shown in Fig. 4-3(b). Now, the receiver can eliminate

[4] Specifically, to align the vector q along the vector p, the transmitter uses a precoding matrix M, such that, $p^{\perp}qM = 0$, where $p^{\perp}$ denotes the vector space orthogonal to p [99, 63].



At Rx1: Interferer q is nulled
At Rx2: Interferer p is nulled

At Rx1: Interferers (q, r) aligned on same *line*
Rx1 decodes by projecting orthogonal to line

(a) **Nulling by motion**

(b) **Alignment by motion**

At Rx1: Interferers (q, r, s) aligned to the same *plane*
Rx1 decodes by projecting orthogonal to plane

(c) **Scaling Motion-based Alignment**

Figure 4-4: MoMIMO enables concurrent transmissions. The figure shows three examples: **(a) Nulling:** Rx1 slides its antenna to null interference from Tx2, and Rx2 slides its antenna to null interference from Tx1. Hence, the network supports 2 concurrent transmissions. **(b) Alignment:** Rx1 slides its antenna to align interference from Tx2 and Tx3. Similarly, Rx2 and Rx3 slide their antennas to align their interferers. Thus, the network can support 3 concurrent streams. **(c) Scaling Alignment:** Rx1 slides one of its antennas to align interference from Tx2, Tx3 and Tx4. Similarly, Rx2, Rx3, and Rx4 each slide one of their antennas to align their interferers. Thus, the network supports 4 concurrent streams.

all interference by projecting the received signal on a direction orthogonal to the direction of the aligned interference.

■ 4.2 Scope of MoMIMO

MoMIMO enables wireless nodes to perform techniques such as interference alignment and nulling by sliding one of their antennas by an inch or two. MoMIMO applies to both single antenna and multi-antenna nodes, i.e., MoMIMO enables both single and multi-antenna nodes to manage interference in a manner that increases the number of concurrent streams beyond what is available purely by the number of antennas on these nodes.

MoMIMO is not meant for mobile or hand-held devices, but is instead targeted for relatively static environments, with stable interference patterns. Examples of such scenarios include: (1) A home Wi-Fi access point that suffers interference from the neighboring apartment. (2) A sensor network collection point suffering from interference. (3) Wireless as a replacement for wires at homes and offices, e.g., a Wi-Fi link used to connect a TV to a DVD player [158], wireless surveillance cameras [22], etc. (4) Flyways in Data centers [68].

■ 4.3 MoMIMO in a Network

We begin by presenting examples that illustrate how MoMIMO can provide significant throughput gains for wireless networks. We explain when nodes may use motion-based alignment and when they use motion-based nulling. For this section, we assume that it is possible to find locations that satisfy the desired interference alignment or interference nulling, by using small position adjustments of an inch or two. We will validate this assumption both empirically in §10.5 and analytically in §4.4.

■ 4.3.1 Motion-Based Interference Nulling

Consider two co-located single-antenna transmitter-receiver pairs: Tx1-Rx1 and Tx2-Rx2 (Fig. 4-4(a)). Say these nodes are initially at positions where they interfere strongly (i.e. the interference to noise ratio (INR) from Tx1 to Rx2 and Tx2 to Rx1 is high). In such a scenario, 802.11 permits only one of Tx1 and Tx2 to transmit at any given time, i.e., the maximum number of concurrent streams is one.

Recall that because Tx1 and Tx2 are single-antenna transmitters, they cannot perform standard interference nulling to each other's receivers. This is because standard interference nulling requires a node to precode its signal across *multiple transmit antennas* as discussed in §7.1.

In contrast, MoMIMO can leverage motion-based nulling to send two concurrent streams that do not interfere with each other. Specifically, each receiver makes small adjustments to its antenna position by one or two inches until the signal from the interfering transmitter is nulled (i.e. below the noise floor). In this example, Rx1 adjusts its antenna until it nulls Tx2's interference. Similarly, Rx2 adjusts its antenna until it nulls Tx1. Now that there is no interference between the two streams, Tx1 and Tx2 can transmit concurrently to their respective receivers.

Reciprocity. We now have the receive antennas at positions where Tx1 and Tx2 can transmit concurrently, without interference. But what happens when Rx1 and Rx2 need to transmit their acknowledgments concurrently to Tx1 and Tx2? Interestingly, we notice that there is no need for any new adjustments in antenna positions, i.e. the ACKs do not interfere at Tx1 and Tx2. This is because reciprocity in wireless channels also leads to reciprocity in motion-based nulling (details in §4.6). Thus, once receivers perform motion-based nulling, they not only receive their desired signals concurrently, but also deliver ACKs to their transmitters, interference-free.

Multiplexing Gains. MoMIMO's motion-based interference nulling, can enable two concurrent streams in this setting, i.e., a $2\times$ multiplexing gain over 802.11.

■ 4.3.2 Motion-Based Interference Alignment

Consider three co-located transmitter-receiver pairs: Tx1-Rx1, Tx2-Rx2 and Tx3-Rx3, as shown in Fig. 4-4(b). The transmitters have only one antenna each, while the receivers are two-antenna nodes. Once again, these nodes are in positions where they interfere strongly with each other. Therefore, 802.11 permits only one of Tx1, Tx2, and Tx3 to transmit packets at a given point in time; hence, the maximum number of concurrent streams is one.

Recall that since Tx1, Tx2, and Tx3 have only one antenna each, they cannot align their interfering signals along a common spatial direction at each receiver. This is because such alignment requires the transmitter to precode its signal across *multiple transmit antennas* as explained in §7.1.

In contrast, MoMIMO leverages motion-based alignment to send three concurrent streams that do not interfere with each other. Let's consider the two-antenna receiver Rx1, which receives its desired signal from Tx1 and unwanted interference from both Tx2 and Tx3. To mitigate this interference, Rx1 performs motion-based alignment by adjusting one of its antennas so that the signals from its two interfering transmitters, Tx2 and Tx3, are aligned in its 2-D space (see Fig. 4-4(b)). Now Rx1 obtains only two signal vectors in this 2-D space: the signal from its desired transmitter (Tx1) and the sum of the interference from the unwanted transmitters Tx2 and Tx3. Thus, as in standard interference alignment (see §7.1), the receiver can eliminate all interference by projecting its received signal along a direction orthogonal to that of the aligned interference. Similarly, Rx2 can decode its

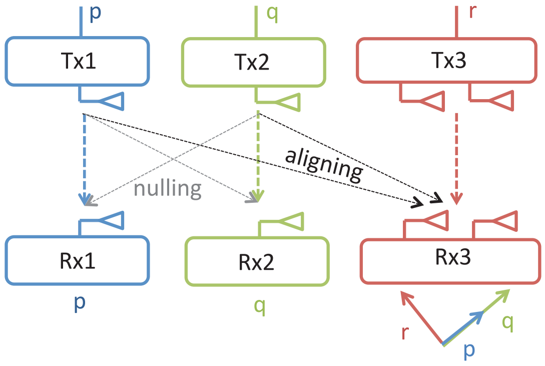


Figure 4-5: MoMIMO in heterogeneous networks. The two single antenna pairs perform nulling as in Fig. 4-4(a). Rx3 slides one of its antennas to align Tx1 and Tx2; similarly, Tx3 slides one of its antennas to align ACKs from Rx1 and Rx2. Tx3 also precodes its signal to null at Rx1. The network can support 3 concurrent streams.

desired signal concurrently by adjusting one of its antennas to align interference from Tx1 and Tx3; while, Rx3 adjusts one of its antennas to align interference from Tx1 and Tx2.

Reciprocity. We now have the receive antennas at positions where Tx1, Tx2 and Tx3 can transmit concurrently without interference. Suppose the receivers want to send ACKs back to their respective transmitters, concurrently. These ACKs may still interfere with each other at the transmitters. For example, Rx1's ACK would cause interference at Tx2 and at Tx3. To address this issue, Rx1 can precode its transmission using standard interference nulling, such that it is nulled at Tx2. Fortunately, nulling the ACK at Tx2 also automatically nulls the ACK at Tx3 as well. This desirable phenomenon is due to channel reciprocity and will be further explained in §4.6. Hence, Rx2 and Rx3 can perform a similar procedure so that the receivers concurrently deliver ACKs to their transmitters, interference-free.

Multiplexing Gains. MoMIMO's motion-based interference alignment can enable three concurrent streams in this setting, i.e., a $3\times$ gain over 802.11. We note that even if the receivers employed uplink multi-user MIMO (as in [166]) instead of standard 802.11, the maximum number of concurrent streams would be two, still $1.5\times$ less than that of MoMIMO.

Scaling. MoMIMO's interference alignment scales beyond the setting in Fig. 4-4(b). For e.g., in a setup with four single-antenna transmitters and four 3-antenna receivers (shown in Fig. 4-4(c)), it can enable four concurrent streams, i.e., a $4\times$ gain over 802.11.

■ 4.3.3 Combining Alignment and Nulling

MoMIMO's throughput gains can be generalized to heterogeneous networks that require combining both motion-based alignment and motion-based nulling. Consider the scenario in Fig. 4-5, with two pairs of single-antenna nodes, and one pair of two-antenna MIMO nodes. Note that in 802.11n, only one of Tx1, Tx2, and Tx3 can transmit packets at any given time.

In contrast, MoMIMO can exploit both motion-based alignment and nulling to deliver three concurrent streams. To begin with, recall that Tx1, Tx2, Rx1 and Rx2 form a topology similar to the one in Fig. 4-4(a). Thus, to enable concurrent transmissions, Rx1 can simply slide its antenna to null Tx2, and Rx2 can similarly slide its antenna to null Tx1, as in §4.3.1.

We still need to make sure that Tx3 can transmit concurrently without interfering with the ongoing transmissions. To this end, we exploit interference alignment, as in §4.3.2. Specifically, Rx3 slides one of its antennas to *align* Tx1 and Tx2's signals. In addition, Rx3 precodes its transmission using standard interference nulling, such that it is nulled at Tx1; by reciprocity, Rx3's transmission is automatically nulled at Tx2 as explained in §4.6. Similarly, Tx3 slides one of its antennas to *align* the ACKs from Rx1 and Rx2; it also precodes its transmission such that it is nulled at Rx1 (and, hence, Rx2).

Multiplexing Gains. In this setting, 802.11n can send 1.33 concurrent streams on average assuming fair time-sharing between streams.[5] In contrast, MoMIMO's motion-based alignment and nulling enables three concurrent streams, i.e. a $2.25\times$ gain over 802.11n. Note that using interference alignment and nulling without motion in this setup will only enable two concurrent streams at any time. Hence, MoMIMO can still achieve a $1.5\times$ gain over advanced techniques that leverage standard nulling and alignment, e.g. n+ [99].

Finally, we note that in this section, we have assumed that it is feasible to find nearby locations that achieve the desired alignment and nulling. In §10.5, we revisit these topologies and provide extensive measurements that span line-of-sight and non-line of sight scenarios in multiple buildings, all of which confirm the gains of MoMIMO.

[5] In any time slot, Tx1, Tx2 or Tx3 sends 1, 1, or 2 streams respectively. Thus, on average, 802.11n sends $\frac{1+1+2}{3} = 1.33$ streams.

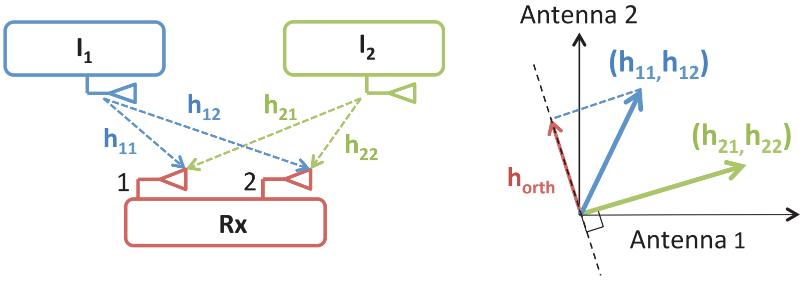


Figure 4-6: Interference Alignment. We wish to align the signals received from the two interferers I_1 and I_2 along the same direction. The goodness of alignment is determined by the *residual interference*; it is the projection of an interferer along a direction orthogonal to the desired alignment, and denoted as h_{orth}.

■ 4.4 MoMIMO Analysis

In this section, we extend signal propagation models to capture how multipath effects enable motion-based alignment. We also give insights for why the required displacement is relatively small and is about one or two inches.

In order to quantify alignment, we need to pick a metric that reflects the goodness of alignment. So, how do we choose this metric? Suppose, a receiver Rx wants to align a given interferer I along a particular direction $\vec{v}$ in its antenna space. If the interferer is perfectly aligned with $\vec{v}$, then its projection along $\vec{v}^{\perp}$ (i.e. the direction orthogonal to $\vec{v}$) is zero. In contrast, if the interferer is poorly aligned with $\vec{v}$, its projection on $\vec{v}^{\perp}$ is large. In fact, the larger this projection is, the worse the alignment is. In the rest of this section, we analytically quantify the goodness of alignment in terms of *residual interference* after alignment [64, 136]. This quantity, which we denote h_{orth} is defined as the projection of the interference along the direction orthogonal to that of the desired alignment.

Consider the setup in Fig. 4-6, where we wish to slide antenna 1 of the receiver Rx to align the signal $\vec{h}_1$ from I_1 along the same direction as the signal $\vec{h}_2$ from I_2. Let h_{tr} denote the channel from interferer I_t to receive antenna r. We then write the residual interference after alignment h_{orth} as:

$$\begin{aligned} h_{orth} &= \vec{h}_1.\vec{h}_2^{\perp} && (4.1)\\ &= (h_{11}, h_{12}).(-h_{22}, h_{21}) && (4.2)\\ &= -h_{11}h_{22} + h_{12}h_{21} && (4.3) \end{aligned}$$

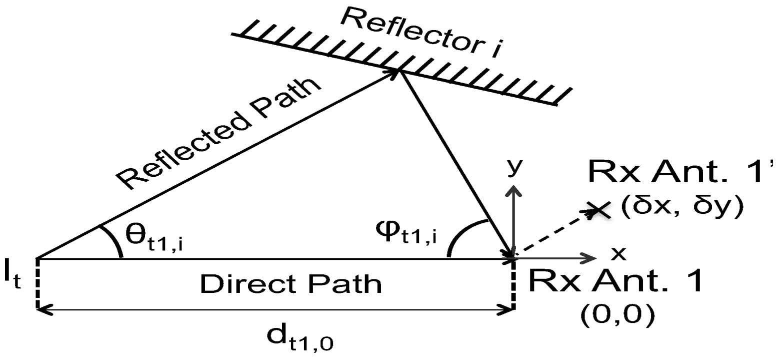


Figure 4-7: Specifying Reflectors. We specify the location of reflector i with respect to interferer I_t by: $\theta_{t1,i}$, the angle of departure of the signal from interferer I_t and $\phi_{t1,i}$, the angle of arrival of the reflected signal at receive antenna 1.

where $\vec{h}_2^{\perp}$ denotes the vector orthogonal to $\vec{h}_2$.[6]

Let's study how the value of these channels h_{tr} and the resulting h_{orth} change as a function of reflectors in the environment. Say that we have n reflectors in the environment. Then the channel between any interferer I_t and receive antenna r is the sum of the channel along the direct path between the two, $h_{tr,0}$, plus the channels along all the reflected paths, $h_{tr,i}$, for all reflectors $i = 1, \dots, n$. We can use standard propagation models [172] to express these channels as a function of path length. Specifically, the channel on the direct path $h_{tr,0}$ is:

$$h_{tr,0} = \frac{1}{d_{tr,0}} e^{-2\pi j \frac{d_{tr,0} f}{c}} \tag{4.4}$$

where $d_{tr,0}$ is the length of the direct path between interferer I_t and receive antenna r, c is the speed of light, and f is the frequency of the transmitted signal.

Similarly, the channel along the i^{th} reflected path as:

$$h_{tr,i} = \frac{\gamma_i}{d_{tr,i}} e^{-2\pi j \frac{d_{tr,i} f}{c} + j\pi} \tag{4.5}$$

where γ_i is the reflection coefficient of the reflector, and $d_{tr,i}$ is the total distance traversed by the reflected signal – i.e., $d_{tr,i}$ is the sum of the two segments of a reflected path, the segment from the interferer I_t to the reflector and the segment from the reflector to the receive antenna r.

[6] Note that, in practice, $\vec{h}_2^{\perp}$ is a unit vector. We ignore normalization of $\vec{h}_2^{\perp}$ for simplicity.

Thus, the channel from interferer I_t to receive antenna r is:

$$\begin{aligned}
h_{tr} = h_{tr,0} + \sum_{i=1}^{n} h_{tr,i} &= \frac{1}{d_{tr,0}} e^{-2\pi j \frac{d_{tr,0} f}{c}} + \sum_{i=1}^{n} \frac{\gamma_i}{d_{tr,i}} e^{-2\pi j \frac{d_{tr,i} f}{c} + j\pi} \\
&= \sum_{i=0}^{n} \frac{\gamma_i}{d_{tr,i}} e^{-2\pi j \frac{d_{tr,i} f}{c} + \alpha_i}
\end{aligned}$$

Where $\gamma_0 = 1$, $\alpha_0 = 0$, and $\alpha_i = j\pi$ for $i = 1, \ldots, n$.

We can substitute these channels in Eqn. 4.3 that defined h_{orth} to understand how reflectors impact the residual interference after alignment. As we substitute, remember that we are only sliding antenna 1 on the receiver, while antenna 2 is fixed. Therefore, we consider channels h_{12} and h_{22} to antenna 2 to be constants, for the rest of our analysis.

$$\begin{aligned}
h_{orth} &= -h_{22} h_{11} + h_{12} h_{21} \\
&= -h_{22} \sum_{i=0}^{n} \frac{\gamma_i}{d_{11,i}} e^{-2\pi j \frac{d_{11,i} f}{c} + \alpha_i} + h_{12} \sum_{i=0}^{n} \frac{\gamma_i}{d_{21,i}} e^{-2\pi j \frac{d_{21,i} f}{c} + \alpha_i} \\
&= \sum_{t=\{1,2\}} \sum_{i=0}^{n} \frac{\gamma_{t1,i}}{d_{t1,i}} e^{-2\pi j \frac{d_{t1,i} f}{c} + \alpha_i}
\end{aligned}$$

Where $\gamma_{11,i} = -h_{22}\gamma_i$ and $\gamma_{21,i} = h_{12}\gamma_i$, are constants.

The above equation computes the residual interference after alignment for a particular position of the receive antenna. We are interested in how h_{orth} changes as the receive antenna slides by $(\delta x, \delta y)$, which we denote by $h_{orth}(\delta x, \delta y)$. To derive $h_{orth}(\delta x, \delta y)$, we express the location of reflector i based on two angles for any interferer I_t and receive antenna r: the angle of departure ($\theta_{tr,i}$) and angle of arrival ($\phi_{tr,i}$), as shown in Fig. 4-7. Given these values, we can show that $h_{orth}(\delta x, \delta y)$ is:

$$h_{orth}(\delta x, \delta y) = \sum_{t=\{1,2\}} \sum_{i=0}^{n} \frac{\gamma_{t1,i}}{\tilde{d}_{t1,i}} e^{-2\pi j \frac{\tilde{d}_{t1,i} f}{c} + \alpha_i} \qquad (4.6)$$

Where $\tilde{d}_{t1,i}$ is given by:

$$\tilde{d}_{t1,i} = \sqrt{(2c_{t1,i}\text{sin}^2\beta_{t1,i} - d_{t1,0} - \delta x)^2 + (c_{t1,i}\text{sin}(2\beta_{t1,i}) - \delta y)^2} \tag{4.7}$$

$$c_{t1,i} = \frac{d_{t1,0}\text{sin}\phi_{t1,i}}{\text{sin}\phi_{t1,i} - \text{sin}\theta_{t1,i}} \tag{4.8}$$

$$\beta_{t1,i} = (\phi_{t1,i} - \theta_{t1,i})/2 \tag{4.9}$$

The proof for the above equations follows from the geometry of the reflectors in Fig. 4-7. It is provided below:

Proof of Eqn. 4.7: In this section, we derive the length of reflected paths from Eqn. 4.7 given the position of an arbitrary reflector. Recall from §4.4 that any reflector i can be specified in terms of two angles $\theta_{t1,i}$ and $\phi_{t1,i}$. Consider a transmitter I_t and receive antenna R_1 separated by a distance $d_{t1,0}$. We can write the position of the reflector in terms of $c_{t1,i}$, its distance from the transmitter, and $\beta_{t1,i}$ its angle to the axis I_tR_1 (Figure 4-8). First, it is easy to see that $\beta_{t1,i} = \frac{\phi_{t1,i} - \theta_{t1,i}}{2}$. Now, by using sine-rule on ΔPI_tR_1, we have:

$$PI_t = d_{t1,0}\frac{sin(\phi_{t1,i})}{sin(\phi_{t1,i} + \theta_{t1,i})} \tag{4.10}$$

Using sine rule on triangle PI_tQ:

$$\begin{aligned} c_{t1,i} &= PI_t\frac{sin\left(\frac{\phi_{t1,i}+\theta_{t1,i}}{2}\right)}{sin\left(\frac{\phi_{t1,i}-\theta_{t1,i}}{2}\right)} = \frac{d_{t1,0}sin(\phi_{t1,i})sin\left(\frac{\phi_{t1,i}+\theta_{t1,i}}{2}\right)}{sin(\phi_{t1,i}+\theta_{t1,i})sin\left(\frac{\phi_{t1,i}-\theta_{t1,i}}{2}\right)} \\ &= \frac{d_{t1,0}sin(\phi_{t1,i})}{2cos\left(\frac{\phi_{t1,i}+\theta_{t1,i}}{2}\right)sin\left(\frac{\phi_{t1,i}-\theta_{t1,i}}{2}\right)} = \frac{d_{t1,0}sin(\phi_{t1,i})}{sin(\phi_{t1,i}) - sin(\theta_{t1,i})} \end{aligned}$$

Now suppose the receive antenna R_1 is initially at the origin (0,0). Then the transmitter is at a coordinate (-$d_{t1,0}$, 0). Let us denote I'_t as the symmetric point of the transmitter about the reflector. Note that since PI_t = PI'_t, $PI_t + PR_1 = PI'_t + PR_1 = I'_tR_1$. Hence the length of the reflected path to any receive antenna coordinate R_1 is I'_tR_1.

Clearly $\angle I'_tI_tR_1 = \pi/2 - (\frac{\phi_{t1,i}-\theta_{t1,i}}{2}) = \pi/2 - \beta_{t1,i}$. From ΔQI_tS we have $SI_t = SI'_t = csin\beta_{t1,i}$. Hence $I'_tI_t = 2c_{t1,i}sin\beta_{t1,i}$. Hence the coordinate of I'_t is ($2c_{t1,i}sin^2\beta_{t1,i} - d_{t1,0}$, $2c_{t1,i}sin\beta_{t1,i}cos\beta_{t1,i}$) = ($2c_{t1,i}sin^2(\beta_{t1,i}) - d_{t1,0}$, $csin(2\beta_{t1,i})$).

Thus the length of the reflected path when the receive antenna is moved to any position (δx, δy) is:

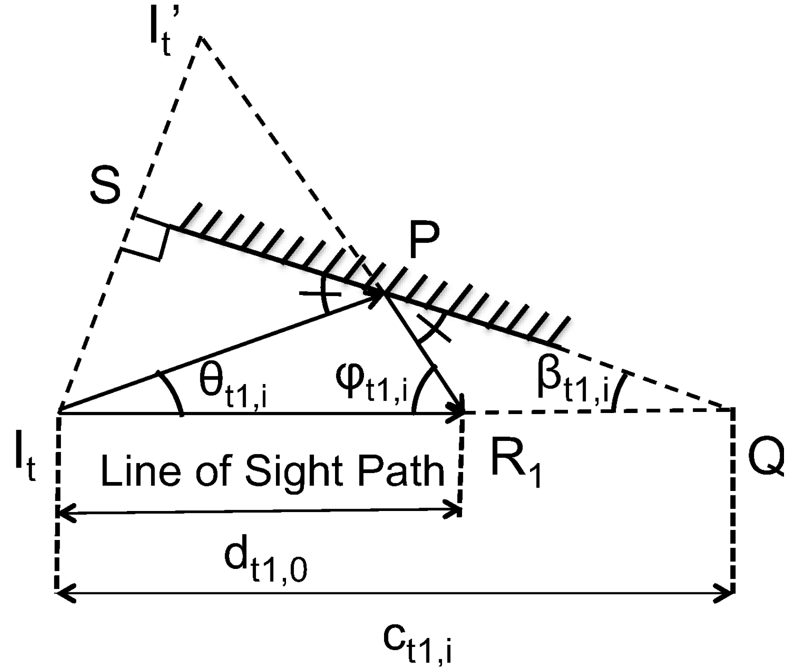


Figure 4-8: Reflected Path. We consider a transmitter I_t and receive antenna R_1. Signals arrive at R_1 from the direct line-of-sight path and a path due to reflector specified by the tuple $(\theta_{t1,i}, \phi_{t1,i})$, where $\theta_{t1,i}$ denotes the angle of departure of the signal from interferer I_t and $\phi_{t1,i}$ denotes the angle of arrival of the signal at receive antenna R_1

$$d'_{t1,i} = \sqrt{(2c_{t1,i}\sin^2(\beta_{t1,i}) - d_{t1,0} - \delta x)^2 + (c_{t1,i}\sin(2\beta_{t1,i}) - \delta y)^2}$$

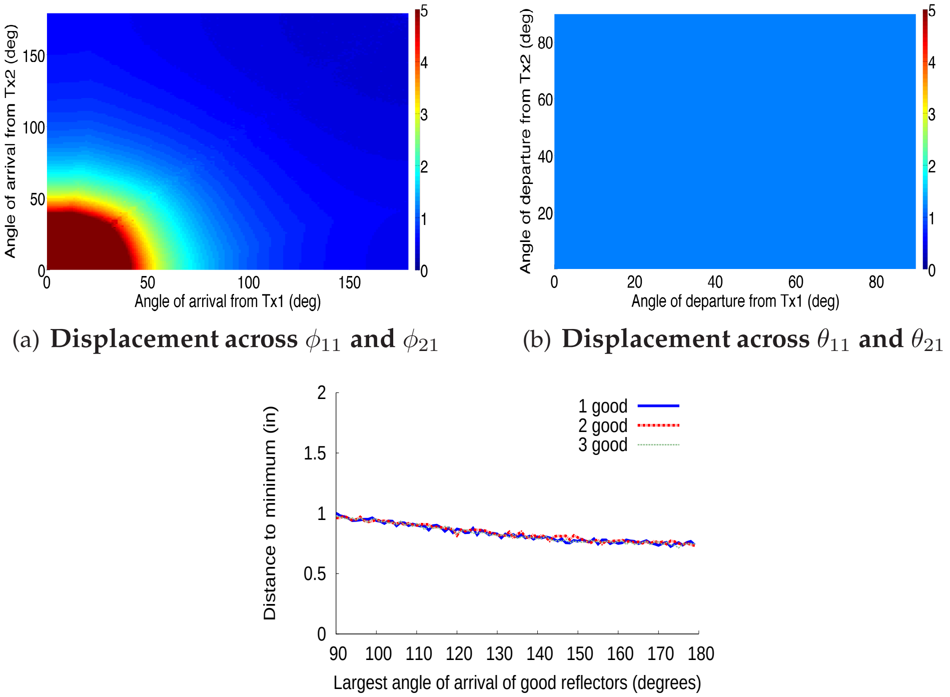
where, $c_{t1,i} = d_{t1,0}\sin\phi_{t1,i}/(\sin\phi_{t1,i} - \sin\theta_{t1,i})$ and $\beta_{t1,i} = (\phi_{t1,i} - \theta_{t1,i})/2$. □

Eqn. 4.6 shows how sliding a receive antenna impacts alignment, or more precisely, the residual interference orthogonal to the desired alignment. As Eqn. 4.6 is fairly complex, in the following sections, we extract insights from it considering specific scenarios.

■ 4.4.1 Impact of a Reflector on Motion-Based Alignment

We first consider the case of single randomly positioned non-absorbent reflector (i.e., $\gamma_i = 1$). Of course in reality, an indoor scenario would have many absorbent reflectors. However, studying this simplified case allows us to understand the trends that control how a reflector impacts alignment. Later in §4.4.2, we use these insights to understand the impact of many absorbent reflectors.

Recall that our objective is to understand, given that the receive antenna is at an ar-



(a) **Displacement across ϕ_{11} and ϕ_{21}**

(b) **Displacement across θ_{11} and θ_{21}**

(c) **Displacement with many reflectors over ϕ_{max}**

Figure 4-9: Displacement (in inches) to position of alignment (a): Across ϕ_{11} and ϕ_{21}, with fixed arbitrary θ_{11} and θ_{21}. We cap the maximum at 5 inches for ease of viewing. (b) Across θ_{11} and θ_{21}, with $\phi_{11}, \phi_{21} = 90°$. Geometry restricts $\theta_{11}, \theta_{21} \leq 90°$. (c) With 10 reflectors placed at random locations across ϕ_{max}, the largest angle of arrival among the good reflectors with respect to any interferer.

bitrary initial position, how much displacement is needed to move it to a location that achieves alignment (i.e., minimizes $|h_{orth}|^2$, the residual interference after alignment). To compute this displacement, we numerically compute $|h_{orth}|^2$ using Eqn. 4.6. Recall that the impact of the reflector on the signal from each interferer I_t is defined by the angle of arrival (ϕ_{t1}) and the angle of departure (θ_{t1}) of the signal, as shown in Fig. 4-7. Since we have two interferers and one reflector, in this example, we have four parameters: θ_{11}, θ_{21}, ϕ_{11} and ϕ_{21}, which determine this displacement.

Fig. 4-9(a) plots the mean displacement from an arbitrary initial position of the receive antenna to a position that achieves alignment by minimizing $|h_{orth}|^2$ locally.[7] The figure shows that the displacement is small when either ϕ_{11} or ϕ_{21} is large. Specifically, when either ϕ_{11} or ϕ_{21} is between $90°$ and $180°$, the required mean displacement varies between

[7] Our analysis of h_{orth} also reveals that these points of local minima, on average, decrease $|h_{orth}|^2$ below the noise floor, provided the environment has reflectors close to the receiver, that are behind it with respect to either sender.

about 0.6 inches and 1.2 inches. We note that this displacement and the results in Fig. 4-9(a) are based on Eqn. 4.6 for $f = 2.5$ GHz, which is the typical carrier frequency of 802.11. Hence the displacements of 0.6 and 1.2 inches correspond to distances of $\lambda/8$ and $\lambda/4$ respectively. The required displacements are even smaller at higher frequencies, where the wavelength is shorter, for e.g., in the 5 GHz Wi-Fi band.

So what does ϕ_{11} or $\phi_{21} > 90°$, mean in terms of the physical location of the reflector? Recall from Fig. 4-7 that ϕ_{t1} denotes the angle along which reflected signal arrives at at the receiver from interferer I_t. Thus, if $\phi_{t1} > 90°$, then this reflector is behind the receiver with respect to interferer I_t. Hence, we arrive at the following insight:

Insight 1: Reflectors which are behind the receiver with respect to the interferer (i.e. ϕ_{11} or $\phi_{21} > 90°$) enable a displacement on the order of 1 or 2 inches to a position that enables alignment. We call such reflectors *good reflectors*.

Note that it is common for most receivers in wireless networks to be placed on some platform (e.g. a table, the floor, etc.), which is usually behind the receiver with respect to at least one interferer, and hence serves as a good reflector.

One may wonder if the displacement to a position of alignment depends on either θ_{11} or θ_{21}, given that ϕ_{11} or ϕ_{21} are fixed at values $\geq 90°$. In fact, we observe that the angles of departure have no impact on the displacement, once we fix the angles of arrival. In particular, Fig. 4-9(b), demonstrates that the required mean displacement is constant versus θ_{11}, θ_{21}, for the case when ϕ_{11} and ϕ_{21} are both fixed at $90°$.

■ 4.4.2 Motion-Based Alignment with Multiple Reflectors

While our discussion so far focused on a single reflector, the natural question to ask is how does the displacement to a position of alignment change, given a large number of reflectors in the environment. To this end, we extend our analysis to multiple absorbent reflectors. In particular, we pick multiple reflectors with random angles of departure and arrival and an absorption coefficient chosen randomly between $[0.2, 0.6]$ [93]. We use our model in Eqn. 4.6 to evaluate how these reflectors affect the mean displacement required from an arbitrary location to reach a position that achieves alignment by locally minimizing $|h_{orth}|^2$.

We consider a scenario with 10 randomly positioned reflectors with different angles of arrival and departure. We consider the following three cases: (1) Only one of these reflectors is a "good reflector" with respect to either interferer I_1 or interferer I_2. (2) Two reflectors are "good reflectors" with respect to either interferer. (3) Three reflectors are "good reflectors" with respect to either interferer.

Fig. 4-9(c) plots the mean displacement needed to reach a position of alignment, i.e. minimize $|h_{orth}|^2$. We plot this quantity across the largest angle of arrival, $\phi_{max} = \max_{t\in\{1,2\},i}\phi_{t1,i}$, among all the good reflectors with respect to either interferer. The figure reveals the following insight:

Insight 2: As long as at least one good reflector exists in the environment, i.e. one reflector is behind the receiver with respect to either interferer, the mean displacement needed to reach a position of alignment, is in the vicinity of $\lambda/8$ to $\lambda/4$, i.e. around 1 or 2 inches, for 2.4-2.5GHz transmissions.

While our analysis so far has focused on interference alignment, it can readily be extended to interference nulling. In particular, we can consider nulling as a special case of alignment where h_{orth} is the projection along the direction of the receive antenna where the signal needs to be nulled (i.e., one of the axes in the antenna space). Hence, the analysis of h_{orth} is similar.

■ 4.5 Stochastic Hill Climbing

In this section, we present a Stochastic Hill Climbing algorithm that automatically adjusts the receive antenna to seek positions of interference alignment or nulling. To find these positions, the algorithm minimizes the *Interference to Noise Ratio* (INR) from the interfering transmitter(s). For interference nulling, INR is the ratio of the power of the interfering transmitter to the power of noise. For interference alignment, we define the INR as follows. Let (α, β) denote the direction along which the projected power from the interfering signals is minimum. Recall that the received signal must be projected along this direction, to decode the desired signal. Thus, we define the INR for alignment as the ratio of the power of the interfering signals projected along this direction, to the noise power σ^2. For a system with two single-antenna transmitters and a two-antenna receiver, the INR can be computed based on the wireless channels as:

$$INR = \frac{1}{\sigma^2} \min_{\alpha,\beta} |h_{11}\alpha + h_{12}\beta|^2 + |h_{21}\alpha + h_{22}\beta|^2$$

where h_{ij} is the channel from transmitter i to receive antenna j in Fig. 4-6. The solution of this minimization problem has a closed form (Proof provided below):

$$INR = \frac{1}{2\sigma^2}\left(|h_{11}|^2 + |h_{21}|^2 + |h_{12}|^2 + |h_{22}|^2\right) - \\ \frac{1}{\sigma^2}\sqrt{\left|\frac{|h_{11}|^2 + |h_{21}|^2 - |h_{12}|^2 - |h_{22}|^2}{2}\right|^2 + |h_{11}h_{12}^* + h_{21}h_{22}^*|^2} \tag{4.11}$$

Proof of Eqn. 4.11: Consider two transmitters and a single two-antenna receiver. Let h_{ij} denote the channel from transmitter i to receive antenna j. Our goal is to minimize the interference to noise ratio (INR) for alignment, given by:

$$INR = \frac{1}{\sigma^2}\min_{\alpha,\beta} |h_{11}\alpha + h_{12}\beta|^2 + |h_{21}\alpha + h_{22}\beta|^2$$

where $|\alpha|^2 + |\beta|^2 = 1$ and σ^2 is the noise power.

Without loss of generality, we can choose α to be a positive real, and $\beta = \sqrt{1-\alpha^2}e^{j\theta}$, where $\theta \in [-\pi, \pi]$. Hence, our goal is to minimize:

$$\begin{aligned}
&|h_{11}\alpha + h_{12}\beta|^2 + |h_{21}\alpha + h_{22}\beta|^2 \\
=&(|h_{11}|^2 + |h_{21}|^2)\alpha^2 + (|h_{12}|^2 + |h_{22}|^2)|\beta|^2 \\
&\qquad + 2\text{Re}\{\alpha\beta^*(h_{11}h_{12}^* + h_{21}h_{22}^*)\} \\
=&(|h_{12}|^2 + |h_{22}|^2) + \alpha^2(|h_{11}|^2 - |h_{12}|^2 + |h_{21}| - |h_{22}|^2) \\
&\quad + 2\alpha\sqrt{1-\alpha^2}\text{Re}\{(h_{11}h_{12}^* + h_{21}h_{22}^*)e^{-j\theta}\}
\end{aligned}$$

We choose $\theta = \pi + \angle(h_{11}h_{12}^* + h_{21}h_{22}^*)$, so as to minimize the third term. Hence, we need to minimize:

$$\begin{aligned}
(|h_{12}|^2 + |h_{22}|^2) + \alpha^2(|h_{11}|^2 - |h_{12}|^2 + |h_{21}| - |h_{22}|^2) \\
-2\alpha\sqrt{1-\alpha^2}|h_{11}h_{12}^* + h_{21}h_{22}^*|
\end{aligned}$$

Let $A = (|h_{11}|^2 - |h_{12}|^2 + |h_{21}|^2 - |h_{22}|^2)$, $B = |h_{11}h_{12}^* + h_{21}h_{22}^*|$ and $C = (|h_{12}|^2 + |h_{22}|^2)$.

Hence, we wish to minimize:

$$INR = \frac{1}{\sigma^2}(A\alpha^2 - 2B\alpha\sqrt{1-\alpha^2} + C) \tag{4.12}$$

To do so, we set its derivative with respect to α to zero, i.e.

$$\begin{aligned}
2A\alpha - 2B\sqrt{1-\alpha^2} + 2B\alpha^2/\sqrt{1-\alpha^2} &= 0 \\
2A\alpha\sqrt{1-\alpha^2} - 2B(1-2\alpha^2) &= 0 \\
A^2\alpha^2(1-\alpha^2) &= B^2(1-2\alpha^2)^2 \\
\alpha^4(A^2+4B^2) - \alpha^2(A^2+4B^2) + B^2 &= 0
\end{aligned}$$

The above equation is quadratic in α^2. Solving, we have:

$$\alpha = \sqrt{\frac{1}{2} - \frac{A}{2\sqrt{A^2+4B^2}}}$$

Substituting α in Eqn. 4.12:

$$\begin{aligned}
INR &= \frac{1}{\sigma^2}\left(A\left(\frac{1}{2} - \frac{A}{2\sqrt{A^2+4B^2}}\right) - 2B\sqrt{\frac{1}{4} - \frac{A^2}{4(A^2+4B^2)}} + C\right) \\
INR &= \frac{1}{\sigma^2}\left((A/2+C) - \frac{A^2}{2\sqrt{A^2+4B^2}}) - 2B^2\sqrt{\frac{1}{A^2+4B^2}}\right) \\
INR &= \frac{1}{\sigma^2}\left((A/2+C) - \frac{1}{2}\sqrt{A^2+4B^2}\right)
\end{aligned}$$

Substituting for A, B and C in the above equation, we have:

$$\begin{aligned}
INR = &\frac{1}{2\sigma^2}\left(|h_{11}|^2 + |h_{21}|^2 + |h_{12}|^2 + |h_{22}|^2\right) - \\
&\frac{1}{\sigma^2}\sqrt{\left|\frac{|h_{11}|^2 + |h_{21}|^2 - |h_{12}|^2 - |h_{22}|^2}{2}\right|^2 + |h_{11}h_{12}^* + h_{21}h_{22}^*|^2}
\end{aligned}$$

Which precisely corresponds to Eqn. 4.11. □

Because channels are continuous functions over space, the INR profile is continuous and smooth. In other words, an incremental movement in any given direction would lead to a gradual increase or decrease in INR. Hence, to minimize the INR, a receiver can slide its antenna gradually until it reaches a local minimum.

We leverage this intuition in designing our stochastic hill climbing algorithm to reduce the INR (pseudo-code in Alg. 2). At a high level, our algorithm proceeds as follows: the receiver continuously monitors the INR from its interferer(s) (using Eqn. 4.11). Initially, it picks an arbitrary direction, and slides one of its antennas in that direction. Specifically, it slides the antenna either forward or backward, depending on which of those decreases the INR. It continues to move the antenna along that direction until the INR reaches a local minimum (in that direction). Then, it rotates the antenna either clock-wise or counter clock-wise about a small arc (of radius a few mm), whichever decreases the INR. Again, the receiver keeps rotating that antenna until the INR reaches a local minimum. The receiver repeats the above process, switching between translations and rotations, until it reaches a local minimum (in both translation and rotation). If the INR at the local minimum is below the noise floor, the algorithm terminates. Otherwise, the receiver moves the antenna to a random location and repeats the process again.

In practice, across all experiments in §10.5, the stochastic hill climbing algorithm converges to an INR below the noise floor in an average of 3 tries.

2 Pseudo-code for Stochastic Hill Climbing

```
  $mode = +1$                                          ▷ 1: Translate, -1: Rotate
  $direction = +1$                                     ▷ 1: Forward, -1: Reverse
  $attempts = 0$
  ▷ Loop while INR is above $0$ dB
  while $INR > 0$ dB do
      switch $mode$
          case $+1$:  TRANSLATE(direction);                ▷ In Translate Mode
          case $-1$:  ROTATE(direction);                   ▷ In Rotate Mode
      end switch
      $prev\_INR = INR$                                 ▷ Previous INR of sender
      $INR =$ GETINR()                           ▷ Update INR based on channels
      if $INR > prev\_INR$ then                               ▷ If INR increased
          $direction = -direction$                           ▷ Switch direction
          $attempts = attempts + 1$
      end if
      if $attempts == 2$ then                                   ▷ Hit a minimum
          $mode = -mode$                                        ▷ Switch modes
          $attempts = 0$
      end if
      if At local minimum with $INR > 0$ dB then
          ▷ Move to Random Location
      end if
  end while
```

4.6 Reciprocity

In this section, we show how MoMIMO leverages channel reciprocity to allow concurrent uplink transmissions (such as ACKs). Consider two wireless nodes i and j. Let h_{ij} and h_{ij}^{rev} denote the forward channel from i to j and reverse channel from j to i, respectively. Then reciprocity [63] states that:

$$h_{ij}^{rev} = t_i h_{ij} r_j \tag{4.13}$$

Where t_i and r_j are fixed constants that depend purely on nodes i and j respectively.

Eqn. 4.13 has an interesting consequence for interference nulling. Specifically, when MoMIMO adjusts j's receive antenna so that $h_{ij} \approx 0$, this also results in $h_{ij}^{rev} \approx 0$. Hence, nulling the downlink channel from interferer i to receiver j by motion, also nulls the uplink channel from j to i.

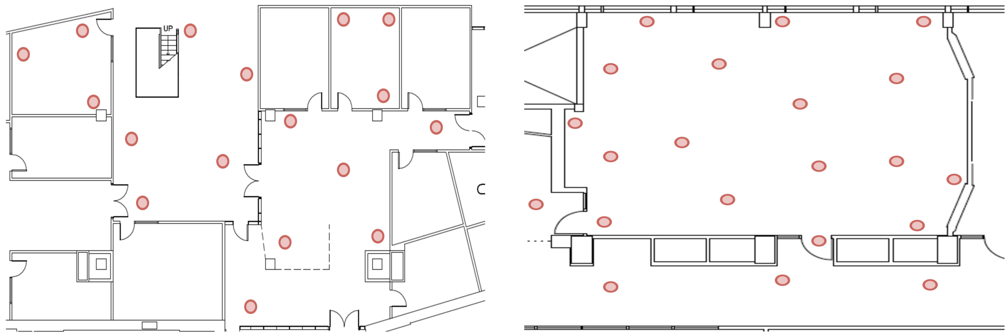
Reciprocity can also be exploited for interference alignment. Consider two single-antenna nodes I_1 and I_2 causing interference at a two-antenna receiver Rx, as shown in Fig. 4-6. MoMIMO moves one of Rx's receive antennas to a position of interference alignment, so that (Eqn. 4.3):

$$h_{11}h_{22} - h_{21}h_{12} \approx 0 \tag{4.14}$$

Now, suppose the two-antenna receiver Rx needs to send a single stream on the uplink without interfering with I_1 or I_2. It precodes its signal by $(-h_{12}^{rev}, h_{11}^{rev})$ so that the effective uplink channel at interferer I_1: $-h_{12}^{rev}h_{11}^{rev} + h_{11}^{rev}h_{12}^{rev} = 0$, is nulled. Interestingly, because the receive antenna is in a position of alignment at the downlink, this precoding also nulls the effective uplink channel at I_2. Specifically, from Eqn. 4.13 and 4.14, the effective uplink channel at I_2 is:

$$\begin{aligned} -h_{12}^{rev}h_{21}^{rev} + h_{11}^{rev}h_{22}^{rev} &= -t_1h_{12}r_2t_2h_{21}r_1 + t_1h_{11}r_1t_2h_{22}r_2 \\ &= t_1t_2r_1r_2(h_{11}h_{22} - h_{21}h_{12}) \approx 0 \end{aligned}$$

Thus, MoMIMO's interference alignment and nulling by motion provides not only concurrent downlink transmissions, but by leveraging precoding and reciprocity, also enables



(a) Testbed in Bldg. 1

(b) Testbed in Bldg. 2

Figure 4-10: Testbed topology. The figure depicts the cropped floor plans of two buildings where the experiments were conducted. The red marks on the floor plans depict locations for wireless nodes. Different random subsets of these locations for the transmitters and receivers are picked for different experiments.

concurrent uplink transmissions.

■ 4.7 Experimental Setting

We describe our experimental environment, which is used to illustrate the gains of MoMIMO.

Implementation. Nodes in our experiments are equipped with USRP-N210 software radios [41] and RFX 2400 daughter-boards. They communicate in the 2.4 GHz Wi-Fi band using 20 MHz signals. We implement OFDM directly in the USRP Hardware Driver (UHD). We use various 802.11 modulations (BPSK, 4QAM, 16QAM, and 64QAM), coding rates, and choose between them using the effective-SNR bitrate selection algorithm [69]. Our implementation periodically measures the channels by listening to packets from the desired and interfering transmitters. It tracks the INR of the interfering signals for interference alignment or nulling.

To emulate sliding antennas, the antenna of each USRP is mounted on an iRobot Create robot, controlled by an ASUS EEPC 1015PX netbook. We implement stochastic hill climbing in a fully distributed manner using iRobot Create's Open Interface. Each receiver measures INR for interference alignment or nulling in real-time from the software radio(s), and automatically maneuvers the robot to minimize INR from any interferer(s). It also queries the robot sensors to ensure that the robot does not move beyond 2 inches from its initial position as it seeks to minimize its INR.

Tested Environments. We evaluate MoMIMO in indoor settings, in line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios, and in two different buildings.

Our main experiments were conducted in Building 1 depicted in Fig. 7-8(a), which hosts a computer science department. The building is built mostly of flat reinforced concrete slabs and columns. The enclosure of the exterior walls is of metal and brick. The interior walls are gypsum with steel sheet metal studs. The rooms are standard offices with several desks, chairs, cabinets, and computer workstations.

We also ran experiments in Building 2, which is constructed in 1963, and has thick concrete interior walls. We performed our experiments in a large classroom connected to a long corridor with the layout as shown in Fig. 7-8(b). The classroom consists of several desks and chairs.

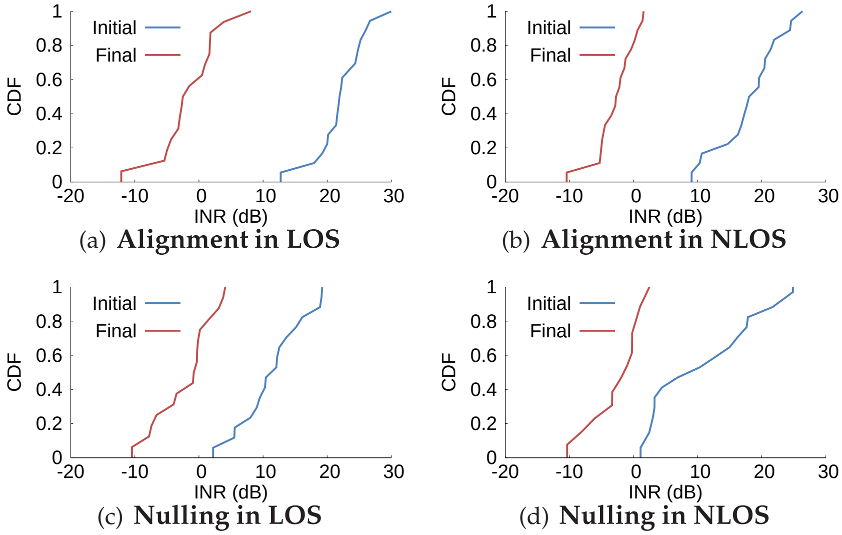
Metrics. To evaluate MoMIMO's effectiveness at achieving interference alignment and nulling, we measure the *interference to noise ratio (INR)*. In the case of nulling, the INR is computed as the total received interference power after running the stochastic hill climbing algorithm. In the case of alignment, the INR is measured after running the stochastic hill climbing algorithm by projecting the interference signals orthogonal to the desired alignment direction, and computing the interference power after projection. In both cases, an INR less than 0 dB means that MoMIMO has reduced the interference below the noise level.[8]

Other metrics of interest include the throughput, which we measure using the effective-SNR metric as described in [69], and the required antenna displacement for nulling or alignment, which we measure as the distance from the initial location of the antenna to its location after running the stochastic hill climbing algorithm.

■ 4.8 Empirical Results

We evaluate the performance of MoMIMO in the testbed environment described in §4.7.

[8] To be able to evaluate the INR accurately, packets sent by the interferer contain a header with twenty known OFDM symbols. This allows us to measure INR up to an accuracy of -13 dB by averaging the wireless channel across these repeated symbols. While this allows us to accurately measure the residual interference, in practice, pushing the INR below -3 dB is unnecessary. Thus 2-3 known OFDM symbols are sufficient for this accuracy.



(a) **Alignment in LOS**

(b) **Alignment in NLOS**

(c) **Nulling in LOS**

(d) **Nulling in NLOS**

Figure 4-11: INR before and after applying MoMIMO's alignment or nulling. The figure depicts the CDF of motion-based alignment in (a), (b) and nulling in (c), (d). Our experiments are performed in line-of-sight scenarios in (a), (c) and non-line-of-sight scenarios in (b), (d).

■ 4.8.1 Microbenchmarks

We first investigate whether local antenna adjustments, using our stochastic hill climbing algorithm, can indeed deliver interference alignment and nulling. To do so, we quantify the INR after MoMIMO's alignment and nulling, and check whether it can be reduced below the noise level, i.e., 0 dB.

Interference Alignment. We experiment with the setup in Fig. 4-6, which consists of two single-antenna interferers and a 2-antenna receiver. We place these nodes randomly in our testbeds in both line-of-sight (LOS) and non-line of-sight (NLOS) scenarios. The two interferers concurrently transmit packets back-to-back. The receiver estimates the total power received on both of its antennas; this power constitutes the initial INR. The receiver then runs stochastic hill climbing until the two received signals are aligned. Subsequently, it computes the final INR as the projection of the received signals along the direction orthogonal to the desired alignment as described in §4.5. We repeat this experiment 20 times, each time placing the interferers and receiver at different locations in our topology.

Figs. 4-11(a) and 4-11(b) show the cumulative distribution functions (CDFs) of both the initial and final INRs in these experiments for LOS and NLOS scenarios. The figures show an average INR reduction of about 22 dB in both LOS and NLOS scenarios. Note that the median residual INR is -0.5 dB and -2.5 dB, in LOS and NLOS respectively, which is below the noise floor, thereby enabling significant throughput gains.

The CDFs also show that in a few percent of the runs, the INR after alignment may be above the noise level by a few dBs. These runs correspond to cases of very strong interference, over 25 dB, and hence the few dB of residual INR. Even in these scenarios, the reduction in INR is large and exceeds 20 dB. Overall, the figure shows that motion-based alignment produces similar accuracy to multi-antenna alignment [99], but requires only single antenna transmitters.

Interference Nulling. We place a single-antenna interferer and a single-antenna receiver randomly in our testbed, in both LOS and NLOS scenarios. The interferer transmits a stream of packets back-to-back. The receiver slides its antennas following the stochastic hill climbing algorithm in order to null the signal from the interferer. Upon convergence, it computes the final INR. We repeat this experiment 20 times, each time placing the nodes in different locations in our testbed.

Figs. 4-11(c) and 4-11(d) show the CDFs of the initial and final INRs in these experiments for both line-of-sight (LOS) and non-line-of-sight (NLOS) scenarios. The figure shows that after MoMIMO nulling, the median residual INR is -0.3 dB in LOS and NLOS, and median reduction in INR is about 15 dB. Our results show that motion-based nulling effectively cancels interference and brings it to the noise level.

In comparison with the alignment experiments, we note that the initial INR for nulling is much lower. This is because in alignment, the initial INR is due to two interferers, whereas here it is due to only one interferer.

■ 4.8.2 Throughput Comparison

We assess the throughput benefits of MoMIMO against two baselines: standard 802.11n and n+ [99]. We compare against n+ to show that MoMIMO delivers gains even when compared with a system that employs advanced MIMO techniques.

For this experiment, we use the heterogeneous network topology in Fig. 4-5. We note that other topologies like those in Fig.4-4(b) and 4-4(c) show a bigger gain for MoMIMO, as explained in §4.3. However, we chose to present empirical results for the heterogeneous network topology because it is more complex than the others and requires combining both

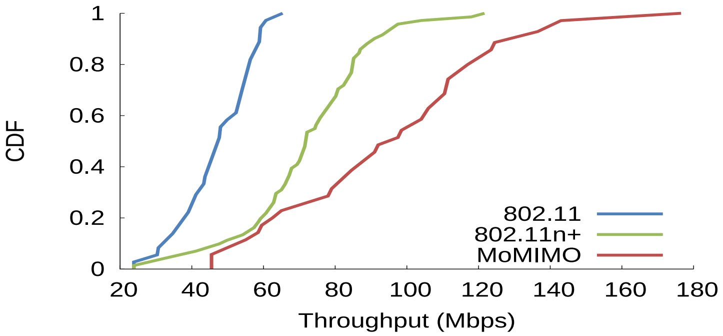


Figure 4-12: Total Network Throughput. Total network throughput of all three schemes: 802.11n, n+, and MoMIMO.

motion-based nulling and alignment.

Recall from §4.3 that the heterogeneous network in Fig. 4-5 has two single-antenna transmitter-receiver pairs and one 2-antenna transmitter-receiver pair. For this network, MoMIMO uses sliding antennas to null the signal of Tx1 at Rx2 and that of Tx2 at Rx1. It also needs to align the signals of Tx1 and Tx2 at Rx3. In this case, MoMIMO can deliver 3 concurrent streams. In contrast, 802.11n has to budget the wireless medium between the three Tx-Rx pairs which yields an average of 1.33 concurrent streams, while n+[99] can enable 2 concurrent streams.

For each run of our experiments we pick a random subset of the nodes in the testbed to represent the nodes in the heterogeneous network in Fig. 4-5. We consider a single long-lasting flow from each sender to its receiver. We repeat the experiment at 20 randomly chosen locations. At each location, we record throughputs for MoMIMO, 802.11n and n+.

For each of the MoMIMO runs, the network initially uses 802.11n with each pair transmitting in a different time slot. During this phase, the nodes do not transmit concurrently; instead they measure the channel and run stochastic hill climbing. Once stochastic hill climbing reaches the desired nulling and alignment, we allow the nodes to transmit concurrently.

Fig. 4-12 shows CDFs of the obtained throughput in MoMIMO as well as those of 802.11n and n+. On average, MoMIMO achieves a throughput gain of $1.98\times$ over 802.11 and $1.31\times$ gain over n+. In comparison, the expected throughput gains for this heterogeneous network as reported in §4.3.3 are $2.25\times$ over 802.11n and $1.5\times$ over n+. Thus, our experimental gains are close to the expected ones. Note that both MoMIMO and n+ suffer from a slight performance dip in comparison to their theoretical gains; this is due to small

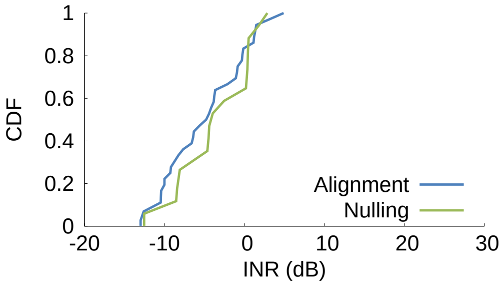


Figure 4-13: Leveraging Reciprocity. CDF of INR and Reciprocal INR for interference alignment and nulling.

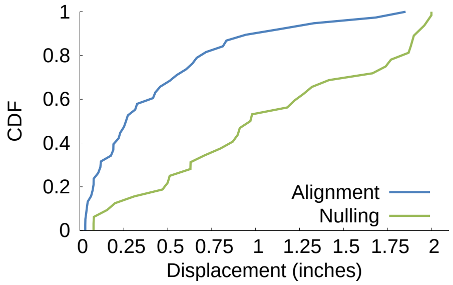


Figure 4-14: Displacement for INR reduction. CDF of the total displacement of the receiver from its initial position, across multiple experiments, for interference alignment or nulling.

residual interference power after alignment and nulling.

■ 4.8.3 Reciprocity

Reciprocity enables MoMIMO to perform motion-based alignment/nulling on the forward path and still enjoy alignment and nulling on the reverse path, thereby enabling ACKs to be transmitted concurrently without interference. In this section, we evaluate how eliminating interference by applying MoMIMO on the forward path translates into eliminating interference on the reverse path.

Here, we use the same experiments described in §4.8.1 for both alignment and nulling. The difference however is that after we align/null along the forward path, we reverse the direction of the transmission, i.e., make the receiver transmit and the previous transmitter receive. For the case of alignment and as described in §4.6, the receiver also precodes the ACK to null it at one of the aligned senders. Our objective is to show empirically that this naturally leads to nulling the signal at the other sender. Hence we measure the corresponding reciprocal INR. For nulling we simply want to show that nulling in the forward direction leads to nulling in the reverse direction. Hence we measure the reverse INR.

Fig. 4-13 plots the CDFs of the reverse INRs for both alignment and nulling. Each CDF is taken over 20 different runs with different node locations. The figure shows that indeed applying stochastic hill climbing in the forward direction virtually eliminated the INR in the reverse direction as well, reducing it below the noise level (i.e., below 0 dB).

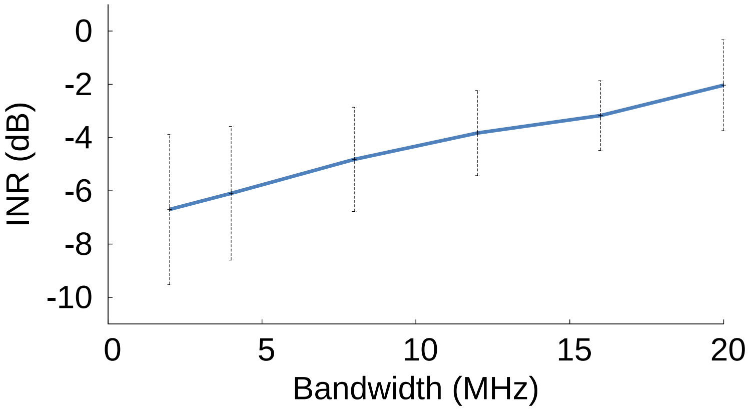


Figure 4-15: INR across Bandwidth. Plot of the mean and standard deviation of the INR at the converged position for interference alignment across bandwidth.

■ 4.8.4 Displacement

Next, we would like to quantify the amount of displacement by which MoMIMO needs to slide the antennas to achieve alignment or nulling. To do so, we measure the displacement between the initial and final positions of the receive antennas in the throughput experiments in §4.8.2.

Fig. 4-14 plots the CDFs of this displacement for both alignment and nulling. As expected, since we restricted the motion of the antennas to within a 2 inch radius, the maximum displacement across all experiments is 2 inches. Further, the figure shows that MoMIMO's stochastic hill climbing algorithm needs a mean displacement of 0.44 inches for interference alignment and 1.17 inches for interference nulling. Alignment is easier to satisfy as many options exist for the direction along which two senders may be aligned, which provides extra flexibility for the choice of channels.

■ 4.8.5 Impact of Channel's Bandwidth

Our previous experiments use a bandwidth of 20 MHz, the common setting for 802.11. In this experiment, we evaluate how channel bandwidth impacts MoMIMO's performance.

As in §4.8.1, we consider two single-antenna interferers and a two-antenna receiver. The receiver performs stochastic hill climbing by moving one of its antennas to align the signals from the two single-antenna interferers. Upon convergence, we fix the initial position of the receiver and repeat the experiment for different channel bandwidths: 2MHz, 4MHz, 8MHz, 16MHz, and 20MHz. For each channel width, we repeat this experiment in 10 randomly chosen locations in our testbed.

Fig. 4-15 plots the mean and standard deviation of the INR at the receiver as a function of the bandwidth, averaged over all our experiments. The plot reveals that the INR rises by an average of 5 dB as the bandwidth is increased from 2MHz to 20MHz. This is expected because a receiver perceives greater frequency diversity across OFDM subcarriers with increased bandwidth. Nevertheless, even at 20MHz, the mean INR across these experiments was around -2 dB, which is well below the noise floor. Thus, even though the performance of MoMIMO is impacted by an increase in bandwidth due to frequency diversity, MoMIMO continues to provide a significant reduction in INR.

■ 4.8.6 Performance with Off-the-Shelf 802.11 Senders

Now, we evaluate how MoMIMO performs with off-the-shelf 802.11 senders. Specifically, since the Stochastic Hill Climbing algorithm only uses the power of the interferer to move the receiver; in principle, it should work even with off-the-shelf 802.11 interferers. To validate this, we use commodity Atheros AR9285 wireless cards as the interfering senders and run the Stochastic Hill Climbing algorithm on USRP-N210 receivers.

Fig. 4-16 plots the CDF of the resulting INR reduction from the interfering 802.11 senders to the USRP-N210 receivers. The CDF is taken across different runs that span both line-of-sight and non-line-of-sight scenarios. The plot shows that, on average, the Stochastic Hill Climbing algorithm reduces the INR of the 802.11 interferer by about 23 dB. Thus, we conclude that, even with off-the-shelf 802.11 cards, the Stochastic Hill Climbing algorithm can significantly reduce interference and hence enables throughput gains.

■ 4.8.7 Performance in Dynamic Environments

In this experiment, we study how MoMIMO responds to changes in the wireless channel. MoMIMO nodes incorporate a fail-safe mode that responds to channel changes by defaulting to 802.11n and re-running the stochastic hill climbing algorithm. Specifically, during the periods in which the receive antenna is still moving and has not reached a nulling/alignment position, as well as in which the nulling/alignment position has changed, the nodes fall back to 802.11n and do not transmit extra concurrent streams beyond what is allowed by 802.11n. Note that this behavior is natural in MoMIMO since, once the nulling does not hold, the receiver immediately starts hearing the interference.

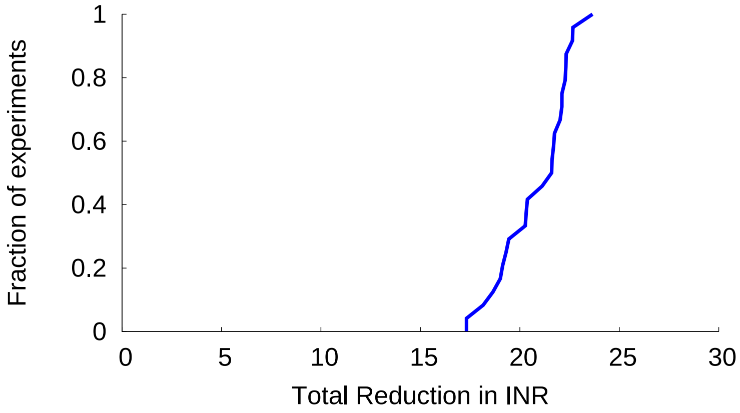


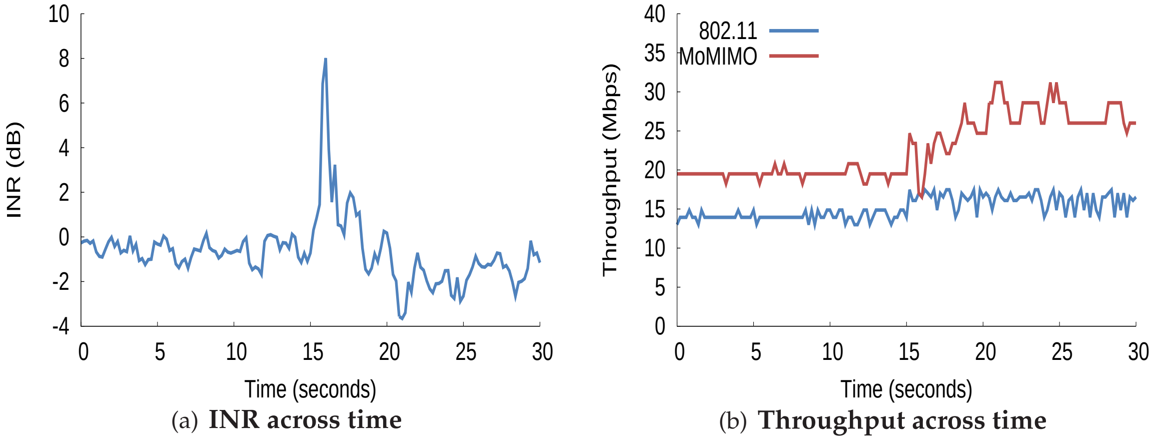
Figure 4-16: Performance with real 802.11 senders. CDF of the net reduction in INR from the commodity 802.11 card to the USRP-N210 receiver across experiments

Similarly, if the alignment does not hold, the receiver sees the interference signal in the whole 2-D space as opposed to along a single direction. Thus, if the channel changes enough to disturb alignment or nulling, the receiver can detect this effect in real-time and fall back to standard 802.11n.

To assess MoMIMO's fail safe mechanism, we repeat the experiment in §4.8.2, which uses the heterogeneous network. We focus on the two-antenna receiver (Rx3 from Fig. 4-5), which slides its antenna to align the signals from the two single-antenna interferers Tx1 and Tx2. We consider a scenario where the interferers (Tx1 and Tx2) are inside an office with an open door and the receiver is outside the office. Before starting the experiment, the receiver runs the stochastic hill climbing algorithm to align the two single-antenna transmitters. Our experiment spans 30 seconds. At $t = 15.4$ seconds, we close the door so that the interferers (Tx1 and Tx2), and the receiver are no longer in line-of-sight of each other. We track the INR of alignment at the receiver for the full duration of 30 seconds. We also measure the receiver's throughput obtained based on the effective SNR [69] metric.

Fig. 4-17(a) plots a trace of the measured INR over time. The trace depicts a distinct spike of about 8 dB in INR at $t = 15.4$ seconds, when the door is closed. However the stochastic hill climbing algorithm responds quickly, reducing the INR to around 0 dB by $t = 16.6$ seconds. Note that the delay of 1.2 seconds was partly due to the limitations of the iRobot Create which is programmed to move at an average speed of 20 mm/s for accurate maneuverability.

Fig. 4-17(b) plots the throughput of the MIMO receiver over time, under identical set-



(a) **INR across time** (b) **Throughput across time**

Figure 4-17: Performance of Interference alignment in Dynamic Environments. The plot shows a trace of the INR and throughput for 30 seconds, where at $t = 15.4s$, the channels change from line-of-sight to non-line-of-sight.

tings for 802.11 and MoMIMO. At $t = 15.4$ seconds, the spike in INR caused by the change in channel triggers MoMIMO's fail-safe mechanism. Hence, MoMIMO ceases concurrent transmissions, and falls back to 802.11n's throughput levels (around 17 Mb/s on average). During this period, stochastic hill climbing moves the antenna but without activating concurrent transmissions. Once stochastic hill climbing causes the INR to fall close to the noise level MoMIMO switches on concurrent transmissions. By $t = 16.6$ seconds, the average throughput is around 24 Mb/s, restoring MoMIMO's original gains over 802.11n for the new channel.

■ 4.9 Related Work

Related work falls in the following two categories:

(a) MIMO Interference Management: Recent years have witnessed advances in MIMO interference management from both the theoretical [112, 21] and the empirical research communities [166, 63, 99, 151, 61, 141, 153]. Past systems implementing interference alignment [99, 63, 87] typically require multi-antenna senders that are aware of channel state information. Blind interference alignment [119], does not require channel state information at the transmitter, but needs symbol-level time synchronization between transmitters and receivers. Some theoretical work proposes interference alignment for single-antenna nodes either in frequency [163], or in time [122]. In contrast, MoMIMO is the first system to demonstrate that interference alignment and nulling can be performed purely by moving the receive antennas by just about one or two inches.

(b) Exploiting Motion to Improve Wireless Performance: Prior work demonstrated that mobility improves the performance of wireless network. For instance, Grossglauser and Tse [66] show that a wireless network of freely moving nodes has a higher capacity than a network with stationary nodes. Also, research from the robotics community [13, 73, 103] has leveraged mobility to improve the quality of wireless channels. For example, in [13, 73], robots improve their connectivity by maintaining a line-of-sight path to their access points. Other work observes that a robot can improve its throughput by sampling different locations in the environment, and choosing the one that maximizes its throughput (due to a higher SNR) [103]. Furthermore, recent work has leveraged node movement to improve beamforming [26] and energy consumption [190]. Additionally, some recent products allow the user to manually slide the antenna to avoid dead spots and improve the SNR [174].

MoMIMO builds on this past work but differs from it by being the first to show that motion enables MIMO-type interference alignment and nulling. Said differently, past work uses motion to improve the SNR but does not show that motion enables MIMO multiplexing gains.

■ 4.10 Discussion

In this chapter, we introduced MoMIMO, a technique that demonstrated, for the first time, that interference alignment and nulling can be achieved, even with single-antenna transmitters, by simply moving the receive antenna. We also showed that the amount of antenna displacement needed is fairly small ($\sim$ one inch); hence, MoMIMO can be achieved by sliding antennas.

This chapter focused on demonstrating the feasibility and potential gains of MoMIMO. Important topics for future work include: 1) designing a medium access protocol that leverages the synergy between motion and interference alignment as demonstrated by MoMIMO; and 2) studying MoMIMO's gains in different classes of dynamic environments, e.g. static vs. mobile clients.

We believe MoMIMO presents a new paradigm for interference management, especially in settings of long-lived interference patterns such as home networks and data-center flyways. Further, we envision that future systems may exploit motion-based interference alignment and nulling to enable new applications at the intersection of networking and robotics, where mobility is already innate.

CHAPTER 5
Indoor Positioning

Beyond communication, my research enables new services, particularly indoor positioning and navigation. Imagine a world where you can walk into a building you've never visited before, and find your location, simply using the wireless infrastructure around you. Such a system can get the best offers on the shoes you are browsing at a mall; or, fetch a description of the painting in front of you at a museum. This vision has motivated much work both in academia and industry. Apple recently acquired WifiSlam, and Google and Microsoft are developing in-house solutions.

Despite massive interest in this space, ubiquitous and accurate indoor positioning is a difficult problem yet to be achieved. The main culprit is that wireless signals bounce off walls and objects causing an effect called multipath – a major problem known to any wireless engineer. Multipath creates ambiguity at the receiver on where the signal originated from, hence hampering its ability to localize the transmitter.

Past solutions attempted to resolve this issue in two ways: 1) elaborately fingerprinting the environment to create a database of how the signal may look like for each possible location of the transmitter, or 2) expensive infrastructure unavailable on cellphones, like large arrays of antennas. This dissertation presents new systems that require neither. Our work spans both Wi-Fi and LTE cellular networks.

Underlying our solution is a new way to actively measure wireless signals to invert the effect of multipath. Specifically, we leverage the fact that as a wireless receiver moves, it perceives different wireless signals, as the signal copies along multiple paths combine differently at each receiver location. By intelligently processing these observations, one can extract the copy of the signal along the direct path and identify the actual direction of the transmitter. We show how this approach can be implemented on commodity Wi-Fi receivers (e.g., Wi-Fi devices, access points or LTE femtocells) to achieve accurate indoor positioning.

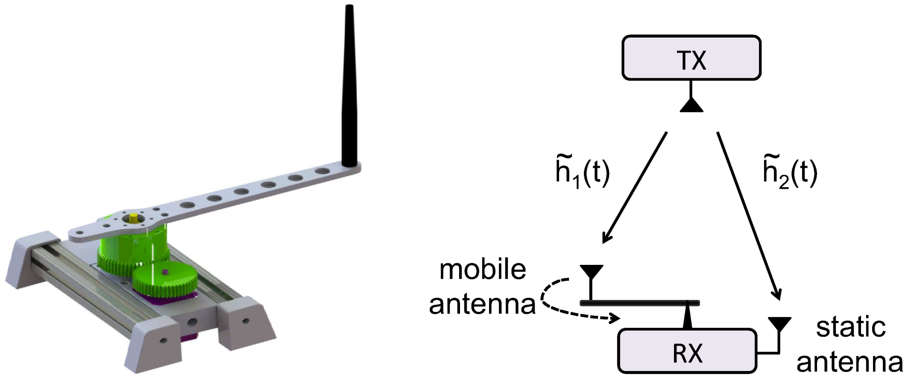
The rest of this section presents two systems that achieve indoor positioning from different perspectives: from that of the wireless infrastructure and user device. The first achieves accurate indoor positioning using simple hardware modifications that can be plugged in to existing wireless infrastructure. This approach requires no modification to user devices and can track them continuously as they move through the deployment space. We demonstrate the system specifically in the context of LTE networks, and show how it can help shine light into the performance of today's LTE networks in the indoor space. The second is a system that runs directly on a user device connected to pre-existing Wi-Fi infrastructure that does not need to be modified in any way. This system is a software-only localization service that a user can run on demand, whenever they twist their device. We demonstrate how both systems achieve tens of centimeters accuracy in indoor environments, without the need for specialized infrastructure or elaborate fingerprinting prior to deployment.

Below, we detail the key challenges and high-level ideas underlying these systems and how the relate to prior work.

■ 5.1 LTE Radio Analytics Made Easy and Accessible

Despite the rapid growth of next-generation cellular networks, researchers and end-users today have limited visibility into the performance and problems of these networks. As LTE deployments move towards femto and pico cells, even operators struggle to fully understand the propagation and interference patterns affecting their service, particularly indoors.

In chapter 6, we present LTEye [89], the first open platform to monitor and analyze LTE radio performance at a fine temporal and spatial granularity. LTEye accesses the LTE PHY layer without requiring private user information or provider support. It provides deep insights into the PHY-layer protocols deployed in these networks. At the heart of LTEye



(a) LTEye Sniffer Prototype (b) Wireless Channels

Figure 5-1: LTEye. (a) Prototype of the rotating antenna on a LTEye sniffer. (b) Depicts the measured wireless channels from a transmit antenna to the mobile and static antennas respectively

is a novel system design that uncovers not just the performance of users, but also where they are located, so that users with poor performance can be actively assisted. To do so, LTEye brings techniques from the world of radar systems to cellular signals – it builds a passive rotating antenna design using 3-D printers that sniff wireless signals from nearby cellphones to localize them. These antennas can be readily attached to LTE femtocells or low-cost beacons in indoor environments. Our approach uses these moving antennas to employ a new form of synthetic aperture radar (SAR), that can operate over cellular signals for the first time. This enables businesses and end-users to localize mobile users and capture the distribution of LTE performance across spatial locations in their facility. As a result, they can diagnose problems and better plan deployment of repeaters or femto cells. LTEye's analytics enable researchers and policy makers to discover performance deficiencies, stemming from inefficient spectrum utilization, inter-cell interference. An experimental evaluation of LTEye threw light into the PHY-layer performance and problems of production AT&T and Verizon base stations - a hitherto unexplored territory.

LTEye's Technique. LTEye leverages Synthetic Aperture Radar (SAR) to localize LTE signals highly accurately. However, past work on SAR require both transmitters and receivers to share a common reference clock. For e.g., SAR devices on airplanes or satellites both transmit signals and receive their reflections to image the topography of the ground[50]. Unfortunately, when performing SAR between *independent* transmitters and receivers, the measured wireless channel varies both due to position and due to carrier and sampling frequency offset between the transmitter and receiver, as well as any phase noise. As a result, LTEye cannot perform SAR in its standard form.

To overcome this challenge, LTEye leverages mobility: It operates as a 2-antenna MIMO receiver, with one static antenna and another movable antenna. The movable antenna may be mounted on a rotating arm attached to the device, which is the approach taken in our implementation (see Fig. 5-1(a)). LTEye uses its MIMO capability to perform SAR over communication signals, but without frequency offset estimation. Our key idea is that instead of applying the SAR equations to the wireless channel of the moving antenna [50], we apply SAR equations to the *ratio* between the channel of the moving antenna to that of the static antenna (see Fig. 5-1(b)). Taking the ratio of the two channels eliminates any effect of frequency offset since both MIMO antennas experience the same offset relative to the sender. However, since the ratio is between two antennas, one moving and the other static, it preserves how antenna displacement changes the channel of the moving antenna. This allows SAR to safely retrieve the location of the wireless signal source from the ratio, modulo frequency offsets.

■ 5.2 Accurate Indoor Positioning with Zero Start-up Cost

Recent years have seen the advent of new RF-localization systems that demonstrate tens of centimeters of accuracy. However, such systems require either deployment of new infrastructure, or extensive fingerprinting of the environment through training or crowdsourcing, impeding their wide-scale adoption.

In chapter 7, we present Ubicarse [88], an accurate indoor localization system for commodity mobile devices, with no specialized infrastructure or fingerprinting. Ubicarse leverages achieves by enabling handheld devices to emulate powerful radar solutions. It does this using a novel algorithm that builds upon Synthetic Aperture Radar that traditionally uses highly-controlled antenna mobility to perform radar imaging. Ubicarse brings such techniques to commodity handheld devices, where users can simply twist their devices to perform positioning, even along arbitrary and unknown paths. Ubicarse is not limited to localizing RF devices; it combines RF localization with stereo-vision algorithms to localize common objects with no RF source attached to them. We implement Ubicarse on commercial tablets and initial results demonstrate few tens of cm in accuracy in 3-D location of mobile devices, and object geo-tagging.

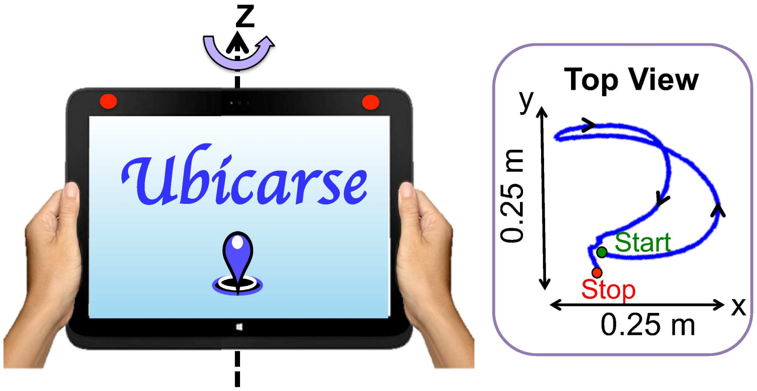


Figure 5-2: Twisting the Mobile Device. The user twists the device about its vertical (z) axis, as shown in the figure on the left. The red circles indicate antenna locations. On the right, we depict the top view of a candidate device trajectory as measured by a Vicon motion capture system.

Ubicarse's Technique. Ubicarse is an indoor geo-location system that enables mobile devices to locate themselves with high accuracy. Past work on localization uses large antenna arrays, that provide high accuracy, but are too bulky to mount on mobile devices. At first, it might seem that one can use Synthetic Aperture Radar (SAR) to directly mimic an antenna array on a handheld device. Unfortunately, SAR in its existing form is unsuitable for handheld devices. To understand why this is the case, note that a moving antenna performs SAR by collecting signal snapshots as it moves along its trajectory, and jointly processing these snapshots to emulate a large antenna array traced out by that trajectory. Therefore for SAR to mimic a particular antenna array geometry, it must first know the position of the moving antenna at every point along its trajectory. Satisfying this requirement is relatively easy in radar systems where the speed and trajectory of the antenna movement are finely controlled [50]. Unfortunately, when a user moves her handheld device, neither its speed nor its trajectory can be accurately controlled. To get a feel for the accuracy required, in 802.11a/n, even a small error of 2 cm in the relative position of the moving antenna leads to an error of 60 degrees in identifying the direction of the source.

One might consider using motion sensors (accelerometer, gyroscope, compass) present in most mobile devices to estimate antenna positions as it moves. Unfortunately, since SAR requires sub-centimeter accuracy in the device position along its trajectory, commercial motion sensors are virtually unusable to measure such fine-grained translation [127] and can at best be used to measure orientation from their gyroscopes.[1] Requiring the user to rotate the device on a perfect arc with zero translation of the device center would preclude measuring this translation but is impractical to enforce.

Ubicarse addresses this challenge by developing a new formulation of SAR that is translation-resilient. Specifically, we allow the user to twist her mobile device about its vertical axis, as shown in Fig. 5-2. Even if the twisting involves unknown trajectories (that include translation), we can accurately compute SAR knowing only the rotation estimate from the gyroscope. To do so, we exploit the MIMO capability of modern wireless cards. Suppose the mobile device has two antennas. The distance between these antennas is fixed independent of how the user moves her device. Thus, whenever the user translates the device, the relative position vector of both its antennas remains the same. In contrast, as the device rotates, the relative position vector of the antennas, also rotates. Leveraging these observations, we develop a new SAR formulation that operates on *relative* wireless channels to estimate the direction of the access point's signal. These relative wireless channels depend purely based on the device's orientation and have no dependency on its exact position or translation. As a result, Ubicarse's formulation of SAR can be performed without prior knowledge of precise device trajectories.

Chapter 7 details Ubicarse's core algorithms in greater detail. It further describes how Ubicarse can combines RF localization with stereo-vision algorithms to localize common objects with no RF source attached to them. Finally, it implements Ubicarse on a HP SplitX2 tablet and empirically demonstrates its accuracy in both 3-D device localization and object geotagging in complex indoor settings.

■ 5.3 LTEye and Ubicarse in the context of Prior Indoor Positioning Literature

Prior to Ubicarse, RF localization past work has had three main approaches for indoor localization: The first require deploying specialized infrastructure to perform accurate indoor localization, e.g. acoustic [154, 107], RFIDs [179, 178], specialized access points [82], and antenna arrays [188]. The second mandates signal fingerprinting [16, 150, 197] either

[1] This is because accelerometers report the net acceleration of the device including gravity, which must be subtracted out and the result integrated twice to obtain translation.

in a training phase or via crowdsourcing [142, 195, 104]. The third advocate modeling the wireless signal instead of measurements [97, 28, 65], but at the expense of accuracy. In other words, past work forces a trade-off in either deployment effort (e.g. new infrastructure, or fingerprinting) or compromises on accuracy (several meters).

In contrast to these systems, our work proposes systems that achieve tens of centimeter of accuracy without requiring fingerprinting or new infrastructure. Our approach emulating large antenna arrays using only simple, commodity wireless devices. Specifically, our approach explores two aspects of the indoor positioning design space: Do we want to deploy indoor position at the infrastructure or user devices? LTEye explores the former, by proposing simple hardware modifications to wireless infrastructure (LTE femtocells or receivers) without modifying user devices. Ubicarse considers the latter, running a software-only system on user devices that can be run on demand when users twist their tablets, using existing unmodified Wi-Fi infrastructure.

In addition, we believe that Ubicarse and LTEye complement solutions that perform indoor positioning using other technologies (besides Wi-Fi or LTE), when such infrastructure is available. These systems include those that employ acoustic beacons [70, 135, 139, 154], cameras or light sensors [185], infrared [183], RFIDs [179, 178] and bluetooth beacons [140, 12]. We believe such technologies can complement our work and further improve accuracy, wherever the infrastructure they require is available.

LTEye and Ubicarse bring techniques from the world of radar systems to wireless devices. They are most closely related to past work on RF-based localization that employs synthetic aperture radar (SAR) [179, 50]. Our solutions build on this past work but differs in that it extends SAR to operate over communication signals exchanged between an RF transmitter and a receiver. This contrasts with the current approach for SAR, which is limited to backscatter and radar signals, where the transmitter and receiver are a single node with no Carrier Frequency Offset (CFO) or Sampling Frequency offset (SFO). In addition, Ubicarse proposes a new formulation of SAR that applies to Wi-Fi devices manually twisted by users. Unlike past work on SAR where radar devices are rotated mechanically along regular and completely known trajectory, Ubicarse's formulation of SAR functions despite arbitrary and unknown trajectories of handheld devices.

Finally, we note that LTEye is also related to past measurement studies [77, 76, 157] and tools [145, 169, 42, 51] that analyze LTE cellular networks. Such studies focus on the higher layers of the stack, e.g., TCP throughput, transfer delay, and power usage. In contrast, LTEye focuses on the LTE radio layer; it provides fine-grained temporal and spatial information and does not require traces from the provider.

Chapter 6
LTE Radio Analytics Made Easy and Accessible

Cellular service has become an integral part of our life. Yet as users and researchers, we have little visibility into the performance and real problems of these networks. Even the little information we have is primarily from trace analysis sanctioned by mobile operators [76, 77]. The lack of open and transparent access into the operation and inefficiencies of the cellular physical layer limits our ability as researchers to contribute effectively to the development of these networks. It also limits the ability of policy makers to independently verify operators' claims of spectrum needs, and make informed decisions on licensed vs. unlicensed spectrum.

The need for increased visibility into the cellular PHY-layer is further emphasized by three recent trends. First, cellular deployment is moving towards small, femto, and pico cells [156], many of which will be deployed by a user to cover her home or small business. As a result, cellular operators no longer have full control over their LTE deployments, and struggle to understand the propagation and interference patterns affecting their service, particularly in indoor settings. Open, cheap, and ubiquitous radio monitoring can help deal with the challenging propagation patterns introduced by small cells. Second, the rise of LTE-based M2M applications motivates a more open access to LTE signal-based analytics. For example, Walmart, Home Depot, or Disneyland may leverage LTE signals and recent RF-based localization techniques to track how clients navigate their space and obtain business analytics. Also, oil and gas companies that deploy LTE-supported seismic sensors [31] can leverage access to LTE radio propagation to better plan their sensor network and debug connectivity problems. Third, the FCC plans to repurpose certain bands (e.g., 3.5 GHz) for multi-tier spectrum sharing, including LTE small cell deployments [43]. Operating in a shared spectrum naturally fits with an open model for signal monitoring and analysis, where all the entities sharing the spectrum can better understand the problems and cooperate to find solutions.

All of these reasons motivate a more open access to the cellular PHY layer, particularly LTE. In this chapter, we ask the following question: Can we access the cellular PHY-layer of today's LTE deployments, without support from mobile operators? Specifically, can we do this without requiring access to private user data or encrypted LTE channels? If such a service exists, it could enable better deployment of femto cells and repeaters, more businesses built over LTE networks, better informed spectrum policies, more efficient sharing of newly released bands, and an overall rise in transparency in an industry that is a major part of the world economy.

We introduce LTEye, an open platform for monitoring and analyzing the LTE PHY layer. LTEye is a passive sniffer that runs on off-the-shelf software radios (e.g. USRPs). It does not require provider support, and hence can serve end users, researchers, policy makers, or mobile broadband providers. LTEye extracts per-user analytics purely from the LTE control channels that contain meta-deta on uplink and downlink transmissions, without accessing private data or system parameters from encrypted data channels. Specifically, it tracks individual users based on their temporal PHY-layer IDs, without requiring or exposing their private information. It then intelligently links these IDs across control messages to generate transmission records for each user. Each record reports the transmission's resource utilization, modulation and coding rate, and frame loss rate. LTEye also records the wireless channel it perceives from base stations and mobile users within radio range. These channels are used to accurately monitor the 3-dimensional physical location of the users. LTEye maintains these records in a database called LTEyeDB. It processes the records to generate two dimensions of fine-grained analytics: temporal analytics, to track LTE performance over time, and spatial analytics, to characterize LTE service across spatial locations.

We implemented LTEye on USRP software radios [41]. We deployed LTEye in four locations in our campus to compare the temporal performance of two major LTE providers: AT&T and Verizon. Our results revealed several inefficiencies in these networks. First, both providers deploy a Frequency Division Duplexing scheme, which uses independent equally sized frequency bands for uplink and downlink traffic. However, LTEye reported that for both providers, the average utilization of downlink resources (25.2% - AT&T, 58.2% - Verizon) far exceeded that of uplink resources (0.6% - AT&T, 2.6% - Verizon). While it is expected that the downlink is higher in demand, our results reveal that the LTE uplink is a remarkable 20 to 40 times less utilized than the downlink. LTEye's analytics can therefore help policy makers encourage operators to adopt revised LTE standards that allow more prudent allocation of resources to the uplink and downlink[1], without relying on data from providers themselves to make the case.

Second, LTEye localized certain spots in our campus, where Verizon cellphones suffer poor link quality and often switch to 3G, despite reporting high signal power from the LTE base station. To investigate this, we moved our LTEye sniffers to these spots and found that they experienced high inter-cell interference (about 27 dB) from as many as five different base stations. To make matters worse, many of these base stations used overlapping channel estimation pilots, significantly impacting the decodability of these transmissions. These results help end-users identify poor placement of femto cells that cause such interference. Further, they benefit cellular providers themselves because they reveal interference problems that end-users face in indoors, hitherto inaccessible to providers. Interestingly, some of these PHY-layer inefficiencies may be unknown even to the operators as they are part of the PHY-layer implementations adopted by the base station vendors.[2]

LTEye also benefits researchers by providing deep insights into the PHY-layer protocols deployed by cellular providers. While the LTE standard spells out much of the PHY layer, the choice of rate adaptation algorithm is still left to individual operators. To gain insights into this algorithm, we analyzed LTEyeDB records of an AT&T base station in our locality. We found that even for static users with completely coherent channels and stable SNRs, the modulation and coding scheme changes significantly even between adjacent transmissions. More interestingly, the average modulation and coding of frames sent to a user changes, based not only on her wireless channels, but also on the network state as

[1] E.g. Asymmetric Carrier Aggregation [83] in LTE Advanced (3GPP Release 10) allows downlink resources to exceed the uplink.

[2] We confirmed this privately with some base station vendors.

a whole. Specifically, if the network is scarcely utilized, the base station transmits to the user conservatively at low modulation on average, even if the wireless link is stable and supports much higher modulation. In contrast, as network utilization increases, under identical SNRs, the base station steadily increases its modulation to support more aggressive data rates to the user. Such analytics on the performance and design choices of today's cellular operators help researchers design better LTE protocols.

LTEye enables businesses and network administrators to continuously monitor the spatial locations of mobile users, and build a geographic heatmap of LTE coverage and performance within their facility. However, accurately localizing mobile users purely based on their LTE signals is a challenging task. This is because past work on accurate indoor localization proposes two classes of solutions that are ill-suited to LTE networks: First, localization using antenna arrays [188, 82] requires large bulky arrays, owing to the relatively low frequencies of LTE signals. Second, recent localization techniques using synthetic aperture radar (SAR) are less bulky, but are limited to signals transmitted and received by the same node (e.g. radar [50] or RFID backscatter [179] systems) and therefore do not apply to LTE signals. LTEye provides the best of both these solutions by extending SAR localization techniques to operate over communication signals as opposed to backscatter or radar signals. It also introduces a novel technique to handle errors due to multipath by identifying the shortest (or most direct) path.

Our evaluation of LTEye's spatial analytics in large indoor environments reveals a median accuracy in 3D localization of mobile users of 61 cm in line-of-sight and 85 cm in non-line-of-sight settings. Further, we visualize the LTE performance of the mobile users across locations, helping building managers find optimal locations for relays or femtocells.

Contributions: This chapter presents the following contributions:

- We present LTEye, the first open platform to monitor and analyze per-user LTE PHY performance at fine temporal and spatial granularity.
- LTEye employs a new technique to identify and track individual users at the LTE PHY layer in a robust manner, without help from operators, and without requiring or exposing private user information.
- LTEye develops an innovative technique for accurate localization of users based on their LTE signals. This involves extending synthetic aperture radar (SAR) to operate over communication signals as opposed to backscatter and radar signals, and a

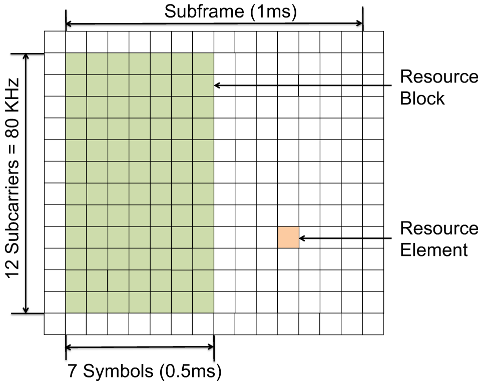


Figure 6-1: Resource Grid. The Resource Grid is divided into multiple Resource Blocks.

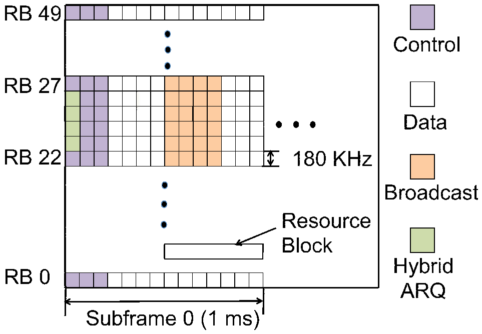


Figure 6-2: Physical Channels. Physical Channels are mapped to well defined Resource Elements in the Grid.

novel technique for identifying the shortest and most direct path in the presence of multipath.

- LTEye's evaluation on AT&T and Verizon LTE deployments reveal deep insights on the inefficiencies, utilization patterns, and differences between these providers. LTEye also provides heat maps to characterize LTE performance across indoor locations, without GPS support.

■ 6.1 LTE Primer

In this section, we briefly introduce LTE concepts relevant to this chapter. LTE networks are divided into multiple geographical regions called cells. Each cell contains a cellular base station that serves multiple mobile users. We focus on Frequency Division Duplexing, the mode of LTE most widely used by cellular operators. This LTE mode uses different dedicated carrier frequency bands for uplink and downlink transmissions. Hence, each base station uses a pair of frequency bands to communicate with users in its cell.

(a) Radio Resources. LTE's uplink and downlink transmissions are based on OFDM. While medium access and resource sharing is largely distributed in typical OFDM-based systems such as Wi-Fi, LTE centralizes much of resource allocation at the base station. Specifically, base stations divide radio resources into multiple frames over time, each containing ten subframes, spanning 1 ms each. Resources in each sub-frame are divided both along time and frequency as a 2-D time-frequency grid, as in Fig. 6-1. Each cell in the

Downlink LTE Channels	
Name	*Description*
PBCH	Physical Broadcast Channel: Carries general information about the cell, like number of antennas on the base station and total bandwidth.
PDCCH	Physical Downlink Control Channel: Sends downlink control messages, e.g. for resource allocation on uplink/downlink.
PDSCH	Physical Downlink Shared Channel: Holds downlink data meant for users in resource blocks indicated by the PDCCH.
PHICH	Physical Hybrid-ARQ Channel: Contains positive or negative acknowledgments for uplink data.
Uplink LTE Channels	
PUCCH	Physical Uplink Control Channel: Holds control information from users to base stations, e.g. ACKs, channel reports and uplink scheduling requests.
PUSCH	Physical Uplink Shared Channel: Mainly carries uplink data in resource blocks indicated by the PDCCH.

Table 6-1: LTE Channels. Details the name and function of PHY-layer channels, as defined in LTE standards [6].

grid, called a resource element corresponds to one OFDM sub-carrier (15 KHz) over the duration of one OFDM symbol (66.7 μs).

The key task of an LTE base station is to allot both uplink and downlink resources between different users along both time and frequency. It allocates resources to users at the granularity of *resource blocks*, each spanning 0.5 ms (i.e., half a sub-frame) by 180 kHz (i.e., 12 sub-carriers). To combat frequency-selective fading, the assignment of resource blocks to users both on the uplink and downlink is not fixed, but hops from transmission to transmission.

(b) Physical Layer Channels. As LTE centralizes PHY-layer control, base stations need to transmit both data and control information to the users. To this end, LTE partitions network resources into well defined channels, each responsible for different types of information. These channels are mapped to well-known resource elements of the grid, as shown in Fig. 6-2.

Broadly, LTE base stations use four main channels on the downlink: (1) A data channel to send users their downlink data. (2) A control channel to allocate network resources. (3) A broadcast channel for new users to learn system parameters. (4) A hybrid-ARQ channel to send ACKs to the users. Similarly, the mobile users on the uplink are allocated data and control channels to transmit their uplink data and control messages. Table 6-1 describes these channels in greater detail.

We point out here that the downlink control channel bears rich information on the LTE

PHY-layer. Specifically, it contains multiple *downlink control information messages* in which the base station allocates resource blocks to specific users for every transmission on either the uplink or downlink. In addition, the control information also specifies the modulation and coding to be used in these blocks. Upon hearing a control message, a user accesses the relevant data channel to send (receive) her uplink (downlink) transmission.

(c) PHY User Identifier. The LTE PHY refers to each mobile user using a temporary unique ID called the Cell Radio Network Temporary Identifier (C-RNTI). The C-RNTI reveals no private information about the user. It is local to the users's serving cell, and is assigned when she enters the cell via a higher-layer connection establishment procedure[7].

A user continues to have the same C-RNTI as long as she is in the same cell and is not idle for more than the pre-configured tail timer value. The timer value is typically a few seconds to tens of seconds (12 sec in the measurement result of [76]). Hence, the C-RNTI assigned to a user may change quite often if it transmits sporadically.

■ 6.2 LTEye

LTEye is an open platform to analyze LTE radio performance. Its design aims to satisfy the following key attributes:

- *Provider-Independent:* LTEye does not require any information or support from mobile providers. Thus, LTEye allows end-users, policy makers and third parties to make a fair assessment and comparison of the service quality of different providers, without relying on information furnished by the providers themselves.
- *Off-the-Shelf:* LTEye must be built on low-end off-the-shelf components, such as standard laptops and software radios. This makes LTEye more accessible to end-users and easy to be deployed in large numbers.

■ 6.2.1 System Architecture

LTEye operates as a passive 2-antenna MIMO receiver. LTEye's architecture is a pipeline of two components: (1) the LTE Logger, and (2) the Data Analyzer. (see Fig. 6-13)

(a) LTE Logger: The LTE Logger sniffs on the LTE control channels to generate transmission records. The logger begins by listening to the broadcast channel to gather system

Field	Description	Format
`Time`	Transmission Time	32-bit timestamp
`C-RNTI`	PHY-layer ID	16-bit Sequence
`UL/DL`	Uplink or Downlink	0/1
`nrb`	Number of Resource Blocks	0 to N_{RB}
`alloc`	Bit-Map of Resource Blocks	N_{RB} bits
`MCS`	Modulation and Coding Scheme	5 bits
`isAcked`	Acknowledgment	±1 (ACK/NACK)
`UE-channel`	Channel from user (if U/L)	$N_{sc}N_{rx}$ Complex Floats
`BS-channel`	Channel from Base Station	$N_{sc}N_{tx}N_{rx}$ Complex Floats
`SNR`, `SINR`	SNR and SINR of Base Station	Floating Point

Table 6-2: Fields of LTEyeDB. Where N_{RB}: Number of resource blocks, N_{tx}: Number of base-station antennas, N_{rx}: Number of sniffer antennas, N_{sc}: Number of OFDM sub-carriers.

parameters. It then sniffs the downlink control channel and performs LTE decoding, i.e., it demodulates the OFDM symbols, applies de-interleaving, de-scrambling and convolutional decoding to extract the actual bits of the downlink control messages.

The logger reads each of these control messages to populate LTEyeDB, a database of LTE information records tagged with their transmission time. Each transmission record consists of several fields retrieved from the control messages, as listed in Table 6-2. Many of these fields help characterize network performance and utilization, both for the user over time, and for the network as a whole, if viewed in aggregate. In addition, notice two important fields in the records: (1) Each record is indexed by the user's C-RNTI (i.e. her PHY-layer user ID), which are key to link multiple records belonging to the same user. (2) Records maintain the uplink channels seen by the LTEye sniffer over time. In §6.4, we show how these channels are essential to localize users.

(b) Data Analyzer: The data analyzer processes the records in LTEyeDB to extract fine-grained analytics on both cellular base-stations and individual mobile users. It extracts two types of analytics: temporal analytics which describe LTE PHY metrics as functions of time (e.g., the per-user resource allocation over time), and spatial analytics which describe position-dependent RF metrics (e.g., the user location and the observed multipath effects). Sections §6.3 and §6.4 discuss both types of analytics in detail.

■ 6.3 Enabling Temporal LTE Analytics

In this section, we describe some of the challenges in obtaining temporal analytics, without access to private user information or system parameters in encrypted data channels.

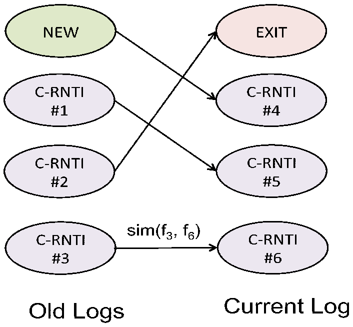


Figure 6-3: Mapping C-RNTI. Mapping C-RNTI from old logs to the current log.

Uniquely Identifying Users: To extract per-user temporal analytics, LTEye needs to uniquely identify each user (i.e mobile device) served by the base station. We do not know the phone numbers of users, but can instead use the series of C-RNTIs (i.e. PHY-layer user IDs) assigned to them. One option is to require LTEye to sniff the LTE channel continuously for an extended time to catch each user at the time she joins the cell and capture her C-RNTI assignment. Unfortunately this option has two limitations: First, the C-RNTI assignment is often transmitted from higher layers in the encrypted downlink data channel [7]. Second, low-end off-the-shelf equipment is unlikely to be able to continuously monitor and decode the LTE channel in real-time [15]. Instead, LTEye sniffers should be able to periodically sniff the channel and obtain representative snapshots of the system. Hence, we need LTEye to uniquely identify all users, including those who joined the cell even when LTEye is not sniffing (i.e., LTEye did not hear their C-RNTI assignment).

To address this problem, we observe that a user's C-RNTI is used to scramble her control information on the downlink control channel. Specifically, recall from §7.1 that the control channel transmitted by the base station consists of multiple downlink control information messages, for different users. At the end of each message is a 16-bit sequence, which is the XOR of the checksum of the control message with the user's C-RNTI. Traditionally, a user de-scrambles this sequence by her C-RNTI to retrieve the checksum and validate correctness of the control information. In contrast, LTEye performs the *opposite* operation to retrieve the C-RNTI: It decodes each control message in the log including their corresponding scrambled checksums. For each of these packets, LTEye reconstructs the expected checksum and XORs them with the scrambled checksum to recover the C-RNTI. Of course, it is important to verify if the control messages and C-RNTIs that are decoded are actually correct. To do this, LTEye convolutionally re-encodes the retrieved control information message and compares it against the original coded control message to obtain the number of bit errors. It then discards control messages and C-RNTIs that report bit errors beyond a few bits.[3] Hence, our solution enables LTEye to map a C-RNTI to a user even if it was assigned when LTEye is not sniffing the channel.

Tracking User IDs: LTEye's second challenge is to map C-RNTIs between logs. Recall from §7.1 that a user's C-RNTI may be re-assigned if she is idle beyond a few seconds. Further, the same C-RNTI can map to multiple users over time. Hence, while the C-RNTI is a natural PHY-layer user ID, LTEye must recognize when a user's C-RNTI changes to find the list of C-RNTIs mapped to her.

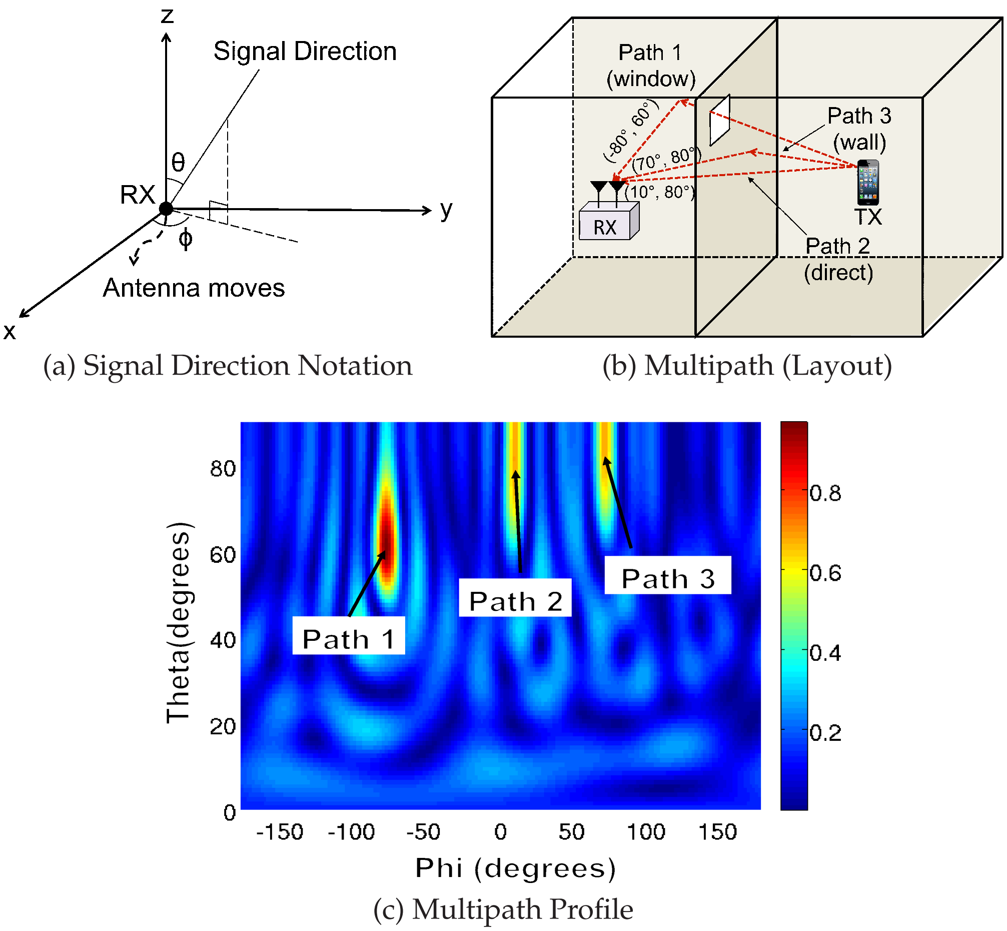
LTEye addresses this challenge by formulating it as a matching problem. The goal of this problem is to map C-RNTIs in the current (most recent) log produced by the LTE logger with the C-RNTIs in prior logs as shown in the bi-partite graph in Fig. 6-3. In addition, the graph has two additional nodes: EXIT and NEW, which account for C-RNTIs in the old log that have "exit" the system, and C-RNTIs in the current log that are "new" to the cell, respectively. The weights in this graph must capture the similarity between the users associated with each pair of C-RNTIs. Specifically, we associate with each C-RNTI #i an RF fingerprint f_i that includes metrics such as the user's location (extracted by our localization method described in §6.4), its SNR and multi-path characteristics. We can then assign a weight to each edge (i, j) in the graph by the similarity metric between these fingerprints $sim(f_i, f_j)$. In §6.4.3, we design effective RF fingerprints and similarity metrics based on spatial analytics.

Given the graph and weights, we can now solve this matching problem using the standard Hungarian Algorithm[86].[4] The resulting matching either maps a C-RNTI to a user in a prior log, or identifies her as a new user.

Extracting System Parameters: LTEye needs to *reverse-engineer* several PHY parameters, otherwise available to users via encrypted data channels. For e.g., the LTE standard allows several possible formats for downlink control information, each spanning multiple lengths [8]. Exhaustively searching for each of the possible format-length pairs in all downlink control messages is highly expensive. Fortunately, operational LTE base stations use only a small subset of these formats for all users (only three possible formats for AT&T and

[3] In our experiments, LTEye correctly retrieved 99.5% of C-RNTIs across locations.

[4] We modify the algorithm to allow multiple C-RNTIs to map to NEW/EXIT simply by replicating these nodes. Edges at NEW/EXIT are weighted by a minimum threshold similarity.



(a) Signal Direction Notation

(b) Multipath (Layout)

(c) Multipath Profile

Figure 6-4: Spatial Analytics. (a) Definition of (ϕ, θ) in 3-D space. (b) Depicts an example layout of a transmitter and receiver in two rooms separated by a wall. The signal has three main paths: Path 1 is the strongest, and reflects from the ceiling through a window. Path 2 is the direct path penetrating a wall. Path 3 is the weakest reflecting at the farthest wall. The figure labels (ϕ, θ) for each path. (c) Simulated multipath profile for (b) has peaks for each path at expected (ϕ, θ). The height of the peaks (in shades of red) corresponds to the relative power of the corresponding paths.

Verizon). As a result, LTEye learns the possible list of formats and lengths using the first few control messages to greatly reduce the search-space of formats for subsequent control information.

■ 6.4 Enabling Spatial LTE Analytics

The core of LTEye's spatial analytics is the ability to localize an LTE source. To this end, LTEye performs the following functions:

- *Extracts the multipath profile of an LTE signal:* First, LTEye extracts the multipath profile of a signal, which measures the power received along each spatial angle, i.e., along each signal path. Fig. 6-4(c) is an example multipath profile, where the signal traverses three paths causing three local maxima in the graph.

- *Identifies the direct path from the source:* Once LTEye has the multipath profile, it is natural to try to identify which path is in direct line of sight to the source (e.g., path 2 in Fig. 6-4(b)). Finding the direct path is an important step in localizing the source. Note that in some cases the direct path may be completely blocked and absent from the multipath profile. Our objective is to identify the direct path provided it exists in the multipath profile.
- *Localizes the source of the LTE signal:* Once LTEye finds the direct path to the source, it can localize the source to within a specific spatial direction. We can then deploy multiple LTEye sniffers to locate the source at the intersection of the direct paths as seen from these sniffers.

Past work on RF-based localization takes two approaches to build multipath profiles: The first approach, shown for Wi-Fi, uses an antenna array to steer its beam spatially and identify the power along each spatial direction [188]. However, LTE runs at much lower frequencies than Wi-Fi (700 MHz as opposed to 2.4 GHz), and hence an LTE antenna array will be 4 to 5 times more bulky than a comparable Wi-Fi array. The second approach uses synthetic aperture radar (SAR) [179], which uses a single movable antenna to emulate a virtual array of many antennas. As the antenna moves, it traces the locations of antenna elements in a virtual array.

SAR has traditionally been used in radar and RFID localization as it assumes backscatter signals where the transmitter and receiver are the same node and therefore have no carrier frequency offset (CFO) relative to each other. Hence, changes in the channel as the antenna moves are a function only of the antenna's location. In contrast, LTE signals are exchanged between an independent transmitter and receiver, with non-zero CFO. As the antenna moves, the channel changes both due to CFO and antenna movement. One option is to estimate and correct the CFO. Unfortunately this solution is fragile since any residual error in CFO estimation accumulates over the duration of movement and causes large localization errors. In the following section, we explain how we perform SAR over LTE signals without CFO estimation to realize the three functions in the beginning of this section.

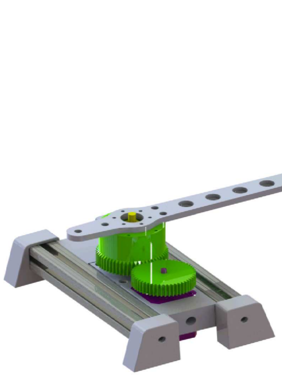


Figure 6-5: Prototype. Prototype of the rotating antenna on a LTEye sniffer.

■ 6.4.1 SAR over LTE Signals Using Channel Ratios

LTEye operates as a 2-antenna MIMO receiver, with one static antenna and another movable antenna. The movable antenna may be mounted on a rotating arm attached to the device, which is the approach taken in our implementation (see Fig. 6-6(a)). The advantage of this approach is that it provides the 3-D spatial direction (i.e. both the azimuthal angle ϕ and polar angle θ) of the various signal paths as shown in Fig. 6-4(a). Alternatively, the antenna may slide back and forth on an arm fixed to the body of the device.

LTEye uses its MIMO capability to perform SAR over communication signals, but without frequency offset estimation. Our key idea is that instead of applying the SAR equations to the wireless channel of the moving antenna [50], we apply SAR equations to the *ratio* between the channel of the moving antenna to that of the static antenna. Taking the ratio of the two channels eliminates any effect of frequency offset since both MIMO antennas experience the same offset relative to the sender. However, since the ratio is between two antennas, one moving and the other static, it preserves how antenna displacement changes the channel of the moving antenna. This allows SAR to safely retrieve the multipath profile of the signal from the ratio, modulo frequency offsets.

Next, we mathematically show the validity of the above technique. Suppose that the receiver, placed at the origin, wants to measure the power of the signal $P(\theta, \phi)$ received from an independent transmitter along a spatial direction specified by the polar angle θ and azimuthal angle ϕ (see Fig. 6-4(a)). According to the SAR formulation, this quantity can be measured as: [45, 131]:

$$P(\theta,\phi) = |h(\theta,\phi)|^2, \text{ where } h(\theta,\phi) = \sum_t a_f(t,\theta,\phi)h(t) \tag{6.1}$$

Here, $h(t)$ is the wireless channel to the moving antenna at time t, assuming zero frequency offset between the transmitter and receiver. The quantity $a_f(t,\theta,\phi)$ captures the relative motion of the transmitter and receiver, and is independent of the wireless channels. For e.g., if the antenna moves along a straight line (i.e., linear SAR) $a_f(t,\theta,\phi)$ is defined as: $a_f(t,\theta,\phi) = e^{-j\frac{2\pi f}{c}x(t)cos(\phi)}$, where $x(t)$ is the antenna location at time t, and f is the frequency of the signal [179]. Similarly, if the antenna rotates with radius r (i.e., circular SAR) $a_f(t,\theta,\phi) = e^{-j\frac{2\pi f}{c}rcos(\phi-\phi_0(t))}$, where $\phi_0(t)$ is the angular position of the antenna at time t (see Fig. 6-6(b)).

Past work on SAR require both transmitters and receivers to share a common reference clock. For e.g., SAR devices on airplanes or satellites both transmit signals and receive their reflections to image the topography of the ground[50]. Consequently, the measured channel $\tilde{h}(t)$ at the moving antenna is independent of frequency offset, i.e., $\tilde{h}(t) = h(t)$.

Unfortunately, when performing SAR between *independent* transmitters and receivers, the measured wireless channel $\tilde{h}(t)$ varies both due to position and due to carrier and sampling frequency offset between the transmitter and receiver, as well as any phase noise. In particular, we denote:

$$\tilde{h}(t) = h(t)e^{j\psi(t)} \tag{6.2}$$

Where $\psi(t)$ denotes the phase accumulated due to any carrier frequency offset, sampling frequency offset or phase noise between the reference clocks of the transmitter and receiver until time t. Thus, the key challenge to measure the power of the signal along any spatial direction, $P(\theta,\phi)$, as in the SAR Eqn. 6.1, is to eliminate this accumulated phase.

To resolve this challenge, LTEye is built on receivers that have at least two antennas: a static antenna, and a mobile antenna that moves along a known path. Let $\tilde{h}_1(t)$ and $\tilde{h}_2(t)$ denote the measured wireless channels from a given transmit antenna to a mobile and static antennas respectively (see Fig. 6-6(c)). As both antennas are connected to the same reference clock, they both accumulate the same phase $\psi(t)$ until time t. Hence, from

Eqn. 6.2, we have:

$$\tilde{h}_1(t) = h_1(t)e^{j\psi(t)}, \qquad \tilde{h}_2(t) = h_2(t)e^{j\psi(t)} \tag{6.3}$$

where $h_2(t) \approx h_2$ is relatively constant over a short duration since antenna-2 is static. Hence, the ratio of the wireless channels is: $\tilde{h}_r(t) = \frac{\tilde{h}_1(t)}{\tilde{h}_2(t)} = \frac{1}{h_2}h_1(t)$; that is, the channel ratio is a constant multiple of the moving antenna channel without the phase accumulation from frequency offset or phase noise. Thus, we can perform SAR as in Eqn. 6.1 by substituting the channel ratio $\tilde{h}_r(t)$ for the value of $h(t)$. Hence, LTEye allows a wireless receiver to perform SAR over LTE signals without frequency offset or phase noise estimation.

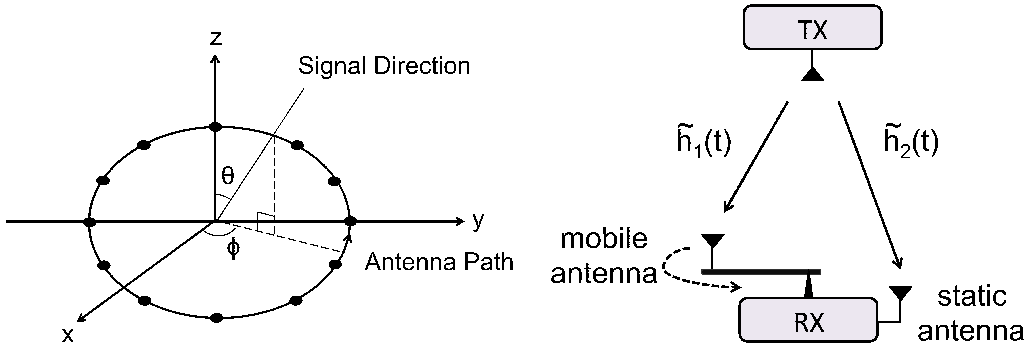
Finally, we make a few important observations:

- The above approach readily extends to OFDM / OFDMA. Specifically, let $h(\theta, \phi)$ $= \sum_f \sum_t a_f(t, \theta, \phi) h_{r,f}(t)/h_{r,f}(0)$ in Eqn. 6.1, where the quantity $\tilde{h}_{r,f}(t)$ on subcarrier f of the OFDM signal is the ratio of the frequency (Fourier) Domain channels $\tilde{h}_1(t)$ and $\tilde{h}_2(t)$ measured on that subcarrier.[5]
- Our solution is resilient to movement of the transmitter that can be neglected relative to the movement of the rotating antenna. LTEye's rotation speed is chosen to deal with typical dynamism in indoor settings, e.g. walking speeds. However, one can easily detect and exclude fast moving transmitters by checking if the channel of the static antenna $\tilde{h}_2(t)$ is coherent over a rotation of the moving antenna, using the coherence metric in §6.6.3.
- While this chapter applies our solution to LTE, our technique can be extended to apply to Wi-Fi and other such technologies.

■ 6.4.2 Identifying the Direct Signal Path

In this section, we describe how LTEye separates the direct path from the reflected paths in a multipath profile reported by SAR so as to localize the transmitter. Intuitively, the direct path is the shortest path among all paths traversed by the signal (even if the direct path is completely blocked, the shortest path is the path closest to the direct path). Thus, one

[5] Note that this is robust to frequency hopping by LTE users.



(a) Circular SAR (b) Channel Notation

Figure 6-6: SAR. (a) Circular Synthetic Aperture Radar (b) Depicts $\tilde{h}_1(t)$ and $\tilde{h}_2(t)$, the measured wireless channels from a transmit antenna to the mobile and static antennas respectively

may identify the direct path (or the path closest to the direct path) by measuring the delay difference between the various paths in the multipath profile of the signal.

Directly measuring time delays (e.g. by correlating with known pilot signals), however is not sufficiently accurate. Specifically, LTE receivers have a channel bandwidth of 10 MHz. However, electromagnetic waves travel at the speed of light. Hence an error of even one time sample for a 10 MHz sampling rate (i.e., each time sample spans $0.1\mu s$) translates to an error in path lengths of 30 m.

Below we explain how LTEye can measure *sub-sample delay differences* between the signal paths. The key idea is to exploit that delay in time translates into phase rotation in the frequency domain. Since LTE signals use OFDM, a time delay of the signal translates into phase rotation in the OFDM subcarriers. Yet, different OFDM subcarrier rotate at different speeds – i.e., higher OFDM frequencies rotate faster than lower frequencies. In fact, the phase rotation of a particular OFDM subcarrier f_i as a result of a delay τ is $\psi_i = 2\pi f_i \tau$. Thus, for each subcarrier, the difference in delay between two paths, p and q, for a particular subcarrier is:

$$\tau = \frac{\psi_{p,i} - \psi_{q,i}}{2\pi f_i} \tag{6.4}$$

One may also average across subcarriers to improve robustness to noise. Multiple subcarriers can also resolve ambiguity if $|\psi_{p,i} - \psi_{q,i}| > 2\pi$ by correcting discontinuities in $\psi_{p,i} - \psi_{q,i}$ across frequencies. Thus, to identify the shortest path in a multipath profile, LTEye does the following:

- First, it computes the phase of the channel for each subcarrier for each path separately. This can be done using the fact that the SAR formulation defines $h(\theta, \phi)$, which provides not just the power, but also the phase of the channel component along each spatial direction (see Eqn. 6.1). Hence, we can simply measure the phase of this path as $\psi = arg[h(\theta, \phi)]$, for each OFDM subcarrier.
- Second, it computes the delay difference between each pair of paths using the Eqn. 6.4 above. It then identifies the shortest path as the one with least delay.

Once the direct path is found, the source is localized along this path. Complete localization can be performed using multiple LTEye receivers and intersecting their direct paths. If the direct paths do not intersect, the best estimate is the point minimizing total distance (or equivalently, delay) to all LTEye receivers. This point can be found using a simple geometric optimization omitted for brevity.

Next, we show how the multipath profile reported by SAR allows us to compute unique RF fingerprints for the users.

■ 6.4.3 Measuring RF Fingerprints with Multipath Profiles

As described in §6.3, LTEye tracks the different C-RNTIs (PHY-layer user IDs) assigned to mobile users between logs. To this end, LTEye employs RF fingerprints [189, 179] to map C-RNTIs between logs. LTEye defines a user's fingerprint as the set of observed multipath profiles at each LTEye sniffer (See §7.3). The key advantage of this fingerprint is that it captures the user's location, multipath, and SNR, as perceived from LTEye sniffers. To measure similarity of two fingerprints, we employ dynamic time warping (DTW[144]), a technique that has recently been applied to capture similarity of two multipath profiles[179]. Given any two multipath profiles, DTW returns a cost function that varies inversely with their similarity. Hence, LTEye defines the similarity metric between two RF fingerprints as the inverse of the total DTW cost function[179] between each pair of profiles in the fingerprints.

One might wonder if LTEye's fingerprint matching algorithm scales, given that a cell may serve a large number of users. Fortunately, while LTE cells can serve hundreds of users, we observed that only a small fraction of these users (about 4%) are re-assigned C-RNTIs between two logs.[6] Further, while capturing C-RNTI reassignments is essential to

[6] This is on average 1-2 users for AT&T and 3-4 users for Verizon.

track individual users over time, it does not alter aggregate radio analytics in a statistically significant way, given that 96% of users in a cell retain their C-RNTI between logs.

6.5 Implementation

We implemented LTEye on USRP N210 software radios[41] and WBX daughter-boards. The USRPs receive in the 700MHz frequency range at a bandwidth of 10 MHz on uplink and downlink channels corresponding to either AT&T (734-744MHz) or Verizon (746-756MHz). We implement an OFDMA receiver for LTE signals that interfaces directly with the USRP Hardware driver (UHD).

To obtain temporal analytics, we decode the downlink control channel in a pipeline of two modules: The first module in C++ performs synchronization, channel estimation and QPSK demodulation. It logs the demodulated soft bits received over one second into a file, and repeats this process every three seconds. The second module in MEX (C++) and Matlab reads the file and performs descrambling, de-interleaving and convolutional decoding. It validates the downlink control information messages by only admitting those passing the convolutional decoder with very high confidence, as described in §6.3. It then processes the control messages to get per-user LTEyeDB records for uplink and downlink traffic, as in §6.2.1.

To obtain spatial analytics, we build prototype LTEye sniffers, each containing two USRPs connected to an external clock. We mount the antenna of one of the USRPs on a lightweight rotating arm fabricated by a 3D printer, with an adjustable radius of 15-30 inches, as shown in Fig. 6-6. The arm is driven by an off-the-shelf stepper motor rotate at 30-120 rotations per minute.[7] We use an Arduino UNO board to rotate the stepper motor accurately at a constant speed and provide real-time feedback on the position of the rotating antenna. We implement a C++ module to use these positions and channel measurements, as in §6.4, to localize the users.

We evaluate LTEye's spatial analytics using five LTEye sniffers in multiple indoor testbeds, in both line-of-sight and non-line-of-sight settings. We employ ten Samsung Galaxy Note LTE smart phones as users. Unless specified otherwise, each user communicates over LTE with a mix of varying traffic patterns including browsing activity, file

[7] We plan to support higher torque in future iterations of the device.

transfer, Skype calls, and video streaming.

6.6 Results on Radio Analytics

In this section, we perform an extensive evaluation of LTEye's temporal and spatial analytics:

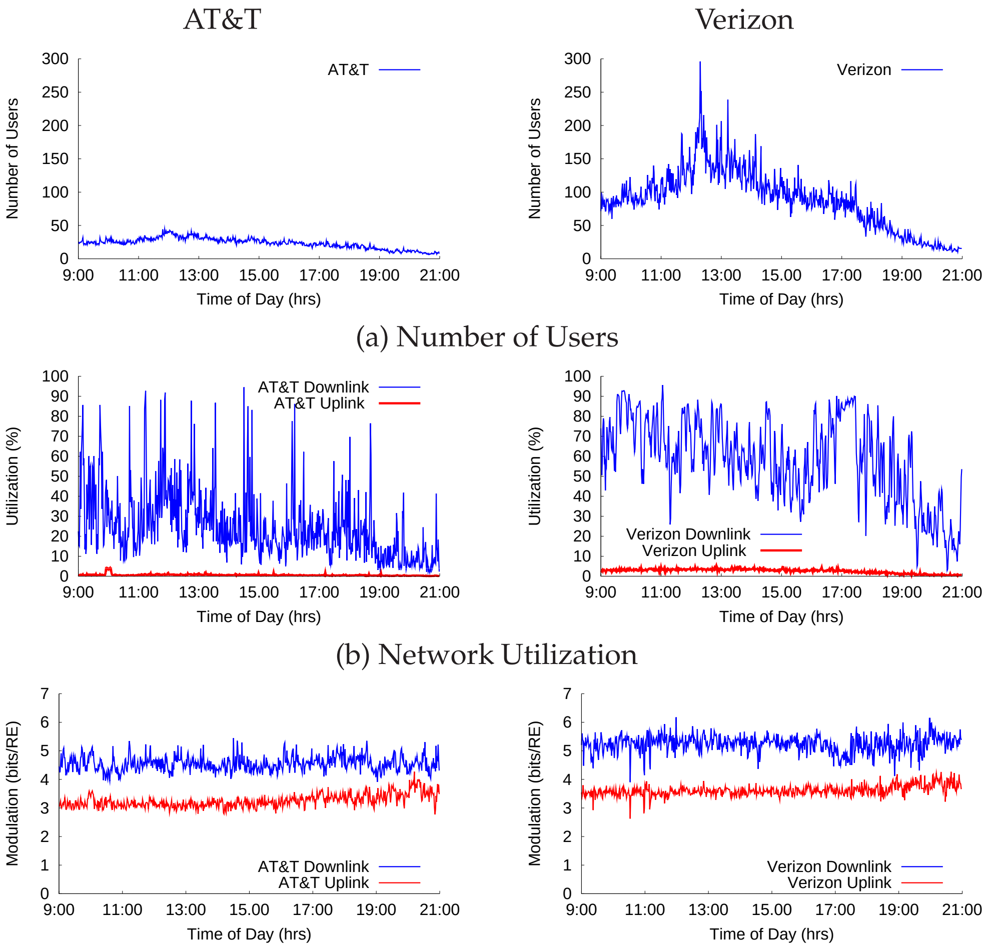
- We compare temporal analytics of two LTE providers and highlight their PHY-layer inefficiencies in §6.6.1 and §6.6.2.
- We provide insights on the LTE rate adaptation algorithm in §6.6.3.
- We apply LTEye's spatial analytics to two applications: detecting cheaters in an exam hall in §6.6.4 and visualizing a spatial heatmap of LTE performance in §6.6.5.
- We perform micro-benchmarks to evaluate the accuracy of LTEye's localization and RF fingerprints in §6.6.6 and §6.6.7.

6.6.1 Comparing Temporal Analytics of Providers

End-users can deploy LTEye sniffers to compare the providers in their locality in terms of usage patterns, quality of service and congestion. In this experiment, we compare aggregate temporal analytics of two providers in our campus: AT&T and Verizon.

Setup. We place LTEye sniffer in four locations in the MIT campus, each listening to the AT&T and Verizon base stations that serve that location. We populate LTEyeDB over the duration of a representative weekday from 9:00am to 9:00pm. To reduce processing overhead, LTEye's logger collects traces for a duration of one second, every three seconds. It validates these traces by only accepting control information with low bit error rate as reported by the convolutional decoder. We then measure the following metrics for each one-second trace: (1) Number of Active Users; (2) Mean Utilization of the Uplink and Downlink; (3) Mean Link Quality measured as the number of bits per resource element (bits/RE) in the uplink and downlink. We average each of these quantities over one minute intervals and plot them over time of day at a representative location (Fig. 9-14). We also estimate the mean value of these metrics across locations over one week to infer aggregate trends (Table 6-3).

Number of Active Users. Fig. 9-14(a) measures the number of active mobile users in a



(a) Number of Users

(b) Network Utilization

(c) Link Quality

Figure 6-7: Temporal Analytics. The plots show the following metrics for AT&T (above) and Verizon (below), measured every minute over a typical working day from a representative base station: (a) Number of Active Users per second; (b) Percentage of Utilized Resource Elements on the Uplink (red) and Downlink (blue); (c) Mean Number of Bits per Resource Element on the Uplink (red) and Downlink (blue).

representative AT&T and Verizon cell over different times of the day. We observe that for both providers, the number of users in the morning increases steadily, and peaks at around 12:00 pm, after which the number begins to decrease. The increase in activity at around noon may be attributed to a greater number of subscribers who access LTE outdoors as they leave for lunch. Across locations, we observe that Verizon has a greater number of active users on average at 87.7, while AT&T has 23.4 active users through the day (see Table 6-3).

Network Utilization. Fig. 9-14(b) plots the utilization of a representative AT&T and Verizon cell, over different times of the day. Specifically, we measure the percentage of resource

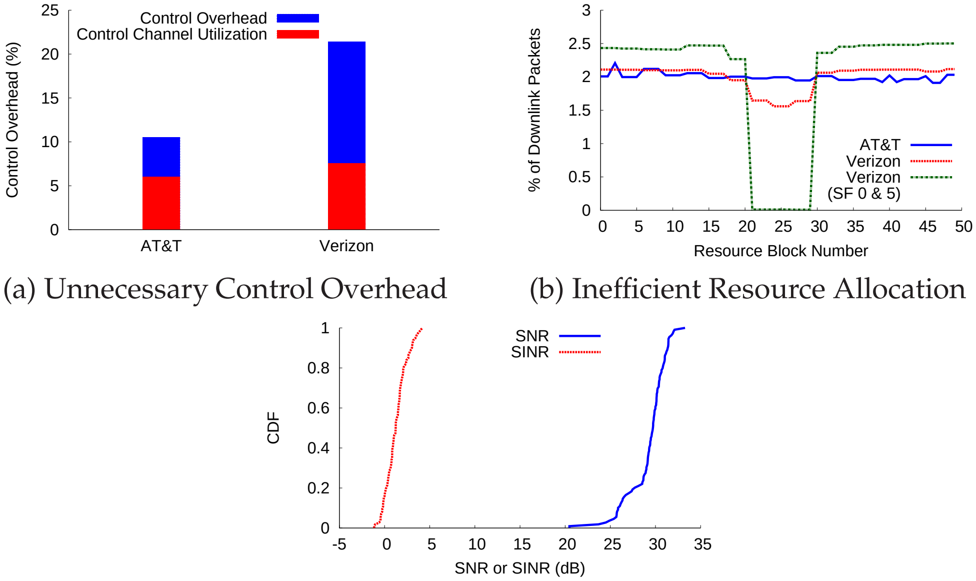
Metric	AT&T		Verizon	
	Mean	Std-dev	Mean	Std-dev
Number of Users	23.37	7.90	87.66	44.75
% Downlink Utilization	25.20	17.35	58.18	20.58
% Uplink Utilization	0.59	0.47	2.60	1.06
Downlink MCS (bits/RE)	4.56	0.26	5.23	0.30
Uplink MCS (bits/RE)	3.25	0.22	3.61	0.18

Table 6-3: Aggregate Statistics. Tabulates mean and standard deviation of statistics over four locations for AT&T and Verizon.

elements used by uplink and downlink traffic. Two trends emerge: First, both providers often achieve high downlink utilization (over 80%) through the day. AT&T achieves such high utilization sporadically through the day (for 2% of the day), while Verizon is heavily utilized for a more significant fraction of the day (for 18% of the day). Second, the utilization of the uplink is significantly lower than the utilization of the downlink, both for AT&T and Verizon.

Specifically, the mean downlink utilization (25.2% - AT&T, 58.2% - Verizon), far exceeds uplink utilization (0.6% - AT&T, 2.6% - Verizon) even when averaged across locations. While it is expected that the downlink is higher in demand, our results reveal that the LTE uplink is an unprecedented 20 to 40 times less utilized than the downlink. This exposes the practical limits of Frequency Division Duplexing mode of the LTE standard used by both AT&T and Verizon, which provides independent equally sized uplink and downlink frequency bands. Hence, our results can help policy makers encourage operators to adopt revised LTE Advanced standards that permit unequal allocation of resources to the uplink and downlink (e.g., via asymmetric carrier aggregation[83]) without relying on data from providers themselves to make the case.

Link Quality. Fig. 9-14(c) measures the average quality of channels in the network for a representative AT&T And Verizon cell measured over different times of the day. In particular, we measure the mean number of bits transmitted per resource element (bits/RE), capturing the modulation and coding scheme (MCS) on the uplink and downlink. Our results show that the mean quality on the downlink (5.2 - Verizon, 4.6 - AT&T) exceeds that of the uplink (3.6 - Verizon, 3.3 - AT&T). As mentioned earlier, this is because users are limited in transmit power and number of MIMO antennas, when compared to the base station. Further, the mean link quality of Verizon is marginally higher when compared to AT&T.



(a) Unnecessary Control Overhead

(b) Inefficient Resource Allocation

(c) Inter-Cell Interference

Figure 6-8: PHY-Layer Inefficiencies. (a) Measures the utilization of the control channel for AT&T and Verizon, across different number of symbols allotted to the control channel. (b) Plots the percentage of data allotted to different downlink resource blocks for AT&T and Verizon. (c) CDF of measured Signal-to-Noise Ratio (SNR) and Signal to Interference Plus Noise Ratio (SINR) at spots of high inter-cell interference.

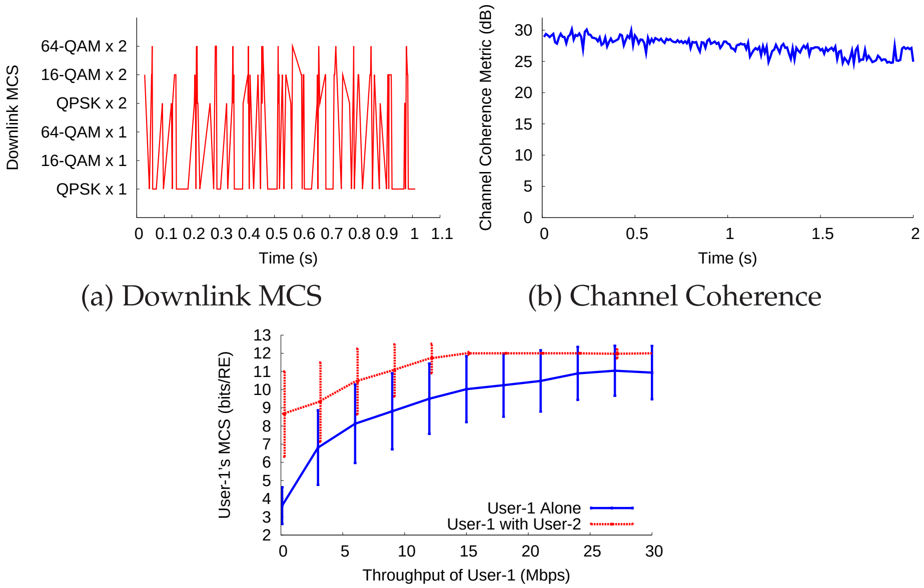
■ 6.6.2 Identifying PHY-Layer Problems and Inefficiency

Cellular Providers and independent researchers can use LTEye to diagnose problems and inefficiencies at the LTE-PHY layer. In this experiment, we identify such deficiencies by analyzing the LTEyeDB database populated for both AT&T and Verizon. In particular, we consider the traces gathered over four locations in our campus served by different base stations over one week, as explained in §6.6.1. Interestingly, many of these PHY-layer inefficiencies may be unknown even to the operators as they are part of the PHY-layer implementations adopted by the base station vendors.[8]

Unnecessary Control Overhead. As explained in §7.1, the LTE resource grid on the downlink is divided into three main PHY-layer channels: the broadcast channel, the control channel and the data channel. The control channel occupies the first 1-3 symbols of each LTE sub-frame resulting in a control overhead ranging from 7% to 21% of all downlink resources. Ideally, an LTE base station should adapt the number of control symbols used in each sub-frame depending on the amount of control traffic that is required to be sent.

In practice, we discovered that the control overhead of AT&T base-stations was 10%,

[8] We confirmed this privately with some base station vendors.



(a) Downlink MCS

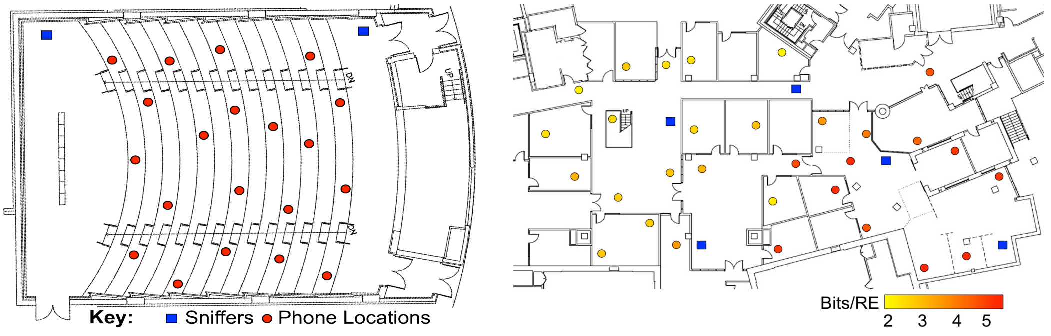
(b) Channel Coherence

(c) MCS vs. Demand

Figure 6-9: Insights into Rate Adaptation. (a) Trace of Downlink MCS for a user over time; (b) Channel Coherence metric of a USRP placed at the user's location over time; (c) Mean downlink MCS (bits/resource element) across demand, with and without another high-demand user.

while that of Verizon base-stations was 21%. This is because, unlike AT&T, Verizon always uses three control symbols in each sub-frame, regardless of the amount of control traffic they contain. One might wonder if this is because Verizon's control channels are significantly more utilized, warranting the additional overhead. Fig. 6-8(a) plots the control overhead of both AT&T and Verizon, as well as how much of this overhead is utilized. Clearly, the overhead of Verizon is significantly larger, despite having only a marginally higher control traffic (7.5%) than AT&T (6.1%). As a result, we estimate that Verizon can gain as much as 10% of additional downlink resources for data, just by adapting its control overhead to control-traffic demand.

Inefficient Resource Allocation. In this experiment, we analyze the downlink resource allocation mechanism of LTE base-stations during periods of high network utilization (over 80%). Fig. 6-8(b) plots the percentage of downlink data transmitted in each resource block across users, measured for both AT&T and Verizon. For AT&T, the graph remains flat across resource blocks, indicating that on average, a user is equally likely to get any of the downlink resource blocks. For Verizon however, we observe a peculiar dip around resource blocks 22-27. To investigate this, we noticed that Verizon avoids these resource blocks completely on subframes 0 and 5. This is because a few symbols in these resource blocks are dedicated for the broadcast channel. As a result, Verizon avoids allocating these resource blocks completely, leading the remaining symbols in these resource blocks to lie completely unused, even during peak hours of demand.



(a) Classroom Testbed

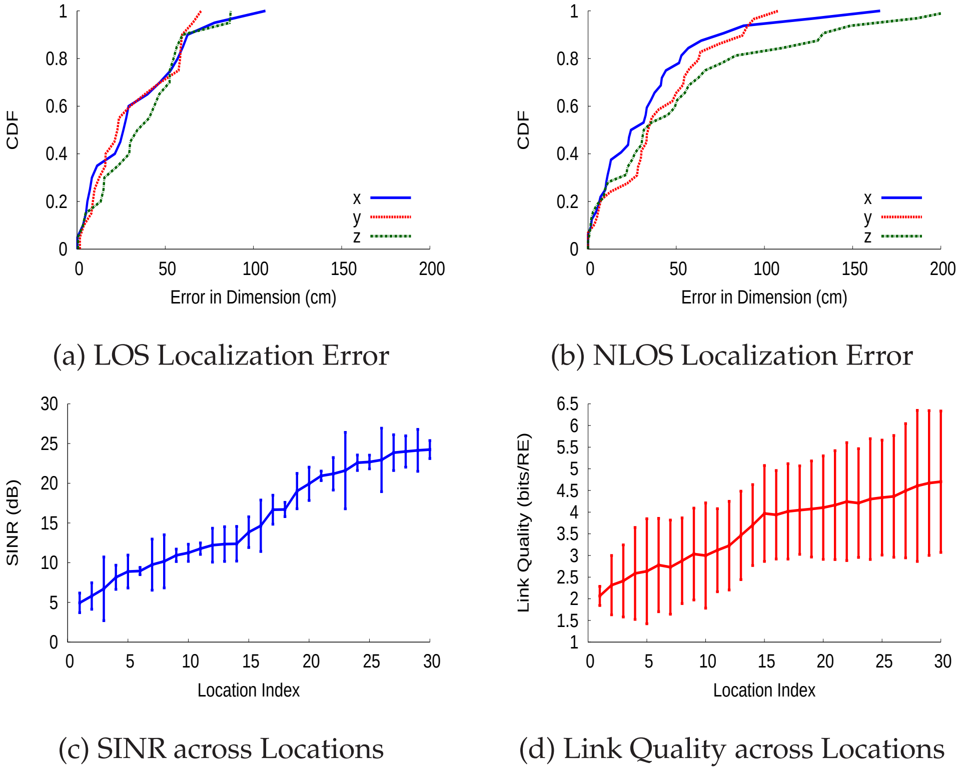
(b) NLOS Testbed & Heatmap

Figure 6-10: Spatial Analytics Testbed. (a),(b): Depicts our two testbeds a large exam hall and a floor of a large building. LTEye sniffers are denoted as blue squares and candidate phone locations as red circles. Further, the red circles in (b) are colored in shades of red, based on observed link quality from the base station;

Inter-Cell Interference. One of the key benefits of LTEye is to provide insight into why users obtain poor performance. During the course of our experiments, LTEye localized users at certain spots that achieved poor link quality on the downlink (the lowest QPSK rate), but high quality on the uplink. To investigate this, we moved our LTEye sniffers and testbed mobile phones to these locations. Surprisingly, our phones at these spots reported very high RSSI[9] from the base station, yet often switched to 3G. We then used our LTEye sniffers at these spots to measure the signal-to-noise ratio (SNR) as well as the signal-to-interference plus noise ratio (SINR) on the downlink. Fig. 6-8(c) reports the CDF of these quantities across these locations. The figure demonstrates that these locations suffer from significant interference, with a mean SINR of 1.3 dB and a minimum of -1.2 dB, despite a high mean SNR of 29 dB. We realized the source of about 27 dB of interference is from neighboring Verizon base stations sharing same downlink spectrum. Specifically, our sniffer could sense as many as five distinct base station cell IDs at a single location. This is problematic as it affects pilot reference signals (known as cell-specific reference signals) that are critical for channel estimation. Base stations transmit these pilots in one of

[9] The phone reported a receive signal strength (RSSI) of around -85 to -95 dBm, where the noise floor is at -120 dBm.

three different subsets of resource elements depending on their cell ID.[10] Given that five base stations are observed at a given location, some of these pilots will inevitably collide, significantly impacting the decodability of signals from those base-stations. Hence, these observations emphasize the need for effective placement and power control of base stations and small cells. They further highlight the importance of careful assignment of cell IDs to neighboring cells, to avoid interference between their pilots.



(a) LOS Localization Error

(b) NLOS Localization Error

(c) SINR across Locations

(d) Link Quality across Locations

Figure 6-11: Spatial Analytics. (a),(b): CDF of the error in 3D localization on each dimension in line-of-sight and non-line-of-sight scenarios respectively; (c),(d): Plots the measured SINR using USRPs and observed link quality from the downlink control channel across phone locations.

■ 6.6.3 Insights into LTE Rate Adaptation

While the LTE standard describes much of the PHY-layer protocol and procedures, the rate adaptation algorithm is still left to the choice of individual operators. In this section, we show how LTEye can shed light on some interesting aspects of this algorithm for an AT&T base-station in our locality.

[10] For 2-antenna base stations, reference signals on both antennas occupy one of three subsets of REs, based on cell ID modulo 3 [6].

We consider a single user device that downloads a random bitfile over a 1 Mbps TCP link from a server (we control TCP rate by throttling the bandwidth at the server using tc[3]). We conduct our experiment during periods of low network utilization, where the mobile device in our testbed is the sole active user of the network. The user device is placed at a static high SNR location, which should support the highest modulation and coding on the downlink.

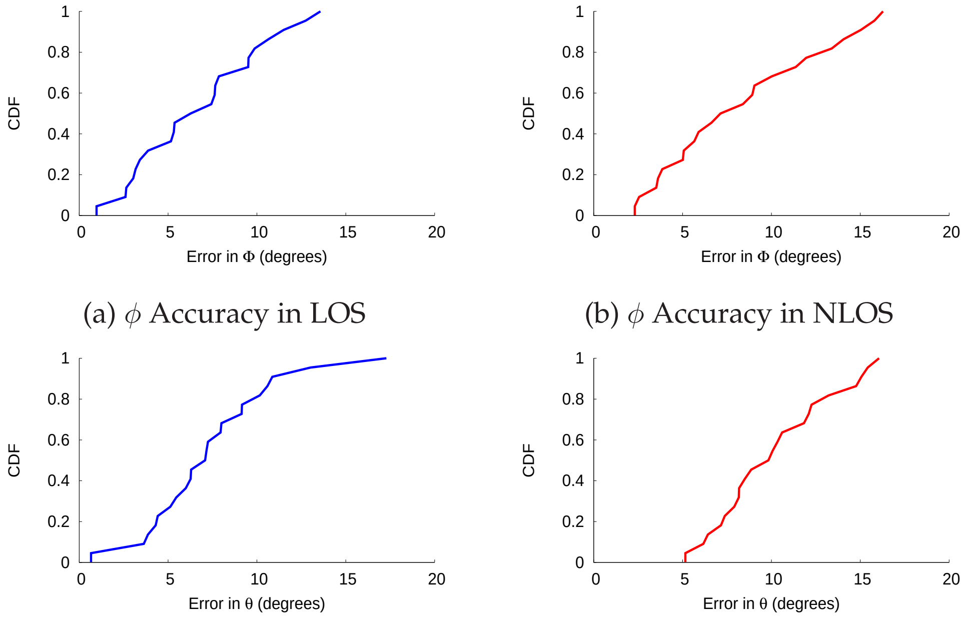
Fig. 6-9(a) plots the user's modulation and coding scheme for one second on the downlink. Surprisingly, the graphs indicate that the rate adaptation algorithm hops over a wide range of modulation, during the experiment. One possible explanation is that the base station is responding to loss of downlink packets. However, the control channel indicated no packet loss on the downlink over the entire experiment. A second explanation is that the wireless channel is not actually coherent over the duration of one second, even though the phone is static. To investigate this, we place a USRP at the user's location and estimate a channel coherence metric[11] capturing the base station's SNR over two seconds, assuming the channel was estimated only once at time 0 (see Fig. 6-9(b)). We observe that the channel is indeed coherent throughout the experiment. Hence, the rate adaptation algorithm is fairly complex, involving aggressive modulation exploration.

Next, we repeat the above experiment for different downlink demands (i.e. TCP throughput) under identical SNR, and plot the mean downlink modulation (in bits per resource element) as shown by the blue line in Fig. 6-9(c). Interestingly, as the downlink demand increases, the observed modulation also increases as well. In other words, the base station avoids transmitting packets to the user at high modulation (i.e. avoids risking higher loss probability), unless it is forced to, since the user demands high throughput.

Finally, we repeat the above experiment, this time adding a second user device (User-2) to the network, while first user (User-1) downloads a file at different TCP throughputs, as before. We allow User-2 to download a large bitfile containing random strings via six simultaneous TCP connections so as to saturate the LTE downlink demand. Interestingly, we now observe that packets to User-1 are sent at higher modulation, across demands (see the red line in Fig. 6-9(c)). To understand why this is the case, note that by sending data to User-1 at higher modulation, the base station consumes less downlink resources per bit for

[11] Given an initial channel $h(0)$ and current CFO-adjusted channel $h(t)$, channel coherence metric is $10\log_{10}|h(t)|^2/|h(t)-h(0)|^2$

User-1's data. This relieves more network resources that the base station can now assign to User-2 to serve its high demand. Therefore, base stations transmit packets at conservative modulation, only when this does not impact overall network throughput.



(a) ϕ Accuracy in LOS

(b) ϕ Accuracy in NLOS

(c) θ Accuracy in LOS

(d) θ Accuracy in NLOS

Figure 6-12: Accuracy of θ and ϕ. Plots CDF of error in measured spatial angles in line-of-sight (LOS) and non-line-of-sight (NLOS).

■ 6.6.4 Detecting Cheaters in a Large Exam Hall

LTEye can enable new applications customized to the need of a particular community of users. For example, many modern exams follow an open book/material policy and allow students access to computers during the exams. However, students are asked to abstain from using the Internet to chat and collude online. Enforcing this policy over Wi-Fi is relatively easy by monitoring the Wi-Fi channels, turning the access point off, or even jamming the signal. However cheaters can still use their cellular service to chat with an accomplice. In this experiment, we demonstrate how LTEye's spatial analytics can help localize such cheaters in a large exam hall that accommodates up to 300 students.

Setup. We consider a large 24m × 17m exam hall as shown in Fig. 6-10(a) that seats up to 300 students. The exam hall has multiple chairs on a platform that slopes upwards from the podium. We place two LTEye sniffers on ledges close to the walls, as shown by the blue squares in the figure. The sniffers localize the 3-D location of ten LTE cellphones, corresponding to ten active cheaters accessing the Internet, placed among twenty randomly chosen locations (the red circles in the figure).

Results. Fig. 6-11(a) plots a CDF of the error in each dimension of the estimated 3D-location of each cellphone. We observe a mean error in localization of 34 cm along each dimension and 61 cm in 3D displacement between the measured and actual location. Note that our errors are in 3-D space unlike past work [188, 179] and the experiments were performed in a large 24m × 17m area. LTEye identified the cheater's seat among the 300 seats in the room with 95% accuracy. Note that a random guess has an accuracy of just 1/300. Even when LTEye makes an error, it reports a seat adjacent to the cheater, which is sufficient to visually identify the cheater in most proctored situations. Note that the cellphones are predominantly in line-of-sight to the LTEye sniffers, due to the nature of the exam hall. In §6.6.5, we estimate the error in localization for non-line-of-sight scenarios as well.

■ 6.6.5 Visualizing LTE Performance over Space

As end-users, we have limited visibility into how LTE performance varies in different parts of our home or work place. In this experiment, we address this issue by synergizing LTEye's temporal and spatial analytics to visualize the performance of a cellular provider across spatial locations.

Setup. We deploy five LTEye sniffers on a 60m×34m floor of a large building, denoted by the blue squares in Fig. 6-10(b). We place ten phones in each of thirty randomly chosen locations shown as red circles in the figure. Note that several phones are in non-line-of-sight relative to all LTEye sniffers. We emphasize that for each client device, 3D-localization is performed using the spatial angles from at most three LTEye sniffers. The localization error can be further improved by incorporating spatial angles from additional LTEye sniffers.

Results. Fig. 6-11(b) plots the CDF of localization error along each of the three dimensions for phones that are in non-line-of-sight relative to all LTEye sniffers. Our results show a mean error in localization of 43.7 cm along each dimension and 84.6 cm in net 3-D displacement. Our algorithm to identify the direct line-of-sight path from §6.4.2 is crucial to localize phones in non-line-of-sight. Of course, the algorithm hinges on the fact that the line of sight path is, at the very least, observable in the multipath profile produced by SAR (See §7.3), even if it is not the most dominant path. Our experiments revealed that the line-of-sight path was always observed in the multipath profile of every phone in our large indoor testbed, including those furthest away from each LTEye sniffer in Fig. 6-10(b). Of course, while this may not generalize to every environment, our observations show the benefits of better penetration of signals in the 700 MHz frequency range through walls and obstacles, compared to Wi-Fi signals at 2.4 GHz or 5 GHz, and the higher transmit power of LTE devices in general.

Fig. 6-11(c) measures the mean and variance of Signal to Interference plus Noise Ratio (SINR) observed by a USRP placed in each of the thirty phone locations. The locations are sorted by mean SINR for ease of visualization. Fig. 6-11(d) plots the mean and variance of link quality for each phone (as bits per resource element) at the same locations, measured from LTEyeDB based on the downlink control channels. We observe that the link quality and SNR follow similar trends, showing that LTEye can effectively characterize the performance of the LTE network across spatial locations.

Fig. 6-10(b) visualizes the spatial distribution of link quality across phone locations, denoting positions of high quality as circles with darker shades of red, and low quality with lighter shades of red. The figure indicates that the link quality is strongest at locations to the bottom right, and weakest along locations to the top left. In fact, we found that placing an LTE relay at the top-left part of the floor significantly improves LTE service across the floor.

■ 6.6.6 Measuring Accuracy of Observed Spatial Angles

In this experiment, we measure the accuracy of the polar angle θ and azimuthal angle ϕ, that are key primitives to LTEye's localization algorithm in §6.4, across spatial locations.

Setup. We consider an LTEye sniffer placed in one of five possible locations (denoted by blue squares) in a floor of a large building, as in Fig. 6-10(b). The LTEye sniffers are elevated on ledges close to the walls.[12] We place ten phones in several randomly chosen locations in both line-of-sight and non-line-of-sight, spanning the full range of spatial angles. We find the ground truth of the spatial angles by noting the actual 3D positions of the phones

[12] LTEye cannot tell apart up from down as the rotating antenna path is symmetric. Placing sniffers on ledges removes this ambiguity.

and the sniffer on a scaled high-resolution building floorplan.

Results. Fig.6-12(a)-(d) plot the CDF of error in θ and ϕ in line-of-sight (LOS) and non-line of sight (NLOS) locations. The figures show a low median error in both ϕ (LOS: $6.9°$, NLOS: $7.8°$) and θ (LOS: $7.2°$, NLOS: $9.9°$) across locations. Note that the accuracy can be further improved with multiple LTEye sniffers, particularly, in cases where the deviation in angles is large. Note that our algorithm to find the path of minimum delay was crucial for the accuracy of spatial angles in non-line-of-sight.

■ 6.6.7 Tracking C-RNTIs using RF Fingerprints

In this section, we evaluate LTEye's RF fingerprinting to map C-RNTIs assigned to the same phone. We consider the setup in §6.6.6 above and populate LTEyeDB across several experiments spanning ten minutes each. We track the correct C-RNTI mapping by constantly listening on the uplink from multiple USRPs placed close to each phone to recognize the high power signals that are sent during connection establishment. We also measure the inferred C-RNTI mapping from RF-fingerprints (See §6.3 and §6.4.3).

Results. We measure two quantities: (1) Precision: The percentage of new C-RNTIs which were correctly mapped to old C-RNTIs. (2) Recall: The percentage of correctly retrieved C-RNTIs-mappings among all actual C-RNTI mappings. Our algorithm achieves a high mean precision of $98.4 \pm 1.3\%$ and mean recall of $96.7 \pm 1.4\%$, demonstrating the effectiveness of LTEye's C-RNTI matching algorithm. Note that we leveraged RF fingerprints to track C-RNTIs of users in §6.6.4 and §6.6.5 above.

■ 6.7 Related Work

(a) LTE Sniffing Equipment: Devices such as Wavejudge, ThinkRF and IntelliJudge [145, 169] are wireless protocol sniffers to capture RF signals. They are mainly tools for wireless development and interoperability testing that provide visibility into the interaction between the PHY and protocol layers. Unlike LTEye, these devices need inputs from the cellular provider and do not perform localization or provide spatial analytics.

(b) Open LTE Implementations: There have been efforts in developing open source implementations of LTE protocols, notably OpenAirInterface [42], and OSLD [51]. These initiatives enable running LTE base stations on software radios; they do not extract spatial or temporal analytics.

(c) LTE Measurement Studies: Many recent LTE studies have been conducted using traces collected on participating smartphones or from inside LTE networks. Findings from these studies include: (1) The available bandwidth of LTE networks is highly variable and TCP is not able to fully utilize the bandwidth [77]; (2) LTE is significantly less power efficient than Wi-Fi [76]; (3) LTE latency is more consistent (less variable) than Wi-Fi [157]. Such studies focus on the higher layers of the stack, e.g., TCP throughput, transfer delay, and power usage. In contrast, LTEye focuses on the LTE radio layer; it provides fine-grained temporal and spatial information and does not require traces from the provider.

(d) Cellular Location-Specific RF Measurements: Cellular operators need location-specific RF measurements to troubleshoot performance problems and plan future deployments. They typically obtain coarse location information by mapping a user to her serving cell. Operators then rely on drive tests to refine the spatial measurements. Drive tests are costly and constitute a big part of the network operating expenditure [78]. Further, they are increasingly inadequate as operators move toward femto cells, and need indoor coverage data. To reduce the cost and improve the spatial measurements, recent LTE releases propose mechanisms known as MDT [5]. MDT techniques localize a mobile phone either using in-network time measurement or by collecting location information using the phone's GPS. It is well-known however that in-network localization in cellular networks is not accurate (at hundreds of meters [109]) as time-delay measurements are only available for the serving cell of a mobile user. Even the E911 service using positioning reference signals can only guarantee 150m accuracy 95% of the time [94]. GPS measurements cannot be invoked often as they drain the user's battery. They also cannot capture indoor location.

(e) RF-based Localization Techniques: Our work is related to past work on RF-based localization. This problem has received much recent interest resulting in highly accurate systems. ArrayTrack [188] and PinPoint [82] are an antenna-array based indoor location systems that tracks wireless clients at fine granularity. Chen et. al.[74] build antenna arrays using software radios synchronized with a reference signal. PinIt [179] is an RFID localization system that combines SAR with a deployment of reference RFIDs to achieve highly accurate localization.

Our design builds on this past work but differs in that it introduces two innovative

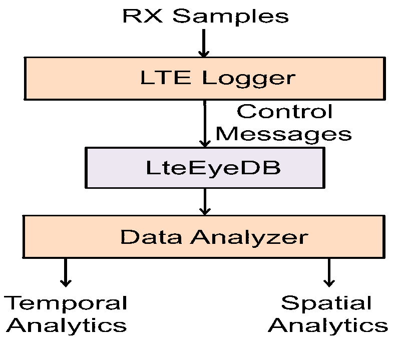


Figure 6-13: LTEye's Architecture: Contains the LTE Logger and Data Analyzer.

localization techniques. First, we extend SAR to operate over communication signals exchanged between a transmitter and a receiver. This contrasts with the current approach for SAR, which is limited to backscatter and radar signals, where the transmitter and receiver are a single node with no Carrier Frequency Offset (CFO) or Sampling Frequency offset (SFO). Second, we introduce a new technique that when combined with SAR or standard antenna arrays, estimates the delay difference between the various paths traversed by the signal to identify the shortest path. In particular, we measure these delays in time based on phase offsets in the frequency domain. Hence, LTEye can resolve differences in delay below one time sample, unlike past work that estimates these delays via correlation in time [82].

■ 6.8 Discussion

We presented LTEye, the first open platform to provide fine-grained temporal and spatial analytics on LTE radio performance, without private user information or provider support. LTEye employs a novel extension of synthetic aperture radar to communication signals to accurately localize mobile users, despite the presence of multipath. We empirically evaluate LTEye on software radios and provide deep insights on the LTE PHY and highlight shortcomings such as inter-cell interference and inefficient spectrum utilization.

Chapter 7
Accurate Wi-Fi Indoor Positioning with Zero Startup Cost

Recent years have witnessed a major interest in developing accurate RF-based localization systems that enable users to navigate indoor spaces much like what GPS provides for outdoor environments [142, 107, 188]. Many advances have been made, leading to localization solutions that deliver an accuracy of tens of centimeters [188, 82, 179]. Unfortunately, none of these solutions have actually reached today's users. This is because past work that provides accurate RF-localization falls in one of two categories: 1) It either requires the deployment of new infrastructure (specialized access points [82], antenna arrays [188], acoustic beacons [107], etc.), or, 2) it needs exhaustive fingerprinting of the environment to learn the spatial distribution of signal strength, either in a training phase [168, 104] or via crowdsourcing [142, 28] to achieve meter accuracy at best. Ideally, we would like to achieve tens of centimeters accuracy for RF localization of mobile devices, without any fingerprinting or specialized infrastructure.

In principle, if mobile devices can be outfitted with a large antenna array [188, 82], they could accurately identify the spatial direction of incoming RF signals. Thus, they could localize themselves relative to the locations of neighboring Wi-Fi access points without requiring fingerprinting or modified infrastructure. In fact, public databases of access point locations are already available for a large number of buildings [184]. Of course, it is infeasible to mount a large antenna array on a small low-power handheld device. But what

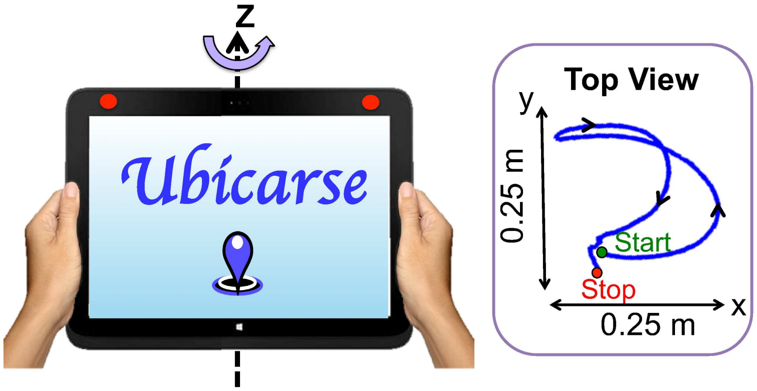


Figure 7-1: Twisting the Mobile Device. The user twists the device about its vertical (z) axis, as shown in the figure on the left. The red circles indicate antenna locations. On the right, we depict the top view of a candidate device trajectory as measured by a Vicon motion capture system.

if today's mobile devices could somehow emulate the localization capability of a large antenna array?

We present Ubicarse an indoor geo-location system that enables mobile devices to indeed emulate a large antenna array. At first, it might seem that one can use Synthetic Aperture Radar (SAR) to mimic an antenna array on a handheld device. Unfortunately, SAR in its existing form is unsuitable for handheld devices. To understand why this is the case, note that a moving antenna performs SAR by collecting signal snapshots as it moves along its trajectory, and jointly processing these snapshots to emulate a large antenna array traced out by that trajectory. Therefore for SAR to mimic a particular antenna array geometry, it must first know the position of the moving antenna at every point along its trajectory. Satisfying this requirement is relatively easy in radar systems where the speed and trajectory of the antenna movement are finely controlled. Unfortunately, when a user moves her handheld device, neither its speed nor its trajectory can be accurately controlled. To get a feel for the accuracy required, in 802.11a/n, even a small error of 2 cm in the relative position of the moving antenna leads to an error of 60 degrees in identifying the direction of the source.

One might consider using motion sensors (accelerometer, gyroscope, compass) present in most mobile devices to estimate antenna positions as it moves. Unfortunately, since SAR requires sub-centimeter accuracy in the device position along its trajectory, commercial motion sensors are virtually unusable to measure such fine-grained translation [127] and can at best be used to measure orientation from their gyroscopes.[1] Requiring the user to rotate the device on a perfect arc with zero translation of the device center would preclude measuring this translation but is impractical to enforce.

Ubicarse addresses this challenge by developing a new formulation of SAR that is translation-resilient. Specifically, we allow the user to twist her mobile device about its vertical axis, as shown in Fig. 7-1. Even if the twisting involves unknown trajectories (that include translation), we can accurately compute SAR knowing only the rotation estimate from the gyroscope. To do so, we exploit the MIMO capability of modern wireless cards. Suppose the mobile device has two antennas. The distance between these antennas is fixed independent of how the user moves her device. Thus, whenever the user translates the device, the relative position vector of both its antennas remains the same. In contrast, as the device rotates, the relative position vector of the antennas, also rotates. Leveraging these observations, we develop a new SAR formulation that operates on *relative* wireless channels to estimate the direction of the access point's signal, purely based on the device's orientation with no information required on its exact position or translation. We analytically demonstrate that Ubicarse's translation-resilience holds under the same assumptions made for standard SAR and antenna arrays, both in single path and in the presence of multipath effects. We also implement the system and demonstrate a localization accuracy of the mobile device within few tens of centimeters, with no specialized infrastructure or fingerprinting.

Ubicarse is not limited to localizing RF devices; it can also localize common objects that have no RF source attached to them. Specifically, we leverage the camera available on most mobile devices to enable a user to learn the location of an object of interest by simply snapping photographs of the object from multiple perspectives. At first, it might seem that this application can be satisfied simply by tagging the object in the photograph with the position of the mobile device, obtained from RF-based localization. Such an approach however would yield poor tagging accuracy, particularly when the user cannot get too close to the object, for e.g., tagging a broken overhead lamp to report it to the facilities department.

To achieve high-precision geotagging, Ubicarse exploits the synergy between its WiFi-

[1] This is because accelerometers report the net acceleration of the device including gravity, which must be subtracted out and the result integrated twice to obtain translation.

based localization capability and stereo vision algorithms used for 3-D reconstruction. Specifically, today's vision algorithms can accurately localize an object with respect to the set of camera coordinates from which the object was imaged [171], but it cannot find the coordinates of the object with respect to a global reference. In contrast, Ubicarse's SAR localization can identify the global positions of the device's camera accurately, but has no information about the object. Therefore, by combining the two methods, we can localize an object to within tens of centimeters in global coordinates – a promising step towards a future where we can tag or search for objects in the physical world using only our mobile devices, much the way we currently search for information on the web.

We implemented Ubicarse on a HP SplitX2 Tablet equipped with Intel 5300 wireless cards and Yei Technology motion sensors. We build on the 802.11 CSI tool [67] to obtain the wireless channels on the tablet. We conducted our experiment in a large university library, with books arranged in multiple shelves and racks that emulate the complex multipath characteristics of large warehouses and departmental stores, where indoor localization has been of particular interest. Our experiments reveal the following:

- **Translation Resilient SAR:** Ubicarse's SAR is resilient to unknown translations. We move the device along complex trajectories including several random walks and curves. We use Ubicarse to localize the direction of access point placed at distances from 2 to 12 m without knowledge of the trajectory of the movement. Our results show a median error in angle of $3.4°$ across trajectories, demonstrating the translation-resilience of Ubicarse's SAR.
- **Device Localization:** We ask a population of users to twist the tablet to localize themselves relative to the access points in the environment. The results show that Ubicarse localizes the tablet with a median error of 39 cm in full three dimensional space. Thus, Ubicarse achieves its goal of accurate localization without requiring either new infrastructure or signal fingerprinting.
- **Object Localization:** We integrate Ubicarse with the VisualSFM computer vision toolkit [186] and evaluate its object geotagging capability. We use Ubicarse to localize books in the library relative to the access points, by taking a few pictures of them. Our results show a median error of 17 cm in full 3-D space, which, surprisingly, is superior to Ubicarse's device localization. Indeed we show that a joint optimization of Ubicarse and vision has an interesting side-effect: we can significantly refine Ubi-

carse's device (i.e. camera) location estimates simply by studying the relative camera locations output by vision algorithms, corresponding to pictures snapped from different perspectives. In doing so, we show that Ubicarse's localization accuracy improves to a median of 15 cm from ground truth.

Contributions Ubicarse is an indoor localization system that achieves tens of centimeters of accuracy on commodity mobile devices, with no specialized infrastructure or fingerprinting. Ubicarse achieves this through a novel formulation of Synthetic Aperture Radar that is translation resilient, making it practical to deploy on handheld devices. Further, Ubicarse uses the cameras on mobile devices to help users geo-tag objects around them to within tens of centimeters. We implement Ubicarse on commodity tablets and experimentally validate its accuracy in complex indoor settings.

■ 7.1 Primer on Synthetic Aperture Radar

Synthetic Aperture Radar has been widely used in aircrafts and satellites to map the topography of the Earth's surface [50]. Recent systems [179, 178] have also leveraged SAR for RFID localization.

SAR's primary goal is to allow a single-antenna wireless receiver to isolate and analyze the multiple signal paths emanating from a wireless transmitter. Specifically, it computes a *multipath profile*, which measures the relative signal power received from the transmitting source along different spatial directions (see Fig. 7-2). To do this, SAR leverages a simple principle: a moving receiver antenna snapshots the wireless signals at different spatial locations. Combined, these channel snapshots mimic a multi-antenna array (Fig. 7-2). Thus, one can simply apply standard antenna array equations [131] on the signals received at each antenna position to compute the multipath profile. For instance, one can apply equations for a linear antenna array [188] if the receive antenna moves on a line, or a circular antenna array [45], if the receiver rotates about a circle.

To understand SAR more formally, lets examine a special case of SAR for a received single path.[2] We focus on emulation of a circular antenna array, a scenario that is particularly relevant for this chapter. Let's suppose the receive antenna rotates about the origin along a circle, taking n snapshots of the wireless channel from a source along a direction

[2] We explicitly deal with the more general multipath case in §7.3.2

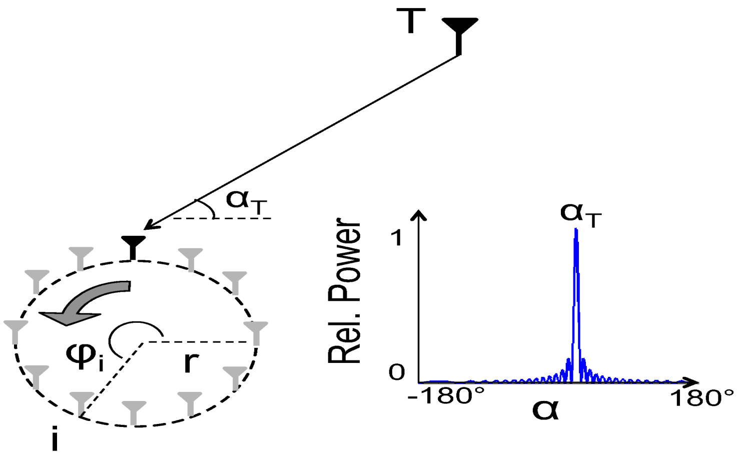


Figure 7-2: Circular SAR. Depicts a single antenna moving in a circle of radius r to emulate a circular antenna array. Gray antennas depict points where channel snapshots were taken. At snapshot i, the antenna is at an azimuthal angle ϕ_i. The multipath profile for a line-of-sight transmitter T has a sharp peak at the direction of the source α_T.

α_T as shown in Fig. 7-2. From basic channel models, we can write the the wireless channel h_i measured by the receiver at the i^{th} snapshot (where $i = 1, \ldots, n$) as the complex number [172]:

$$h_i = \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + rcos(\alpha_T - \phi_i))} \tag{7.1}$$

Where (r, ϕ_i) is the antenna's polar coordinates at snapshot i (Fig. 7-2), d is its distance to the source, and λ the signal wavelength.

At this point, SAR computes the relative signal power along each spatial direction to generate a multipath profile. To do this, it needs two inputs: The measured channel snapshots h_i along the antenna trajectory, and the positions (r, ϕ_i) of the antenna along this trajectory. It then finds the relative power $P(\alpha)$ along direction α as:

$$P(\alpha) = \left| \frac{1}{n} \sum_{i=1}^{n} h_i e^{\frac{+j2\pi}{\lambda} rcos(\alpha - \phi_i)} \right|^2 \tag{7.2}$$

Notice that the above multipath power profile $P(\alpha)$ is maximum precisely if $\alpha = \alpha_T$, i.e. along the true direction of the source. This is because the terms $h_i e^{\frac{+j2\pi}{\lambda} rcos(\alpha - \phi_i)}$ are identical in phase, and therefore add up constructively when $\alpha = \alpha_T$ (see Eqn. 7.1), and tend to add up destructively as α deviates from α_T. Therefore, the multipath profile can be used to accurately ascertain the physical direction of a transmitter, a property crucial for indoor localization.

SAR makes an important assumption: It assumes accurate knowledge of the position of the device at different points in time relative to some fixed origin in space (e.g. its initial location). Clearly, from Eqn. (7.2), in order to compute $P(\alpha)$ one must know the polar coordinates (r, ϕ_i) of the antenna positions. To get a feel for the accuracy in position required, in 802.11a/n, even a small error of 2 cm in the relative position of the moving antenna leads to an error of about 3 m in localization, when the transmitting source is 6 m away. The assumption of accurately known antenna trajectories is valid in radar systems where the speed and trajectory of the movement are finely controlled. In the following sections, we describe how this assumption poses an important challenge to our system, where users cannot be expected to move their devices along accurately known trajectories.

■ 7.2 Overview of Ubicarse

Ubicarse enables accurate indoor localization of commercially available mobile devices with no prior infrastructure or fingerprinting. At the heart of Ubicarse's device localization is the ability to perform SAR using today's off-the-shelf Wi-Fi cards and motion sensors (i.e. chips containing accelerometers, magnetometers and gyroscopes). In its traditional form, SAR assumes that channel models have a clear dependence on both the position and orientation of the antennas (see Eqn. (7.1) and Fig. 7-2). Therefore computation of an accurate multipath profile requires precise knowledge of how the antennas are located in space. However, users of Ubicarse cannot be expected to move their devices along precise trajectories. Consequently, Ubicarse must rely on the device's motion sensors to infer its position and orientation over time. Unfortunately, applying SAR using commodity motion sensors is challenging for the following reasons: (1) They are virtually unusable to measure device translation at fine granularity [127], as necessitated by SAR; (2) While the orientation they measure is accurate for short time intervals, the accuracy reduces over time due to integration drift [56].

In §7.3 and §7.4, we describe how Ubicarse resolves the above challenges to accurately perform SAR even if a user twists her device along unknown trajectories to obtain multipath profiles. In §7.5, we explain how Ubicarse uses these multipath profiles to accurately localize the mobile device. In §7.6, we show how Ubicarse integrates with vision algorithms to accurately geotag objects of interest in the environment, enabling a new class of mobile applications.

■ 7.3 Translation-Resilient SAR

In this section, we describe how Ubicarse performs SAR even if the user moves her device along complex unknown trajectories. Specifically, we address the challenge that commodity motion sensors only provide accurate orientation information and do not provide accurate translation data that is crucial to perform SAR (see §7.1). Therefore, Ubicarse must perform SAR in a manner resilient to any device translation the user causes as she twists her device, yet correctly obtains the multipath signal profile.

To achieve this goal, Ubicarse leverages the MIMO capability of modern wireless cards. In particular, we consider mobile devices with just two MIMO antennas that measure the wireless channels from any single antenna of a wireless access point. As the user twists her device, the two antennas sample the wireless channel at each point in the device trajectory. The channel snapshots emulate a virtual array of antennas, where the number of virtual antennas depend on the number of channel samples. The main challenge however, is to perform SAR using the channel samples gathered from the two moving physical MIMO antennas, using only the orientation of the device and without its translation information.

Ubicarse's key idea to achieve this is as follows: Instead of performing SAR by plugging in the wireless channels measured at the antennas, Ubicarse feeds into SAR a quantity we call the *relative wireless channel* between the two physical antennas. Unlike channels on individual antennas that depend on the absolute antenna positions, the relative channel only depends on the relative position vector between the two antennas. Recall that the distance between the two antennas on a device is fixed, regardless of how the user moves her device. Thus, whenever the user translates her device, the relative position vector of both its antennas remains the same. In contrast, as the device rotates, the relative position vector of the antennas, also rotates. Hence, by plugging in relative channels into the SAR formulation, Ubicarse obtains multipath profiles with no information on device center translation.

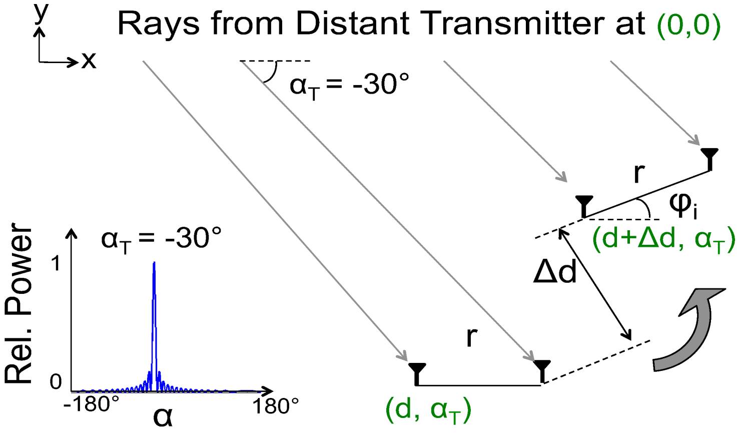


Figure 7-3: Ubicarse's SAR. Shows polar coordinates (in green) of the device at an offset Δx_i and Δy_i along the x and y-axis and orientation ϕ_i. Note that $\Delta d = \Delta y_i / \sin \alpha_T$. The origin of the system $(0, 0)$ is at the transmitter.

7.3.1 Description of Translation Resilient SAR

We explain how Ubicarse satisfies two properties: 1) It is translation-resilient; 2) It correctly estimates SAR profiles.

Translation-Independence. We begin by presenting our solution for line-of-sight scenarios (i.e. free-space scenarios), where the signal from the transmitter arrives along one dominant path. For simplicity, we assume that both the transmitter and our device lie on a two-dimensional plane (extension to 3D is described in §7.3.3). We consider a transmitting source T placed at the origin and a two-antenna receiver device, as shown in Fig. 7-3. As the user twists the receiving device, it takes n different snapshots at the wireless signals, each at different orientations and positions. The two antennas of the device are separated by a distance of r (typically, 6 to 20 cm).[3] Denote the initial polar coordinates of the first receive antenna as (d, α_T) relative to the transmitter where d is the distance between the transmitter and receiver. Note here that the main goal of SAR is to estimate α_T which captures the direction of the transmitting source relative to the device.

As the user moves the device, its antenna coordinates at any snapshot i (where $i = 1, \dots, n$) depend on two quantities: 1) The translation of the device along the x and y axes at the snapshot relative to its initial location denoted by Δx_i and Δy_i. 2) The orientation of the device ϕ_i, relative to the x-axis is shown in Fig. 7-3. Hence, our goal is to perform SAR

[3] Ubicarse assumes that the distance between the two antennas on the device is known. This is typically available from the manufacturer's specification for a given device.

purely based on orientation ϕ_i, with no information on translation along the x and y axes: Δx_i, Δy_i.

Though our goal is to perform SAR without knowledge of the translation (Δx_i and Δy_i), clearly if the device is translated by a large distance with respect to the source, the angle of arrival itself (the quantity that we are interested in estimating) will change. Therefore in our system we assume that during SAR computation, the source is at a far distance compared to the movement of the tablet (a standard assumption for SAR and antenna arrays in general [188, 50]). The form of translation independence we seek in the current SAR formulation is robustness to local translations of the device center as the person twists the tablet as shown in Fig. 7-1. To achieve this goal, we first need to characterize the wireless channel of the source's signal measured at the antennas in each snapshot taken by the receiving device. We can apply simple geometry to write the polar coordinates of the two antennas as: $(d + \frac{\Delta y_i}{\sin \alpha_T}, \alpha_T)$ and $(d + \frac{\Delta y_i}{\sin \alpha_T} + r\cos(\alpha_T - \phi_i), \alpha_T)$ (as shown in Fig. 7-3) where the polar angle α_T is the same for the two antennas under our far distance assumption. From basic wireless channel models [172], we can write the wireless channels $h_{1,i}$ and $h_{2,i}$ measured at the receive antenna 1 and 2 during its i^{th} snapshot as:

$$h_{1,i} = \frac{1}{d + \frac{\Delta y_i}{\sin \alpha_T}} e^{\frac{-j2\pi}{\lambda}(d + \frac{\Delta y_i}{\sin \alpha_T})} \approx \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + \frac{\Delta y_i}{\sin \alpha_T})} \tag{7.3}$$

$$h_{2,i} = \frac{1}{d + \frac{\Delta y_i}{\sin \alpha_T} + r\cos(\alpha_T - \phi_i)} e^{\frac{-j2\pi}{\lambda}(d + \frac{\Delta y_i}{\sin \alpha_T} + r\cos(\alpha_T - \phi_i))} \tag{7.4}$$

$$\approx \frac{1}{d} e^{\frac{-j2\pi}{\lambda}(d + \frac{\Delta y_i}{\sin \alpha_T} + r\cos(\alpha_T - \phi_i))} \tag{7.5}$$

Where λ denotes the wavelength of the wireless signal. Note that the approximations hold assuming $\Delta x_i, \Delta y_i \ll d$, i.e. the source is far relative to the movement of the device, as in standard SAR [50].

We now define the relative wireless channel at the i^{th} snapshot $\hat{h}_i = h_{2,i} h_{1,i}^*$, where $(.)^*$ denotes the complex conjugate. Intuitively, the relative wireless channel captures the relative phase between channels of the two antennas. Mathematically, the relative channel is given by:

$$\hat{h}_i = h_{2,i}h_{1,i}^* = \frac{1}{d^2}e^{\frac{-j2\pi}{\lambda}(d+\frac{\Delta y_i}{\sin\alpha_T}+r\cos(\alpha_T-\phi_i))}e^{\frac{+j2\pi}{\lambda}(d+\frac{\Delta y_i}{\sin\alpha_T})} \tag{7.6}$$

$$= \frac{1}{d^2}e^{\frac{-j2\pi}{\lambda}r\cos(\alpha_T-\phi_i)} \tag{7.7}$$

Interestingly, note that the relative channel is independent of device translation (Δx_i and Δy_i) and is dependent only on the orientation ϕ_i. In fact, we observe that the relative wireless channel $\hat{h}_i$ is identical in form to the wireless channel of a single antenna rotating about a circle of radius r as discussed in Eqn. 7.1 of §7.1 (barring a constant factor $\frac{1}{d}e^{\frac{j2\pi d}{\lambda}}$). Intuitively, this stems from the fact that relative to the frame of reference of the first antenna, the second antenna simply appears to move around a circle.

As a result, we can simply "plug-in" the relative channels into standard circular SAR equations (Eqn. 7.2 from §7.1) to derive the multi-path profile of the transmitter's signal as follows:

$$P(\alpha) = \left|\frac{1}{n}\sum_{i=1}^{n}\hat{h}_i e^{\frac{+j2\pi}{\lambda}r\cos(\alpha-\phi_i)}\right|^2 \tag{7.8}$$

Hence, even if the user twists the device in arbitrary ways, causing both rotation and translation to the device, its two antennas never translate relative to each other, and can therefore emulate a circular antenna array.

We make the following important observations on computing the multipath profile as in Eqn. 7.8 above:

- The complexity of our algorithm is $O(n * \log(m))$ where n is the number of channel snapshots, and m is the desired resolution in angle of arrival. The complexity is logarithmic in m since one can first perform SAR at a low resolution, and repeat SAR iteratively at higher resolutions only at angles of highest power.
- Note that the wireless channels used in the above equations must capture both magnitude and the phase of the received signal. Our implementation therefore uses the channel state information (CSI) measured by a wireless card that are complex numbers, unlike received signal strength (RSS).
- The above solution can be readily extended to OFDM systems (like 802.11n), by averaging the power-profile computed for each OFDM sub-carrier.

- Our system only requires wireless channels measured from a single antenna at the 802.11 access point and two antennas at the client. If channels from more than one antenna on a Wi-Fi access point are available, the power profiles can be measured from each individual antenna, and the results averaged.

Correctness of Multipath Profile. One might wonder if the multipath profiles generated by the above technique are indeed correct. Specifically, do the power profiles indicate a sharp peak at precisely the direction of the source α_T to help localize the source? Indeed, it is easy to observe from Eqn. 7.7 and 7.8 that $P(\alpha)$ achieves a maximum value of $1/d^4$, when $\alpha = \alpha_T$, since all components $\hat{h}_i e^{\frac{+j2\pi}{\lambda} r \cos(\alpha - \phi_i)}$ add up constructively in phase, just as in a circular antenna array. As α deviates from α_T, these components gradually begin to add up destructively. Of course, the precise shape of $P(\alpha)$ over α depends on how the user physically re-orients the device, i.e. the distribution of orientations ϕ_i. In fact, one can show that if we expect the user to simply twist the device randomly, so that ϕ_i's at least span a semi-circle (i.e. an angle of π)[4], the following lemma holds:

Lemma 7.1 *Suppose the orientation $\phi_i \sim \mathcal{U}(\gamma - \pi/2, \gamma + \pi/2)$, at snapshot $i = 1, \ldots, n$ is uniformly distributed for some constant γ, the expected value of the multipath power profile $E[P(\alpha)]$ in line-of-sight, at each $\alpha \in [0, 2\pi]$ over the distribution of $\{\phi_1, \ldots, \phi_n\}$ is given by:*

$$E[P(\alpha)] = \frac{1}{d^4}\left[J_0(\kappa)^2 + \frac{1 - J_0(\kappa)^2}{n}\right] \tag{7.9}$$

$$\textit{Where, } \kappa = \frac{4\pi r \sin\left(\frac{\alpha_T - \alpha}{2}\right)}{\lambda} \tag{7.10}$$

And $J_0(\kappa)$ is the Bessel-function of the first kind[20]. □

Proof of Lemma 10.1: In this section, we derive the expected value of SAR multi-path power profiles for line-of-sight scenarios as stated in Lemma 10.1. We consider the scenario in Fig. 7-3, where a transmitting source T is in line-of-sight relative to a two-antenna receiver device. By substituting the relative wireless channel from Eqn. 7.7 into the multi-

[4] Our algorithm requires the user to twist the device by an angle of at least π (i.e. a semicircle). We then sample the channels gathered by the device such that the resulting distribution of orientations resembles a uniform distribution over some interval of π.

path power profile from Eqn. 7.8, and simplifying terms we get:

$$\hat{h}_i = h_{2,i}h_{1,i}^* = \frac{1}{d^2}e^{\frac{-j2\pi r cos(\alpha_T - \phi_i)}{\lambda}} \tag{7.11}$$

Thus, the multi-path power profile (See Eqn. 7.8) is:

$$P(\alpha) = \frac{1}{d^4}\left|\frac{1}{n}\sum_{i=1}^{n} e^{-j\kappa \sin\psi_i}\right|^2 = \frac{1}{d^4}\left|\frac{1}{n}\sum_{i=1}^{n} e_i\right|^2 \tag{7.12}$$

Where $\kappa = \frac{4\pi r \sin\left(\frac{\alpha_T - \alpha}{2}\right)}{\lambda}$, is independent of device orientation, $\psi_i = \phi_i - \frac{\alpha_T + \alpha}{2}$ and $e_i = e^{-j\kappa \sin\psi_i}$.

At this point, our goal is to evaluate the expected value of $P(\alpha)$ over the device orientation between snapshots. To do this, let us consider the case where the snapshots are taken at random device orientations spanning an angle of π (i.e. a semi-circle) so that $\phi_i \sim \mathcal{U}(\gamma - \pi/2, \gamma + \pi/2)$ is chosen uniformly at random. Since γ depends on ϕ_0, we choose ϕ_0 such that, $\psi_i \sim \mathcal{U}(-\pi/2, \pi/2)$ is also a uniformly distributed random variable. Under these assumptions, let us look at the distribution of the quantity $e_i = e^{-j\kappa sin\psi_i}$. Specifically, the mean and variance of e_i is given by $E[e_i] = \frac{1}{\pi}\int_{-\pi/2}^{\pi/2} e^{-j\kappa sin\psi} d\psi = J_0(\kappa)$ and $\text{Var}[e_i] = E[|e_i|^2] - |E[e_i]|^2 = 1 - J_0(\kappa)^2$ respectively, where $J_0(\cdot)$ is the well-known Bessel-function of the first kind[20].

Next, we look at $\frac{1}{n}\sum_{i=1}^{n} e_i$. Applying central limit theorem for large n, this quantity is normally distributed with mean $J_0(\kappa)$ and variance $\frac{1-J_0(\kappa)^2}{n}$. Specifically, the variance of this distribution:

$$\begin{aligned}
\text{Var}\left[\frac{1}{n}\sum_{i=1}^{n} e_i\right] &= E\left[\left|\frac{1}{n}\sum_{i=1}^{n} e_i\right|^2\right] - \left|E\left[\frac{1}{n}\sum_{i=1}^{n} e_i\right]\right|^2 \\
\frac{1-J_0(\kappa)^2}{n} &= d^4 E[P(\alpha)] - J_0(\kappa)^2 \\
\tilde{P}(\alpha) = E[P(\alpha)] &= \frac{1}{d^4}\left[J_0(\kappa)^2 + \frac{1-J_0(\kappa)^2}{n}\right]
\end{aligned}$$

Which desired expected SAR profile, as in Lemma 10.1. □

Notice that as the number of snapshots n increases, the first term dominates and the expected power profile resembles a squared Bessel function $J_0(\kappa)^2$. Visually, this function resembles a squared sinc-function with a sharp peak when $\kappa = 0$, i.e. $\alpha = \alpha_T$. Fig. 7-3 plots the expected profile for a line-of-sight source at $\alpha_T = -30°$. The profile indeed resembles

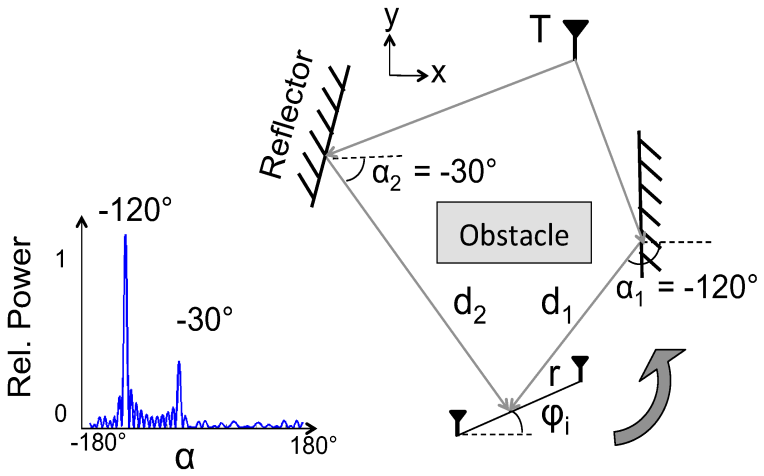


Figure 7-4: Multipath Scenarios. Depicts the path lengths d_k's and direction of the sources α_k's for a simple multipath scenarios. The corresponding power profile reveals peaks at the desired α_k values.

the squared Bessel function with a clear peak at $\alpha = -30°$.

7.3.2 Extending to Multipath Scenarios

While our discussion so far has considered a source at line-of-sight (i.e. free-space scenarios), in this section, we focus on multipath scenarios (i.e. non-free space scenarios). Specifically, we show that Ubicarse continues to be resilient to translation in these scenarios, while generating correct multipath profiles, indicating the direction of the various paths of the signal.

Assume that the device receives signals from m distinct paths, of length $d_1 \ldots, d_m$, arriving along directions $\alpha_1, \ldots, \alpha_m$, as in Fig. 7-4. Hence, SAR's goal is to infer the set of signal directions $\alpha_1, \ldots, \alpha_m$ based on the wireless channels and device orientation.

Suppose the device captures n snapshots of the wireless channels as the user twists it. As before, we denote by $(\Delta x_i \text{and} \Delta y_i)$, the translation of the device along the x and y axes and by ϕ_i, the orientation of the device relative to its initial position and orientation during the i^{th} snapshot. We then express the wireless channels on the two receive antennas as:

$$h_{1,i} \approx \sum_{k=1}^{m} \frac{s_k}{d_k} e^{\frac{-j2\pi}{\lambda}(d_k + \frac{\Delta y_i}{\sin \alpha_k})} \tag{7.13}$$

$$h_{2,i} \approx \sum_{k=1}^{m} \frac{s_k}{d_k} e^{\frac{-j2\pi}{\lambda}(d_k + \frac{\Delta y_i}{\sin \alpha_k} + r\cos(\alpha_k - \phi_i))} \tag{7.14}$$

Where s_k is a constant complex number that refers to the attenuation and phase shift along signal path k ($k = 1, \ldots, m$) and λ is the wavelength of the signal, as before.

We can then compute the relative wireless channels $\hat{h}_i = h_{2,i}h^*_{1,i}$ that we plan to feed into SAR as:

$$\begin{aligned}\hat{h}_i =& \sum_{k=1}^{m} \frac{s_k}{d} e^{\frac{-j2\pi}{\lambda}(d + \frac{\Delta y_i}{\sin\alpha_k} + r\cos(\alpha_k - \phi_i))} \sum_{k=1}^{m} \frac{s_k}{d} e^{\frac{+j2\pi}{\lambda}(d + \frac{\Delta y_i}{\sin\alpha_k})} \\ =& \sum_{k=1}^{m} \frac{s_k}{d_k} e^{\frac{-j2\pi}{\lambda} r\cos(\alpha_k - \phi_i)} \left[\frac{s_k}{d_k} + \sum_{l \neq k} \frac{s_l}{d_l} e^{\frac{+j2\pi}{\lambda}(\frac{\Delta y_i}{\sin\alpha_l} - \frac{\Delta y_i}{\sin\alpha_k})} \right] \end{aligned} \qquad (7.15)$$

Notice that the first term of the relative channel in Eqn. 7.15 above is nearly identical to Eqn. 7.7 in the line-of-sight scenario, and is independent of any translation $\Delta x_i, \Delta y_i$ between channel snapshots. Unfortunately, the second term indeed depends on translation. However, two observations work in our favor: First, if the device's translation remains constant between snapshots, the second term reduces to a constant multiplier, which merely scales the profile. Second, even if the device translation varies between snapshots, any variance (i.e. noise) caused by the second term drops significantly when summing over a large number of snapshots.

Surprisingly, these observations can help us show that by performing SAR over several snapshots, Ubicarse is resilient to device translation in multipath scenarios. Further, this property holds regardless of how the device is actually translated by the user as she twists the device. Of course, we stress that this translation cannot be unbounded, and must be relatively small compared to the distance of the device from the transmitting source, a standard assumption made by SAR and antenna array systems [188, 50]. More formally, for a device that is twisted randomly by the user over an angle of π or more, the following lemma holds:

Lemma 7.2 *Suppose the orientation $\phi_i \sim \mathcal{U}(\gamma - \pi/2, \gamma + \pi/2)$, at snapshot $i = 1, \ldots, n$ is uniformly at random for some constant γ, the expected value of the multipath power profile $E[P(\alpha)]$*

in multipath scenarios, at each $\alpha \in [0, 2\pi]$ over the distribution of $\{\phi_1, \dots, \phi_n\}$ is given by:

$$E[P(\alpha)] = \left(\sum_{k=1}^{m} \left[\frac{s_k^2}{d_k^2} + E_k \right] J_0(\kappa_k) \right)^2 + \frac{1}{n} Var[\hat{h}_i e^{\frac{j2\pi r \cos(\alpha - \phi_i)}{\lambda}}]$$

$$\textit{Where, } \kappa = \frac{4\pi r \sin\left(\frac{\alpha_T - \alpha}{2}\right)}{\lambda}, E_k = E\left[\sum_{l \neq k} \frac{s_k s_l}{d_k d_l} e^{\frac{j2\pi \Delta y_i \left(\frac{1}{\sin \alpha_l} - \frac{1}{\sin \alpha_k}\right)}{\lambda}} \right]$$

And $J_0(\kappa)$ is the Bessel-function of the first kind[20]. □

Proof of Lemma 7.2: We derive the expected SAR power profiles in multipath scenarios as in Lemma 7.2. Recall the relative wireless channel between two antennas of the receiver at snapshot i is given by Eqn. 7.15, such that the multipath profile $P(\alpha)$, depends on the sum of terms of the form $g_i = \hat{h}_i e^{\frac{j2\pi r \cos(\alpha - \phi_i)}{\lambda}}$ (Eqn. 7.8). Each term is:

$$g_i = \sum_{k=1}^{m} D_k e^{\frac{-j2\pi}{\lambda} r(\cos(\alpha_k - \phi_i) - \cos(\alpha - \phi_i))} = \sum_{k=1}^{m} D_k e^{-j\kappa_k \sin \psi_{i,k}}$$

Where similar to Appendix 7.3.1, $\kappa_k = \frac{4\pi r \sin\left(\frac{\alpha_k - \alpha}{2}\right)}{\lambda}$ and $D_k = \frac{s_k}{d_k} \left[\frac{s_k}{d_k} + \sum_{l \neq k} \frac{s_l}{d_l} e^{\frac{+j2\pi}{\lambda} \left(\frac{\Delta y_i}{\sin \alpha_l} - \frac{\Delta y_i}{\sin \alpha_k} \right)} \right]$, depend only on translation, while $\psi_{i,k} = \phi_i - \frac{\alpha_k + \alpha}{2}$ depends only on orientation. As a result, D_k and $e^{-j\kappa_k \sin \psi_{i,k}}$ are independent, and we can write: the expectation $E[g_i] = \sum_{k=1}^{m} E[D_k] E[e^{-j\kappa_k \sin \psi_{i,k}}] = \sum_{k=1}^{m} \left[\frac{s_k^2}{d_k^2} + E_k \right] J_0(\kappa_k)$, assuming that ϕ_i's are chosen uniformly at random. Here we denote $E_k = E\left[\sum_{l \neq k} \frac{s_k s_l}{d_k d_l} e^{\frac{j2\pi \Delta y_i \left(\frac{1}{\sin \alpha_l} - \frac{1}{\sin \alpha_k}\right)}{\lambda}} \right]$.

Finally, let us look at the summation $\frac{1}{n} \sum_{i=1}^{n} g_i$. Applying central limit theorem for large n, this quantity is normally distributed with mean $\sum_{k=1}^{m} \left[\frac{s_k^2}{d_k^2} + E_k \right] J_0(\kappa_k)$ and variance $\frac{Var[g_i]}{n}$. Specifically, the variance of this distribution is:

$$\text{Var}\left[\frac{1}{n} \sum_{i=1}^{n} g_i \right] = E\left[\left| \frac{1}{n} \sum_{i=1}^{n} g_i \right|^2 \right] - \left| E\left[\frac{1}{n} \sum_{i=1}^{n} g_i \right] \right|^2$$

$$E[P(\alpha)] = \left(\sum_{k=1}^{m} \left[\frac{s_k^2}{d_k^2} + E_k \right] J_0(\kappa_k) \right)^2 + \frac{1}{n} \text{Var}[\hat{h}_i . e^{\frac{j2\pi r \cos(\alpha - \phi_i)}{\lambda}}]$$

Which desired expected SAR profile, as in Lemma 7.2. □

Over a large number of snapshots n, the expected multipath profile resembles the squared sum of Bessel functions (which resemble sinc functions) with peaks corresponding to the different multipath directions scaled by their relative signal power. In other words,

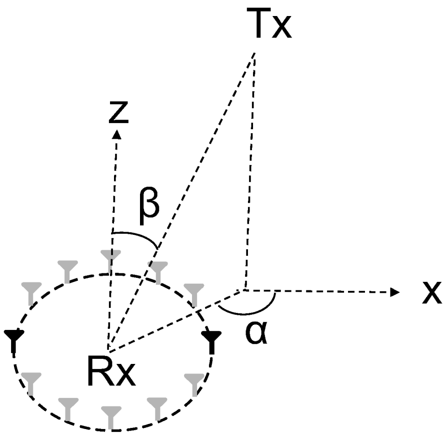


Figure 7-5: Generalizing to 3-D. Depicts the definition of α and β, respectively the azimuthal and polar angle from which the signal arrives.

by inspecting the peaks of the profile $P(\alpha)$ one can infer the different multipath directions $\alpha_1, \ldots, \alpha_m$. Fig. 7-4 plots the profile for a non-line-of-sight source with two paths along -30° and -120°. Notice that we indeed observe distinct peaks at each of the two angles, scaled by their relative signal strength.

■ 7.3.3 Generalizing to Three Dimensions

Our above solution can be readily generalized to three-dimensions. Such a generalization is particularly important, since it is difficult to restrict users of Ubicarse to perfectly rotate their devices in a two dimensional plane. To better understand our solution, consider a two-antenna receiver device, as before and a signal path arriving from azimuthal angle α and polar angle β, as shown in Fig. 7-5. Suppose the user twists the device about a given axis.[5] Let ϕ_i be the current azimuthal angle of the second receive antenna relative to the first and θ_i denote the polar angle, during the i^{th} snapshot. From basic geometry, we can derive the difference between the path lengths to the two antennas is $r\cos(\alpha - \phi_i)\sin(\beta - \theta_i)$, where r is the separation between the two antennas. As a result, we can derive the power-profile along the direction (α, β) by slightly modifying Eqn. 7.8 as:

$$P(\alpha, \beta) = \left| \frac{1}{n} \sum_{i=1}^{n} \hat{h}_i e^{\frac{+j2\pi}{\lambda} r \cos(\alpha - \phi_i) \sin(\beta - \theta_i)} \right|^2 \tag{7.16}$$

[5] In principle, the user can twist the device about any axis, provided the axis of rotation is not parallel to the line joining the two antennas, i.e. so that the relative rotation of the two antennas is non-zero.

Notice that the SAR power profile equation indeed resembles the expression for circular SAR in 3-D [45], as before. Further, note that much like standard circular antenna arrays [45], even if the user rotates the device entirely on a 2-D plane (e.g., the xy-plane), the above expression returns the angle-of-arrival in a 3-D space.

■ 7.3.4 Robustness to Frequency Offsets

Recall that a wireless receiver (e.g., the user's tablet) experiences a carrier frequency offset (CFO) and sampling frequency offset (SFO) with respect to a wireless transmitter (e.g., an access point) [141]. The CFO and SFO cause an additional phase rotation, independent of the antenna movement, which unless eliminated can accumulate over time, causing errors in the SAR output. Fortunately, since the two antennas on the mobile device experience the same offsets with respect to the source, taking the relative channel eliminates that effect. Consequently, Ubicarse's SAR is robust to frequency offsets as well.

■ 7.4 Active Drift Compensation

In this section, we address the following challenge: The error in orientation reported by motion sensors accumulates gradually over time, leading to significant errors in SAR. This is because gyroscopes measure angular velocity and any systematic error in this velocity integrates over time and leads to a drift in orientation [56]. However, over time windows of a few seconds, this drift can be approximated as an unknown but linear shift in orientation [128]. Ubicarse leverages this property to actively estimate and correct for drift in real-time, just as the user twists her device.

Ubicarse resolves orientation drift by making a key observation: A linear drift in orientation leads to a constant shift in SAR multipath profiles generated by Ubicarse. Specifically, suppose we perform SAR over two different time windows to obtain two multipath profiles. Let's say that the gyroscope incurred a drift of δ between these intervals. Recall from Eqn. 7.8 that any multipath profile $P(\alpha)$ depends on angles of the form $\alpha - \phi_i$, where ϕ_i is the orientation of the device during the i^{th} snapshot. Consequently, if each ϕ_i is shifted by an offset δ between the two time windows, then α is also shifted by the same value δ between them. In other words, any accumulation of drift δ between the two intervals effectively results in a shift of precisely δ between the two multipath profiles. Hence,

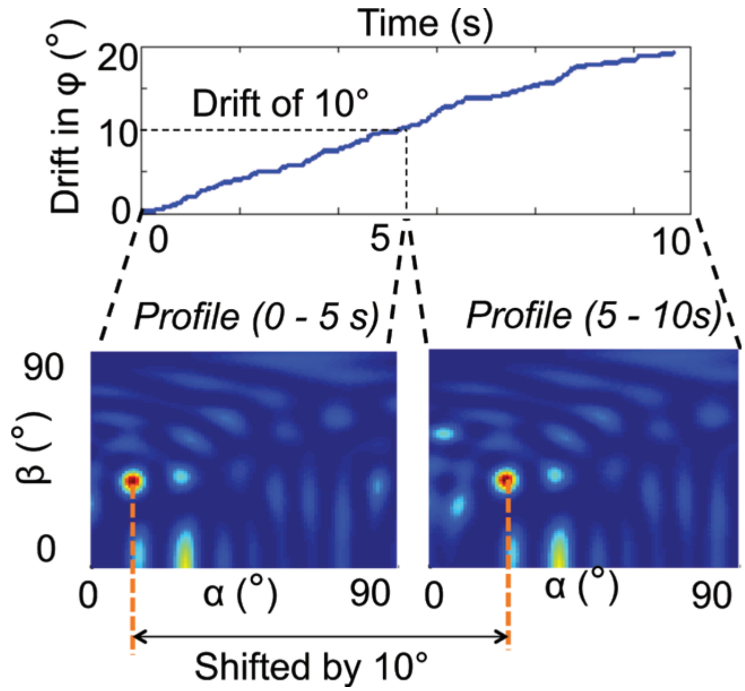


Figure 7-6: Drift in Orientation. The figure (above) plots the drift error over time for 10 seconds. The accumulated drift is 10° between two $5s$ intervals. The 3-D multipath profiles corresponding to these intervals (below) indicate a shift of precisely 10° (profiles zoom-in over a small range for clarity).

by estimating the shift δ between the two multipath profiles, we can accurately estimate, and therefore correct for the gyroscope drift between the two intervals.

To illustrate this in an example, Fig. 7-6 depicts the orientation in drift of 10° between two 5 second intervals during one of our experiments. We observe that the corresponding multipath profiles are also shifted by about 10°. We can then directly estimate and actively correct for this drift in real-time by applying phase correlation [53], a standard technique from computer graphics to estimate shifts between two (noisy) images by calculating their phase difference in the 2-D Fourier domain.

Finally, note that since Ubicarse performs drift correction in real-time, it does not rely on predetermined characteristics of the gyroscope and is thus general to different hardware configurations.

■ 7.5 Accurate Device Localization

In this section, describe how Ubicarse leverages SAR multipath profiles to accurately localize the device. Ubicarse first obtains these profiles by asking the user to twist the device about its vertical axis. It then issues beacon frames to multiple neighboring access points, to elicit response frames and estimate channels from these access points. It then simultaneously performs SAR to these access points and obtains multipath profiles. In the absence of multipath, each profile contains a single peak towards the direction of the corresponding access point. Ubicarse can then apply standard triangulation [96] to localize itself relative the access points. In principle, Ubicarse needs to perform SAR to a minimum of three surrounding access points to localize the device.

Localization under Multipath. In practice, SAR profiles may contain multiple peaks owing to multipath. As a result, there exists an inherent ambiguity in the direction of these access points relative to the device. Ubicarse addresses this challenge using the following two key observations.

First, as noted in [188], the multipath peaks in an antenna-array profile can be differentiated from the direct path peak using the following observation: The direct peak in a profile persists even as the receiving device's position is slightly perturbed, whereas multipath peaks change significantly or disappear altogether (Fig. 9 in [188]). In fact, as Ubicarse explicitly requires a user to twist the device to perform SAR, such perturbations are inherent to our system and this is a fact that we exploit.[6] Therefore, by performing SAR on two different segments of the device's trajectory (emulating two antenna arrays at slightly different locations), one can compare the resulting multipath profiles to eliminate peaks that do not persist between these profiles.

Second, often devices are surrounded by more than three access points. This leads to an over-constrained optimization problem for triangulation, which can be formulated as a least-square fitting problem [96]. Notice that the peaks corresponding to the direct path to each of many different access points agree on a unique location for the device, and therefore lead to a good fit. In contrast, multipath peaks of the access points indicate conflicting locations for the device leading to poor fits. Therefore, by selecting the set of peaks to access points that achieve the best fit in the optimization problem (i.e. agree the most geometrically), Ubicarse can identify the set of direct paths to these access points.

Obtaining Global Device Orientation. One of the key benefits of Ubicarse is the ability to not just find the device's global position, but also its global orientation, relative to the

[6] Notably however, the direct path peak does experience a shift due to drift as discussed in §7.4, although the peak's magnitude remains the same, and its shift is consistent.

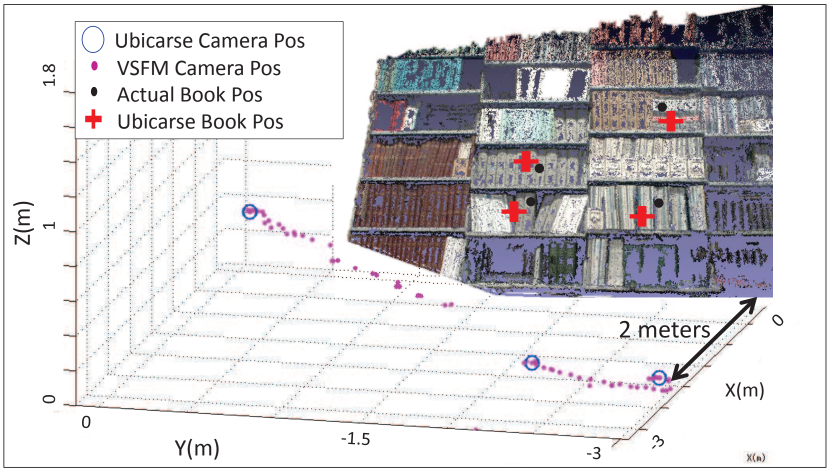


Figure 7-7: Object Geotagging. Point cloud of library shelving and VSFM reconstruction with camera positions and anchored camera positions. Anchored camera positions are those camera positions with both Ubicarse localization coordinates in the global frame and relative coordinates in the coordinate frame of the 3D reconstruction. These anchor points are used in Equation (7.17) to find a suitable transform to a the global frame.

access points' frame of reference. Notice that global orientation is not available from the device's gyroscope, which only measures angular velocity, and therefore can only be used to find relative orientation between two time intervals.[7] The key reason Ubicarse can readily estimate device orientation is that it computes the angle-of-arrival of signals from the access points. In particular, Ubicarse knows the angle of the access point, relative to the axis of its own body . But after performing localization, Ubicarse also knows the direction of this access point relative to the global x-axis. Hence, by simply subtracting the two angles, Ubicarse can compute its orientation ψ relative to the world's x-axis.

■ 7.6 Application to Object Geotagging

In this section, we show how Ubicarse's accurate indoor localization of mobile devices opens the door to a new class of applications: precision geotagging of indoor objects with no RF source attached to them. This would enable users to catalog and share the location of objects of interest they observe around them, i.e., products in a warehouse or artifacts in a museum. To enable this application, we leverage the fact that in addition to Wi-Fi chips and gyroscopes, most wireless devices also have cameras on them. Today, vision

[7] While the device's compass can be used to compute global orientation, such compasses are fairly inaccurate [79] and easily misled by any surrounding magnetic fields.

toolkits can be applied to overlapping photos taken from these cameras to produce 3D pointclouds, or *relative* 3D positions, for imaged objects [171].

In particular, these vision toolkits use multiple snapshots taken of a single scene from different vantage points in order to reconstruct the 3D positions of objects in the image as a point cloud. While these tools are impressively accurate at inferring relative distances and orientations in the local coordinate frame of the camera, they cannot provide information on the positions of the camera or photographed objects in the *global* frame that a user cares about. For example, a vision-based reconstruction can accurately determine that the biology magazine stack is one foot away from the physics magazine stack, but cannot determine that these magazine stacks are located in shelf-5 of the mezzanine level room in the NY public library. In contrast, Ubicarse's core ability is that it can determine the global position of the device's camera at a given point in time. Hence we can synergize the fine-grained relative distances of object-to-camera computed by the vision toolkit, with the accurate global camera-position provided by Ubicarse. This allows for a joint optimization problem that provides the global position of the objects of interest. Surprisingly, we show that this joint optimization has an interesting side-effect: one can further refine Ubicarse's device localization by using the relative camera positions between the different snapshots, that are output by the vision toolkit. This leads to an integrated system that capitalizes upon the best of both technologies for device and object localization.

Vision Algorithms. To better understand how object localization works, we provide a briefly introduce vision algorithms for 3-D imaging. The goal of these algorithms is to read a set of 2-D images of the object snapped by the device's camera from multiple perspectives to obtain two outputs:

- *3-D reconstruction of the object*: obtained in the frame of reference of the device camera. This reconstruction is a 3-D point cloud (Fig. 7-7), containing the (x, y, z) coordinates of points on the object's surface in a 3-D frame of reference whose origin is at the camera's initial position and axes along its initial orientation.
- *Relative Camera Positions and Orientation*: of the vantage points where the 2-D images were taken, defined relative to the camera's initial position and orientation.

At a high level, vision algorithms obtain these quantities in three phases: First, they identify salient features [11] (e.g. corners or distinctive textures) in each of the images. Next, these features are used to identify positions of the same object between these images. Finally, they apply advanced algorithms such as [171], which uses the distance of the same features to the different camera positions from where the images were taken, to infer relative object positions and camera positions.

Accurate Object Localization. Once a 3D pointcloud of the imaged objects is constructed via the vision algorithm, a suitable transformation must be found in order to map the object locations into the (global) reference frame of the environment. To do this, we employ Ubicarse's translation-resilient localization in N different positions and orientations where the snapshots were taken (assuming N is at least three). The N camera positions can now serve as anchor points, points for which we have both global positions $\{x_{g_1}, \ldots, x_{g_N}\}$ provided by Ubicarse, and relative positions, $\{x_{p_1}, \ldots, x_{p_N}\}$ in the coordinate frame of the reconstructed point cloud. A relationship (or transformation) between the two coordinate systems can be found via the following optimization problem that can readily be solved for the unknown transformation (rotation R and translation t) that minimizes the following error in fit (see Algorithm 3):

$$\min_{R,t} \sum_{i=1}^{N} ||x_{g_i} - (Rx_{p_i} + t)||^2 \tag{7.17}$$

In general the more anchor points you have, the more accurately you can find the transformation. [8] The rotation R and translation t are now the key relationship between the point cloud and global coordinate frames and in particular, it can be applied to any object in the point cloud in order to find the global position of that object.

Refining Device Localization. Finally, we note that the above joint optimization has an interesting side-effect: Ubicarse can leverage the relative camera locations output by vision algorithms to greatly refine its own device localization. Specifically, visual algorithms excel at finding relative camera locations, particularly when they observe a rich set features at pixel resolution. Therefore, by applying the transformation to the camera positions returned by VSFM we now have two sets of position estimates for the camera, both in the global reference frame. In this way, one can accurately compensate for relative errors in Ubicarse's indoor positioning. Indeed, our results in §7.8.5 show that such an optimization

[8] In practice, the above optimization also needs to account for camera calibration and scaling, whose details we omit for brevity.

greatly improves median localization accuracy by as much as 86% as compared to just using Ubicarse alone.

3 Object Geo-tagging Algorithm. Finding the transformation, rotation R and translation t, between local and global coordinates

```
function GETTRANSFORM($X_g$, $X_p$)
    ▷ $X_g = \{x_{g_1}, \ldots, x_{g_N}\}$: Set of $N$ positions in global frame
    ▷ $X_p = \{x_{p_1}, \ldots, x_{p_N}\}$: Set of $N$ positions in local camera frame
    $C = \sum_{i=1}^{N} (x_{g_i} - \text{mean}(X_g))(x_{p_i} - \text{mean}(X_p))^*$
    ▷ Where $(.)^*$: conjugate transpose, and mean$(.)$: average of elements
    $[U, S, V] = \text{svd}(C)$                      ▷ Perform SVD of covariance matrix $C$
    $R = VU^*$                                        ▷ Rotation Matrix
    $t = \text{mean}(X_p) - R \times mean(X_g)$       ▷ Translation Vector
    return $R, t$
end function
```

7.7 Implementation and Evaluation

We implemented Ubicarse on a HP SplitX2 Tablet running Ubuntu Linux equipped with Intel 5300 wireless cards and a Yei Technology motion sensor (i.e. an accelerometer, gyroscope and compass). We build on the 802.11 CSI tool [67] to obtain the wireless channels on the tablet. We implement Ubicarse's translation resilient SAR completely in software (C++ and Matlab). For each experiment, we allow users to twist their devices randomly along its vertical axis, spanning a wide range of orientations. We configure the tablet to transmit beacon packets (10 times per second) to elicit responses from nearby access points and measure their channels.[9] We then combine this with gyroscope readings from the motion sensor to perform SAR *simultaneously* to all access points, as the user twists the device.

We implement Ubicarse's object localization by integrating it with the VisualSFM (VSFM) toolkit [186] for 3-D reconstruction of objects. We use the tablet's built-in 5.7 Megapixel camera to take images of objects of interest.

We conducted our experiment in a university library, with books arranged in multiple shelves and racks that emulate the complex multipath characteristics of large warehouses and departmental stores, where indoor localization has been of particular interest, as shown in Fig. 7-8. We leverage five unmodified 802.11n access points in and around the library, as shown by red squares in Fig. 7-8 to perform device localization and vary the tablet's location between several randomly chosen locations. The access points are configured to the 5 GHz Wi-Fi frequency band. We obtain the locations of access points and tablet centers to sub-centimeter accuracy in a global coordinate frame through direct measurements and using architectural drawings of the library. To perform object lo-

[9] The results in this chapter sample the wireless channels for a minimum of 25-30 packets over a semi-circular arc of rotation.

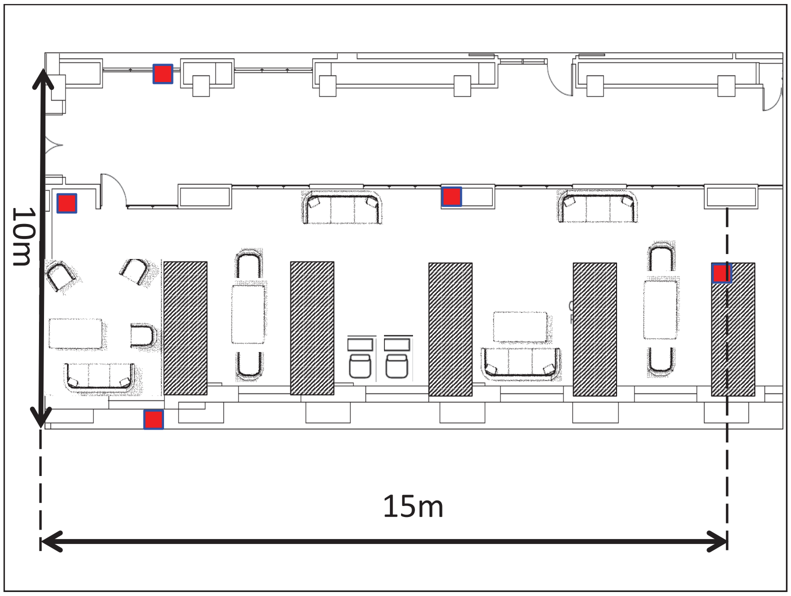


Figure 7-8: Library floorplan and testbed. APs are red squares; book shelves are shown as hashed rectangles.

calization, we capture images from multiple perspectives of different randomly chosen books in the library. We then apply a joint optimization using VSFM's reconstructed 3D point cloud and Ubicarse's device localization to locate the object.

Baseline: We compare Ubicarse against an angle-of-arrival scheme that treats the two antennas on the device as a 2-antenna array. We chose this baseline, since it uses only 2-antennas like our system, does not require specialized infrastructure or additional calibration input unlike other schemes, leading to a fair comparison.

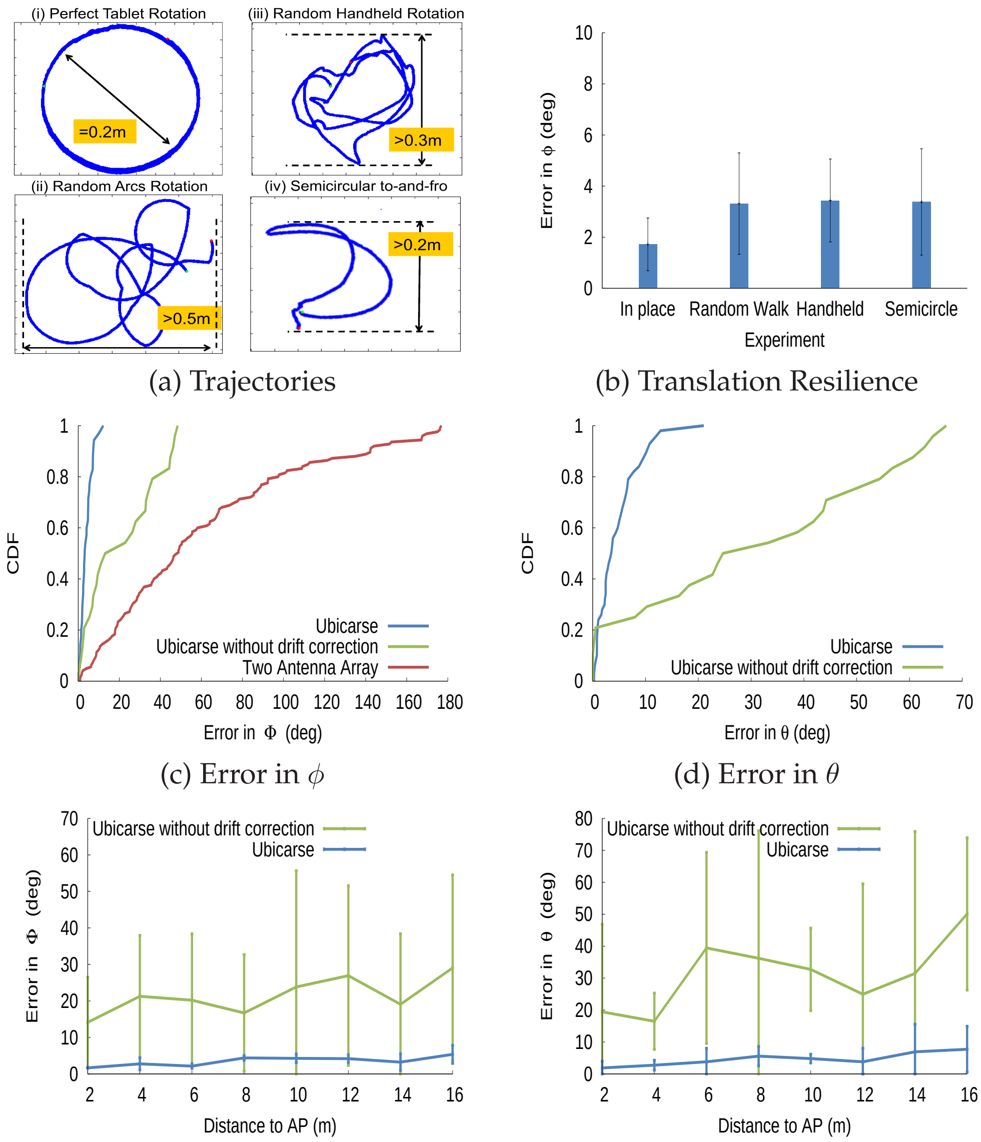
■ 7.8 Results

We present four categories of results to evaluate Ubicarse: 1) Validating Translation Resilience of Ubicarse's SAR for different trajectories. 2) Computing Angle of Arrival in 3-D space. 3) Localizing devices in 3-D. 4) Geotagging Objects of Interest.

■ 7.8.1 Translation Resilient SAR

In this experiment, we demonstrate that Ubicarse's new formulation of SAR is resilient to device translation, as described in §7.3. Specifically, we consider different device trajectories, including random motion of the antenna's center on the order of half a meter.

Method: We perform our experiments in a mm-precision motion capture room [4] where the mobile device was tagged with infrared markers to accurately track the motion of the antenna centers for the purpose of evaluation. SAR was computed entirely using one YEI technologies



(a) Trajectories

(b) Translation Resilience

(c) Error in ϕ

(d) Error in θ

(e) Error in ϕ vs. distance

(f) Error in θ vs. distance

Figure 7-9: Microbenchmarks. (a) Plots the different trajectories of antenna centers (b) Translation resilience of SAR: Plots mean and Standard deviation of error in azimuthal angle for different trajectories. (c)-(d) CDF of error in Angle of arrival (ϕ and θ), for Ubicarse, Ubicarse without drift correction and 2-Antenna Array. (e)-(f) Error in azimuthal, polar angle versus distance from the access point.

gyroscope mounted on the tablet and channel measurements acquired from the Wi-Fi card. A single transmitting source was present and the experiment was conducted at distances of 2m to 12m from the source across multiple line-of-sight locations (we consider non-line-of-sight in §10.5.1).

We moved the tablet in four different types of trajectories: (1) In-place rotation; (2) 2D random trajectory within 0.5 m diameter; (3) A random handheld 3D twisting motion (i.e. including variations in polar angles) up to 0.3 m in diameter (4) A handheld semi-circular to-and-fro twist. For the first two trajectories, we mount the device on a roomba robot for stable trajectories on a 2-D plane.

Fig. 7-9(a) demonstrates actual trajectory traces from our experiments where these traces are representative of each of the four classes of trajectories described above.

Results: Our results in Figure 7-9(b) show a mean error of $3.3°$ for the random 2-D trajectory, $3.4°$ even as the center of the antennas is translated randomly by the user in three dimensional space and $3.3°$ if the user traces a simple semi-circular to-and-fro twist. Our results demonstrate that Ubicarse's translation resilient SAR correctly identifies the angle of arrival for the two cases to within $1.6°$, $1.7°$ and $1.6°$ respectively on average as compared to the ideal case of rotation in place.

■ 7.8.2 Angle of Arrival Estimation on a Tablet

We measure the accuracy of angle-of-arrival estimation in both the azimuthal angle (ϕ) and polar angle (θ) obtained over each handheld twist of the device in line-of-sight (LOS) and non-line-of-sight (NLOS) settings.

Method: We conduct our experiment in the university library as in §7.7. We place the device at different distances from the access points. In NLOS settings, Ubicarse identifies the direct path using the methods in §7.5. We compare our results against two important benchmarks: 1) A 2-antenna array composed of the two antennas on the device directly, as opposed to performing SAR. 2) Ubicarse without the gyroscope drift compensation presented in §7.4.

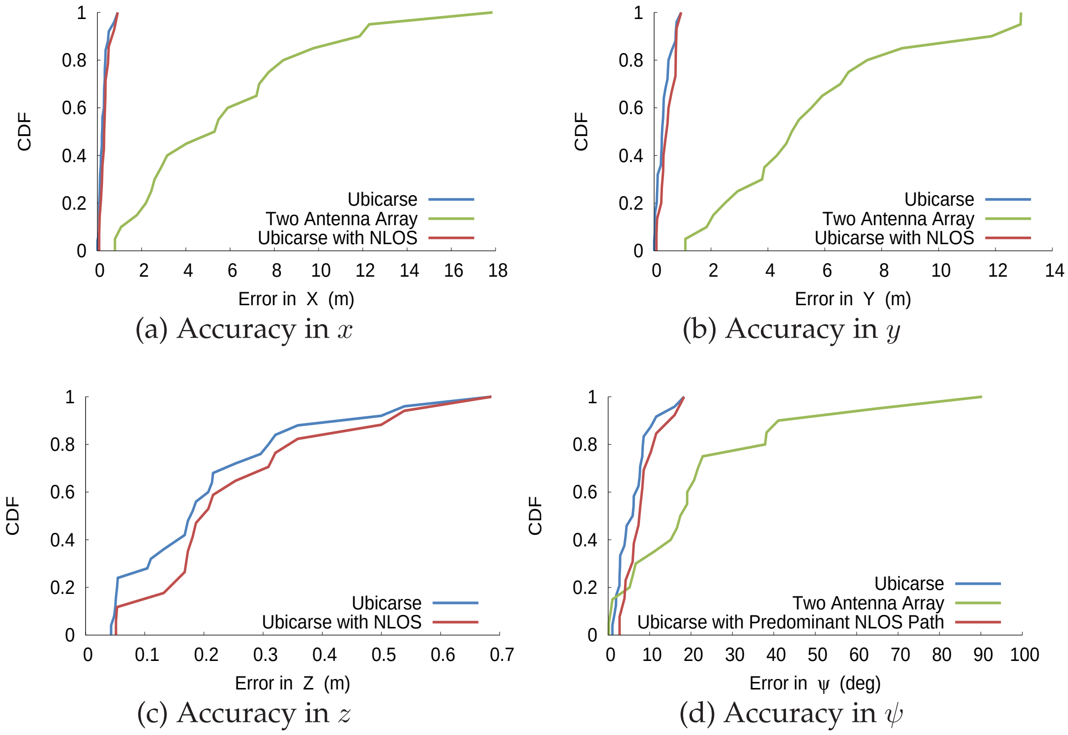
Results: Fig. 7-9(c)-(d) shows that our method achieves a median of accuracy of $3.2°$ in the azimuthal orientation of the device and $3.6°$ in polar angle. We observe that gyroscope drift can cause angle inaccuracies of more than $18°$ in ϕ (29° in θ) and as large as $48°$ in ϕ (67° in θ) if left uncompensated. As a result, Ubicarse's drift compensation in §7.4 plays an important role in achieving high accuracy. Fig. 7-9(c) also shows that the 2-antenna array has a poor median error of $49°$ in azimuthal angle owing to the lack of resolution from using only two antennas. Note that Figure 7-9(d) does not compare with the two antenna benchmark since two antennas alone are insufficient to obtain 3-D information [131].

Fig. 7-9(e)-(f) depicts the error in ϕ and θ respectively, against the distance of the device to the wireless access point. We observe that the error in these angles increases only marginally across distances ranging from $2m$ to $16m$.

■ 7.8.3 Tablet Localization

In this experiment, we evaluate Ubicarse's accuracy in device localization in the university library setting.

Method: We gather aggregate results on device localization by in twenty five randomly chosen locations where users twist a Ubicarse enabled tablet in the university library setting. Ubicarse's localization was performed to find relative position of the device with respect to commodity access points whose locations were known with respect to the library floor plan. For device and object lo-



(a) Accuracy in x

(b) Accuracy in y

(c) Accuracy in z

(d) Accuracy in ψ

Figure 7-10: Tablet Localization. Plots CDF of error in device localization.

calization we present results in the three-dimensional Euclidean space as well as global orientation (in the X-Y plane) as given by the ψ angle described in §7.5.[10] These experiments were conducted both in line-of-sight (LOS) and non-line-of-sight (NLOS) settings during peak library hours with pedestrian foot traffic. In particular, we classify locations as non-line-of-sight (NLOS) only when the predominant (highest power) peak in the multipath profile is not the direct path to at least one neighboring AP. Our localization technique explicitly handles multipath as described in section 7.5.

Results: Our results in Fig. 7-10 for localization using five surrounding access points demonstrate a median device localization error of 39 cm, where along each dimension we demonstrate a median error of 22 cm in x, 28 cm in y, and 18 cm in z and $6.09°$ in global device orientation (ψ). Furthermore we show that Ubicarse still performs well in NLOS, incurring a small additional median error of 10 cm in x, 18 cm in y, and 2 cm in z and 7.7° in ψ as compared to the LOS case. Thus, Ubicarse achieves its goal of accurate localization without requiring either new infrastructure or signal fingerprinting.

Fig. 7-11 plots Ubicarse's error in localization along x, y and z measured against the number of randomly chosen access points used to triangulate the location of the device. We notice that the error in localization reduces with increasing number of surrounding access points. This is because,

[10] In principle, Ubicarse can be used to find orientations along other planes as well. We omit these results for brevity.

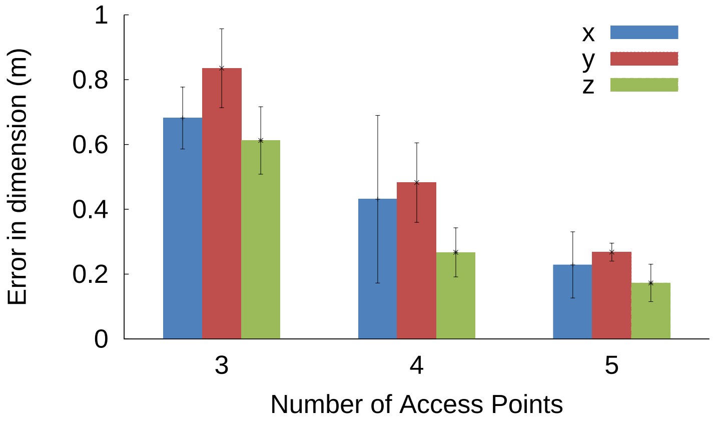


Figure 7-11: Device Loccalization Accuracy. Ubicarse's Device Localization Accuracy measured against the number of wireless access points.

as described in §7.5, computing multipath profiles to an increasing number of surrounding access points, can help mitigate ambiguity due to multipath. Further, redundant measurements from over three APs reduces the noise in localization.

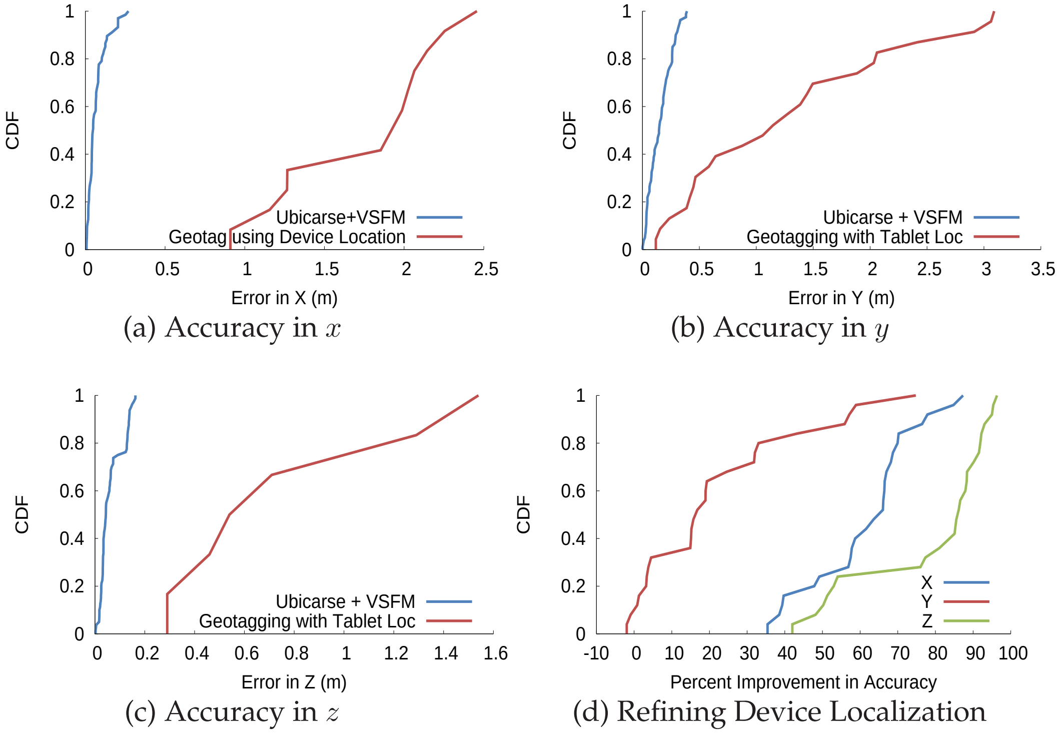
■ 7.8.4 Object Geotagging

In this experiment, we evaluate Ubicarse's object geotagging capabilities (See §7.6) to localize arbitrarily chosen books on a bookshelf in the library setting.

Method: We conduct our experiments in a library setting with shelving that houses many books, journals, and magazines, a picture of which can be found in Figure 7-7. This implementation emulates similar applications in a warehouse, department store, or other such settings where a person can catalog objects of interest. In particular we show that there is a distinct advantage to localizing far away objects more accurately than simply using the camera's location as a geotag. We compare the object localization we achieve using Ubicarse plus VSFM against the accuracy of using the camera's location directly as a position tag for the book.

Results: Fig. 7-12(a)-(c) shows that the method of Ubicarse plus VSFM attains a median error in distance of 17 cm, where along each dimension we find an error of 5 cm in x, 15 cm in y and 4 cm in z for localization accuracy of books on library bookshelves whose distance from the camera varies from 1 to 3 m. Therefore simply using the camera's position as a geotag for the books would result in up to 3 m of inaccuracy.

Surprisingly, a comparison of Figures 7-12 and 7-10 reveals that *a greater accuracy in object localization* can be obtained by a combination of Ubicarse with VSFM than using either method standalone. This is because these two methods are highly complimentary and a thoughtful integration unlocks an effective method for localization *refinement* (see §7.6). We provide further results on this concept in the next section.



(a) Accuracy in x

(b) Accuracy in y

(c) Accuracy in z

(d) Refining Device Localization

Figure 7-12: Object Localization Accuracy. (a)-(c)Plots CDF of error in object geotagging. (d) Demonstrates refinement of Device localization using object geo-tagging

7.8.5 Refined Tablet Localization using Vision

The previous section suggests that a method to achieve a fine-scaled object position fix is to integrate an accurate device localization method, Ubicarse, with visual toolkits like VSFM. In this experiment, we study if this improvement applies both ways, i.e. if device localization can be refined using inputs from vision toolkits.

Method. Our method for refinement of camera positions derives from the discussion in §7.6. We transform the camera locations discovered by VSFM into the global coordinate system, using Algorithm 3. We then output the transformed camera locations as the result of Ubicarse's localization.

Results. Fig. 7-12(d) shows that a Ubicarse plus VSFM integration achieves a significant improvement in localization performance with a median of 15 cm error from ground truth. We report a median improvement in localization accuracy of 66 % in X, 16% in Y and 86% in the Z dimension. Therefore, integration of Ubicarse with vision algorithms also serves to improve device localization.

7.9 Related Work

Related work falls under two broad categories:

Indoor RF Localization. RF localization have had three main approaches for indoor localization: The first require deploying specialized infrastructure to perform accurate indoor localization, e.g. acoustic [154, 107], RFIDs [179, 178], specialized access points [82], and antenna arrays [188]. The second mandates signal fingerprinting [16, 150, 197] either in a training phase or via crowd-sourcing [142, 195, 104]. The third advocate modeling the wireless signal instead of measurements [97, 28, 65], but at the expense of accuracy. In contrast to these systems, Ubicarse obviates the need for new infrastructure or fingerprinting to achieve tens of centimeter of accuracy by emulating large antenna arrays using only commodity mobile devices.

Ubicarse builds on past work in Synthetic Aperture Radar (SAR) that leverage precise mechanically controlled antenna movements or advanced motion sensing equipment for radar imaging [50] and RFID localization [179, 178]. Unlike these systems, Ubicarse provides a new formulation of SAR that is translation resilient, making it suitable for mobile devices held by users.

Several RF localization papers use mobile motion sensors as hints for localization, either coupled with Wi-Fi signal strength for fingerprinting [187, 16, 142] or with GPS measurements [32, 177]. Instead, Ubicarse uses the device's gyroscope only to measure its relative orientation to help emulate antenna arrays and achieves significantly improved accuracy.

Object Geo-tagging. There has been much work in computer vision on stereo-imaging and 3D-reconstruction to localize objects relative to known camera locations or to find relative locations of different objects in the same scene [11, 171]. Ubicarse builds on this work, coupled with RF localization, to obtain object locations in the global frame.

Several solutions for indoor device and object localization have been proposed in the literature that use specialized hardware, e.g. depth cameras [118] or odometry/laser-based simultaneous localization and mapping [46], which are unavailable for today's mobile devices, unlike Ubicarse.

7.10 Discussion

We presented Ubicarse, an indoor localization system to achieves tens of centimeters of accuracy on commodity mobile devices, without requiring specialized infrastructure or fingerprinting. Ubicarse's provides a new formulation of SAR that allows mobile devices to emulate an antenna array; where this formulation is tolerant to unknown translation of the antenna center of up to half a meter. We implement Ubicarse on a commodity tablet and demonstrate tens of centimeter accuracy in both device localization and object geotagging in complex indoor settings. We believe evaluating our system on a wide-range of mobile devices, particularly smartphones, is an important task for future work.

CHAPTER 8

New connections between wireless and robotics

Wireless networks and robotics have a long history. Any multi-robot system requires wireless networks to communicate, be it for manufacturing, agriculture, mining or search-and-rescue. Traditionally, robotics employs wireless networking to simply exchange messages.

This dissertation reveals the benefits of cracking the blackbox open. It tackles important problems in robotic by effectively using commercial Wi-Fi radios as sensors. We describe algorithms that effectively sense how wireless signals from a transmitter reflect around obstacles before they reach the receiver.

These algorithms are at the heart for two novel systems. First, we make the observation that robots can learn how to avoid obstacles by sensing how radio waves reflect around them. This enables a novel system where robots that can navigate to improve signal quality by following the wireless signals emitted by their targets, akin to how moths navigate towards a lamp, following visible light. Second, we observe that signals from a wireless transmitter carry unique information that is dependent both on the transmitter's hardware, its location and its environment (e.g. the location of obstacles around it). As a result, they can be used to extract fingerprints that remain unique to specific transmitters. We demonstrate how this primitive enables a natural authentication mechanism for robots to secure against the Sybil attack, where a malicious robot can disrupt a multi-robot network by pretending to be a large number of fake clients.

We describe the main challenges and core ideas behind both these systems below and highlight their relation to prior work.

8.1 Adaptive Multi-Robot Communication Using Signal Directionality

Multi-agent robotic systems perform many complex tasks through coordination, such as cooperative search of an environment, consensus, rendezvous, and formation control [33, 80, 129]. As cooperation is at the core of multi-robot tasks, the performance of these systems directly hinges on the robots' ability to communicate reliably. To maintain certain communication guarantees, these systems need a mapping of communication quality to robot placement. Producing such a mapping however is quite challenging [113]. State-of-the-art approaches to do so, either assume models that are mathematically simple such as the Euclidean disk model [33, 80], yet known to be impractical, or require stochastic sampling along directions counter-productive to the overall coordination goal of the robots [95, 159].

Chapter 9 presents two key contributions: i) we introduce a novel method for capturing the spatial variation of wireless signals in the local environment *without* sampling along counter-productive directions, or requiring prior information about the environment; and ii) we use this model to derive a robotic controller that can guide robots to navigate to locations so as to satisfy the heterogeneous communication demands of other robots in the network. The controller is formulated to maintain the structural (quadratic) simplicity of the Euclidean disk model while adapting to signals measured in real-world environments.

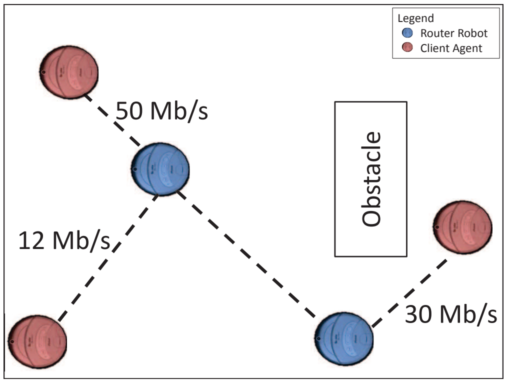


Figure 8-1: Multi-robot Network. Picture of a network of two robot routers satisfying the demands of three clients in an environment with an occluding obstacle whose position is unknown.

Our Approach: Consider a team of robot routers that want to optimize their locations to satisfy the diverse communication demands of clients in a multi-robot network (the classic communication coverage problem, like in Fig. 8-1). Our approach first calculates a mapping between a robot's current position and a profile of signal strength that it receives along *each spatial direction*, for every wireless link with other robots. We then design a controller that uses the profile to find directions of movement that yields better communication quality. Such a profile also helps estimate the confidence with which the controller can improve signal power by navigating the robot along any of these directions. The confidence can then be used to control the speed of the robot, thereby improving stability and convergence time. Furthermore, the controller can leverage the entire profile of signal strength across directions, to optimize communication with multiple robots by choosing a direction of movement corresponding to a strong signal that strikes trade-offs between competing communication demands of different pairs of robots. Interestingly, we show that such optimizations can be formulated in terms of simple quadratic costs, similar in spirit to the disk model. Further, they can be made independent of environment-dependent parameters, or even robot positions.

A key question remains: how do we calculate the signal strength along each spatial direction? The naive approach would use directional antennas, a type of antenna that receives signals only from a cone in space. Unfortunately, directional antennas are bulky and have low spatial resolution [125] (about $60°$), making them ill-suited for small agile robots. To address this problem, we employ a new form Synthetic Aperture Radar (SAR), a technique that leverages movement to emulate a high-resolution directional antenna [50]. In order to achieve this, we derive a method for implementing SAR using off-the-shelf wireless cards using single-antenna devices that compacy robots can carry, a challenging task since these devices are not intended for this purpose.

Chapter 9 details our algorithm to enable a robotic receiver to find the profile of signal strength across spatial directions for each sender of interest. It describes a new form of synthetic aperture radar (SAR) using standard Wi-Fi packets exchanged between two independent sintgle-antenna nodes. It then details an optimization that leverages this directional signal profile to position robotic routers to satisfy heterogeneous communication demands of a network of robotic clients, while adapting to real-time environmental changes. Finally, it implements and experimentally evaluates our design to demonstrate gains in comparison to state-of-the-art approaches: the disk-model and stochastic sampling.

Relation to prior work. Prior work employs two broad strategies to improve communication quality in multi-robot networks: On the one hand, there is the Euclidean disk model which assumes that the signal quality of a link is a function of distance between the communicating vehicles. This model is deterministic and simple, and hence when incorporated in a robotic controller, yields simple positional optimizations for a wide range of collaborative tasks [33, 129, 80]. Unfortunately, the Euclidean model is too simplistic and fails to represent wireless signals in realistic environments [113]. On the other hand, there are stochastic sampling methods [191, 113, 47] that measure the wireless signal strength in a robot's vicinity to fit parameters for intricate probabilistic communication models. While such methods are not oblivious to wireless channels, they require exploratory sampling [101] along directions that may be counter-productive to the overall coordination goal. We introduce a system that combines the best of both methods: Like the disk model,

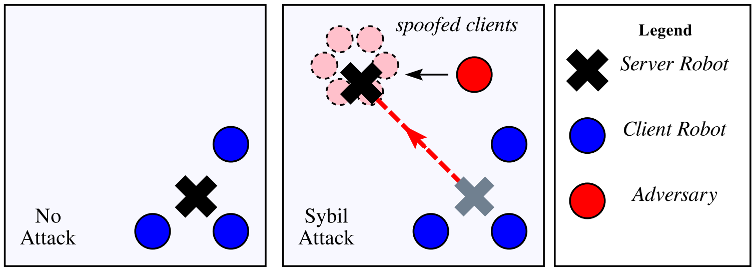


Figure 8-2: Sybil Attack on Coverage. A server robot provides locational coverage to legitimate clients when no attack is present. In a Sybil attack, an adversary spoofs many fake clients to draw away coverage from the legitimate clients.

we do not require prior knowledge of the environment and its obstacles, or a model of channel's distribution. Like the stochastic methods, our approach accurately captures actual signal characteristics and hence can help multi-robot systems satisfy their desired communication demands in real-world environments.

■ 8.2 Guaranteeing Spoof-Resilient Multi-Robot Networks

Multi-robot networks rely on wireless communication to enable a wide range of tasks and applications: coverage [132, 33, 148], disaster management [110], surveillance [18], and consensus [130] to name a few. For these multi-robot systems to perform their tasks optimally, transmitted data is often assumed to be accurate and trustworthy; an assumption that is easy to break. A particularly challenging attack on this assumption is the so-called "Sybil attack", where a malicious agent generates (or spoofs) a large number of false identities to gain a disproportionate influence in the network. These attacks are notoriously easy to implement [152] and can be detrimental to multi-robot networks. An example of this is coverage, where an adversarial client can spoof a cluster of clients in its vicinity in order to create a high local demand, in turn denying service to legitimate clients (Figure 8-2). Indeed, traditional network security approaches such as key passing or cryptographic authentication are difficult to maintain due to the highly dynamic and distributed nature of multi-robot teams where clients often enter and exit the network, leaving them largely vulnerable to attack [72, 146].

Chapter 10 addresses the challenge of guarding against Sybil attacks in multi-robot networks. Our core contribution is a novel algorithm that analyzes the received wireless signals to detect the presence of spoofed clients spawned by adversaries. We call this a "virtual spoofer sensor" as we do not use specialized hardware nor encrypted key exchange, but rather a commercial Wi-Fi card and software to implement our solution. These sensors can defend against the Sybil attack, without requiring expensive cryptographic key-distribution.

Key Idea. At a high level, our virtual sensor leverages the rich physical information already present in wireless signals. At a high level, as wireless signals propagate, they interact with the environment via scattering and absorption from objects along the traversed paths. Carefully processed, these signals can provide a unique signature or "spatial fingerprint" for each client, measuring the power of the signal received along each spatial direction (Fig. 10-2). Unlike message contents such as reported IDs or locations which adversaries can manipulate, spatial fingerprints rely on physical signal interactions that cannot be exactly predicted [60, 113]. We show how these derived fingerprints can be readily integrated into a wide variety of multi-robot controllers. Further, we derive analytical bounds that provably limit the ill-effects of spoofers on the performance of these controllers. In particular, we demonstrates this capability in the context of the well-known locational coverage algorithm [33, 148].

Chapter 10 describes our virtual sensor for spoofing detection which provides performance guarantees in the presence of Sybil attacks and is applicable to a broad class of problems in distributed robotics. It shows that the influence of spoofers is analytically bounded under our system in a coverage context, where each robotic node providing coverage remains within a bounded radius of its position in the absence of an attack. Our theoretical results are validated extensively through experiments in diverse settings.

Relation to prior work. Although a vast body of literature is dedicated to cybersecurity in general multi-node networks (e.g., a wired LAN, see Table 7 in [182])) and wireless networks [81, 194, 189, 192] the same is not true for multi-robot networks [72, 146], leaving them largely vulnerable to attack. This is because many characteristics unique to robotic networks make security more challenging; for example, traditional key passing or cryptographic authentication is difficult to maintain due to the highly dynamic and distributed nature of multi-robot teams where clients often enter and exit the network. Unlike past work, our solution has three attributes that particularly suit multi-robot networks. (1) It captures physical properties of wireless signals and therefore does not require distributed key management. (2) It relies on cheap commodity Wi-Fi radios, unlike hardware-based solutions [189, 194]. (3) It is robust to client mobility and power-scaling attacks.

Chapter 9
Adaptive Multi-Robot Communication Using Signal Directionality

There are many projects on today's frontier that are pushing the capabilities of multi-agent systems. Swarm robotic systems perform many complex tasks through coordination, such as cooperative search of an environment, consensus, rendezvous, and formation control [33, 80, 129]. Google's *Project Loon* [92], Facebook's *Connectivity Lab* [91], and other similar projects [30, 52, 120] envision using a network of controllable routers to provide wireless communication infrastructure in remote areas of the world. At their core, these systems rely on coordination between agents [34, 160, 121, 165], making reliable communication of primary importance. Beyond simply maintaining connectivity, reliable communication may mean supporting heterogeneous and possibly time-varying communication rates amongst different pairs of agents. For example, some agents may need to use the network for transmitting video while others may simply wish to transmit status information.

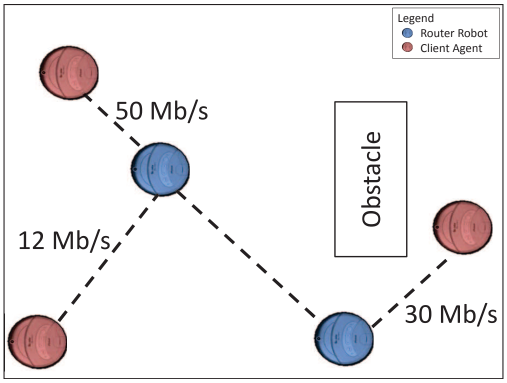


Figure 9-1: Multi-robot Network. Picture of a network of two robot routers satisfying the demands of three clients in an environment with an occluding obstacle whose position is unknown.

We focus on problems where an auxiliary team of robot routers can be deployed to establish reliable wireless communication to a team of client agents who are performing an independent task. As depicted in Figure 9-1, we wish to control the positions of robot routers to establish communication links that are capable of supporting variable demanded rates to the client agents. The problem of providing wireless communication coverage amongst multi-robot systems requires tight feedback between spatial positioning of the robots and sensing of the communication quality. The richer the information on signal quality, the more effective the control. A key realization makes this problem very challenging: Robotic tasks leverage mobility in Euclidean space and thus require knowledge of how position effects communication. However, the relationship of signal quality with spatial position is notoriously hard to predict due to complex interactions with the environment such as multipath, where the signal is reflected and/or attenuated by multiple objects in the environment before arriving at a receiver [113, 60, 101]. Past literature employs two broad strategies to address this challenge. On the one hand, there is the Euclidean disk model which assumes that the signal quality of a link is a function of distance between the communicating vehicles. This model is deterministic and simple, and hence when incorporated in a robotic controller, yields simple positional optimizations for a wide range of collaborative tasks [33, 129, 80]. Unfortunately, the Euclidean model is too simplistic and fails to represent wireless signals in realistic environments [113]. On the other hand, there are stochastic sampling methods [191, 113, 47] that measure the wireless signal strength in a robot's vicinity to fit parameters for intricate probabilistic communication models. While such methods are not oblivious to wireless channels, they require exploratory sampling [102] along directions that may be counter-productive to the overall coordination goal. Further, they often assume the knowledge of parameters based on the structure and material composition of the environment.

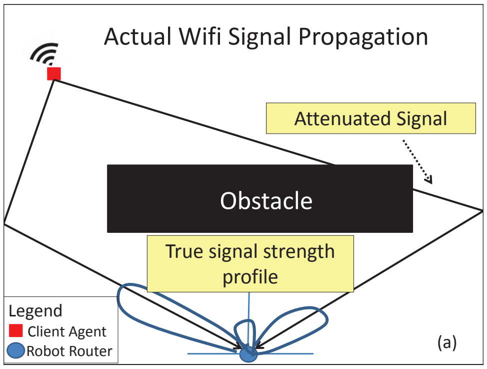


Figure 9-2: Wi-Fi Signal Propagation. Schematic drawing of a true signal strength profile in the local environment of a robotic router. Large lobes indicate directions of high signal strength.

Our objective is to i) present a novel method to capture the spatial variation of wireless signals in the local environment *without* sampling along counter-productive directions, or requiring information about the environment or the channel's distributions and ii) derive a control formulation that maintains the structural (quadratic) simplicity allowed by the Euclidean disk model while accounting for wireless channel feedback.

First, we introduce an innovative approach for mapping communication quality to robot placement. We calculate a mapping between a robot's current position and the signal strength that it receives along *each spatial direction*, for every wireless link with other robots (see Figure 9-2). This is in contrast to existing methods [191, 47], which compute an aggregate signal power at each position but cannot distinguish the amount of signal power received from each spatial direction. Our approach combines the best attributes of both the Euclidean disk model and the stochastic sampling methods: Like the disk model, we can compute our mapping without knowledge of the environment and its obstacles, or a model of the channel's distribution. Like the stochastic methods, our approach uses feedback from the actual wireless signals and hence can help multi-robot systems satisfy their desired communication demands in a real-world implementation. A naive approach to achieve this would be to mount directional antennas atop the routers; but these antennas are bulky and prohibitive for small agile platforms [125]. Instead, we present a novel algorithm based on Synthetic Aperture Radar (SAR) [50], where a single omnidirectional antenna emulates a high-resolution directional antenna. This chapter presents the first such algorithm for implementing SAR using off-the-shelf wireless cards in a non-radar setting, a challenging task since these devices are not intended for this purpose.

Second, we construct an optimization for positioning a team of robot routers to provide communication coverage over client vehicles using the directional information provided by our mapping. Being able to measure the profile of signal strength across spatial directions in real-time yields a much more capable controller. For example, the direction that improves signal strength the most is immediately attainable from these profiles (see Figure 9-2 for a schematic interpretation). Therefore as a direct consequence, the controller has access to the gradient of communication quality for each of its "links", or neighbors to which it communicates. While in the favorable scenario, there is a single recommended direction of movement, in real-world implementations it is possible for there to be multiple such directions due to multipath or even noise that may be affecting the wireless link. This information is important for gauging the confidence with which the controller can improve signal quality by navigating the robot along any of the recommended directions. To this end, we present a method for computing a confidence metric from the data and show that this metric can accurately and automatically identify the three scenarios of strong single-peak, multi-peak, or noisy peak in actual experimental data. Our control algorithm leverages the gradient directions and their associated confidences to automatically tune the speed of the robot, improving both stability and convergence time. Finally, our controller optimizes communication with multiple robots by choosing a direction of movement corresponding to a strong signal that strikes trade-offs between competing demands.

The result of a tight integration between our wireless signal quality mapping and positional controller yields algorithms for router placements that do not rely on environment-dependent parameters, obstacle maps, or even client positions. The overall solution presented is adaptive to variable communication quality demands by the clients, as well as changes in the wireless channels due to natural fluctuations or a dynamic environment.

We implement our method in a multi-robot testbed that has two robotic routers serving three robotic clients. We conduct our experiments in different indoor environments without providing the robotic controller the environment map or the clients' positions. We observe the following: 1) Our system consistently positions the robotic routers to satisfy the robotic client demands, while adapting to changes in the environment and fluctuations in the wireless channels; 2) Compared to the disk model [33, 80] and the stochastic approach [95, 159] under identical settings, our system converges to accurately satisfy the communication demands, unlike the disk model, while significantly out-performing the stochastic method in terms of empirical convergence rate (see Fig. 9-14 in Sec. 9.4.5).

Contributions: We present a method to enable a robotic receiver to find the profile of signal strength across spatial directions for each sender of interest. To this end, we perform synthetic aperture radar (SAR) techniques using standard Wi-Fi packets exchanged between two independent nodes with single omni-directional antennas. We derive a quantitative metric, the confidence, that can accurately and automatically identify the presence of multipath or noise for each communication link. This provides valuable information to the controller in gauging the effectiveness of each recommended direction of movement in improving communication quality. We develop an optimization that leverages the directional signal profiles and their confidences, to position robotic routers to satisfy heterogeneous (and possibly variable) communication demands of a network of robotic clients, while adapting to real-time environmental changes. Finally, we provide aggregate empirical data to show that our method outperforms existing Euclidean disk or Stochastic sampling methods both in convergence time (3.4× faster) and variability of performance (4× smaller variance).

■ Organization

Section 9.1 presents a formulation of the router placement problem for achieving communication coverage for clients with heterogeneous demands. The following sections describe each component of our solution to the problem:

- Section 9.2 derives a new method for measuring rich directional information from wireless channel feedback.

- Section 9.3 presents an algorithm for finding a configuration of routers that *balances* the network, i.e. maximizes the signal quality of the weakest link for a fair network.
- In Section 9.3.2 we derive a confidence metric using channel feedback, that captures the effects of multipath and noise.

Finally, Section 9.4 experimentally evaluates our approach against the disk model and stochastic sampling methods.

9.1 Problem Statement

We consider a mobile network with two classes of members, n robotic clients (or *clients*) whose positions are not controlled, and a team of k robotic routers whose mobility we control. Our goal is to position the robotic routers to provide adaptive wireless communication coverage to the clients, while allowing variable communication quality demands for all clients, and where exact client positions are unknown. For each client $j \in [n] = \{1, \dots, n\}$, we define demanded communication quality $q_j > 0$, and achieved communication quality ϱ_{ij} to each router i (where $i \in [k]$), both expressed in terms of *Effective Signal to Noise Ratio* (ESNR) that has a direct mapping to rate in Mb/s [67].[1] Additionally, let every client j be given an importance $\alpha_j > 0$. We allow all quantities in this section (ie. $q_j, \varrho_{ij}, \alpha_j$) to be time dependent though we omit this dependency henceforth for simplicity.

We define the notion of *service discrepancy* for each pair of robots (i, j) to be the difference between the demanded and achieved communication quality scaled by the importance of the client.

$$w_{ij} = \max(\alpha_j(q_j - \varrho_{ij})/q_j, 0) \tag{9.1}$$

Physically, this is the fraction of the client's communication demand that remains to be satisfied, scaled by α_j. Denote by $c_i \in \mathbb{R}^d$ the position of the ith robot router and by $p_j \in \mathbb{R}^d$ the position of the jth client and $C_t = \{c_{1,t}, \dots, c_{k,t}\}$ is the set of all router positions at time t. In this chapter we give explicit treatment to the case for $d = 2$ although all concepts are extensible to $d = 3$.

9.1.1 Problem Formulation

Given a cost g in terms of signal quality, communication demands, and agent positions, we wish to position each robotic router to minimize the largest discrepancy of service between routers and clients. However, the true form of this function g has an intricate dependence on the position of the client, router, and the environment. Thus an inherent challenge to solving this problem is approximating the influence of spatial positioning on communication quality that generalizes across environments. Our goal is to 1) find $f_{ij} : [-\pi, \pi] \to \mathbb{R}$ (a relation capturing directional information

[1] ESNR is a continuous signal quality measure that has a one-to-one mapping to the maximum data rate supported by a link [67]. We work with ESNR values rather than rates since the latter are discretized (non-continuous).

about the signal quality between i and j), and an approximation $\tilde{g}$ of g, which is a cost, characterizing the anticipated communication quality for the router-client pair (i, j) at a proposed router position c_i, and 2) use this cost to optimize router positions to minimize the service discrepancy to each client. Formally,

Problem 1 *Find i) a mapping $f_{ij} : [-\pi, \pi] \to \mathbb{R}$ that maps spatial direction to wireless signal strength directly from channel measurements, and ii) a cost*

$$\tilde{g}(c_i, C_t, w_{ij}, f_{ij}) > 0 \tag{9.2}$$

that is independent of the environment and satisfies the following properties:

- *Property 1: All link costs $\tilde{g}$ are quadratic.*
- *Property 2: Minimization of a link cost $\tilde{g}$ over c_i directly relates to increasing signal quality for client j and optimization over all link costs $\tilde{g}$ allows trade-offs between clients with competing demands.*
- *Property 3: The link costs $\tilde{g}$ are independent of client positions p_j.*

Given a known number k of routers, client demands q_d, and the mapping f_{ij} for all links in the network, position routers to minimize the worst-case link. Specifically, we aim to find a position for the routers that minimizes the maximum service discrepancy by solving for C in the following problem:

$$C_{t+1} = \arg\min_{c_i \in C} \{\max_j \min_i \tilde{g}(c_i, C_t, w_{ij}, f_{ij})\} \tag{9.3}$$

Intuitively, the solution to this optimization problem favors a "fair" network. Specifically, the solution aims to minimize the "worst service discrepancy" among clients in the network, at any point in time. The worst service discrepancy is given mathematically by the bracketed expression in Equation (9.3) and can be understood intuitively as follows: 1) The service discrepancy of a router-client link captures the difference between the measured quality of the link and the client's demanded communication quality (see Equation (9.1)). 2) Each client is served by a router that offers the minimum service discrepancy to it, at any given time (the innermost $\min$ over i in Equation (9.3) above). 3) The worst service discrepancy, is the maximum service discrepancy among all clients to their chosen routers in the network (the inner $\max$ over j in Equation (9.3)). Notice that the client with the worst service discrepancy may change at any point in time, depending on the configuration of the routers. The optimal configuration of the routers (given by the outer $\arg\min$ term), is therefore the configuration that best satisfies communication demands across the entire network.

■ 9.1.2 Problem Scope

We specify that our aim in this chapter is to position mobile routers to establish a communication network whose links have high enough ESNR to support given client communication demands, q_d. In other words, we are interested in providing the *infrastructure* to support the requested quality of communication. This is in contrast to solving for routing protocols that would optimize the communication traffic over the infrastructure to ensure successful message passing from a sending node to a receiving node. While this is another common metric of connectivity, it is often times treated as a layer on top of an existing communication infrastructure and is an out-of-scope problem with a vast body of dedicated literature (See [49] for an example of routing in robotic networks). Finally, we assume throughout that router-router links are high capacity and that router-client links are the limiting factor that must be optimized.

We dedicate the next sections of this chapter to 1) Developing a method that computes f_{ij} as the profile of signal qualities along each direction θ for each link (i, j) found directly from channel measurements; and 2) Developing an optimization framework that utilizes this directional information to handle trade-offs between competing client demands, and position all routers to jointly minimize the maximum service discrepancy across links in the network.

■ 9.2 Directional Power Profile of a Wireless Link

In this section, we develop the first component of the solution of Problem 1; namely, we derive a method to calculate $f(\theta)$, the mapping to capture the signal strength from a robotic client to its router along each direction θ, where this mapping can be updated often, roughly once every 6cm of motion.[2]

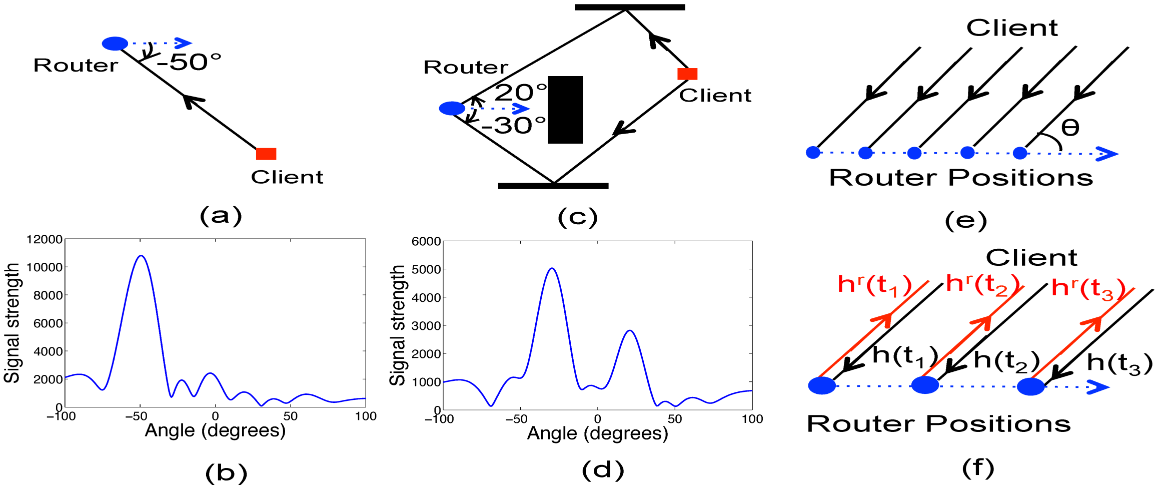


Figure 9-3: Directional Power Profiles. (a)/(c) LOS and NLOS topologies annotated with signal paths. (b)/(d) $f(\theta)$ of the signal in LOS and NLOS. (e) Shows how θ is defined in SAR. (f) Shows $h(t_i)$, the forward channel from transmitter to receiver and $h^r(t_i)$, the reverse channel from receiver to transmitter at time t_i.

Before we explain how we compute $f(\theta)$, we describe this function to help understand what it captures. Assume we have a robotic client and router, where the router moves along some trajectory. We will define the direction θ relative to the tangent to the router's trajectory at each point. Consider the scenario in Fig.9-3(a), where the robotic client is in line-of-sight at $-50°$ relative to the

[2] For simplicity, we denote $f_{ij}(\theta)$ as $f(\theta)$ as we consider only the single link between robotic router i and client j for the rest of this section.

robotic router, which is moving along the horizontal axis. In this case, one would expect $f(\theta)$ to have a single dominant peak at $-50°$, as shown in Fig.9-3(b). Now consider the more complex scenario in Fig.9-3(c), where the environment has some obstacles and one of these obstacles obstructs the line-of-sight path between the router and its client. In this case, $f(\theta)$ would show two dominant peaks at $20°$ and $-30°$ that correspond to the two reflected paths from surrounding obstacles, as shown in Fig.9-3(d).

Advantage over Sampling Methods: One may estimate $f(\theta)$ by sampling the signal power similar to stochastic techniques [191, 95, 159]. In this case, one has to move the router along each direction, compute the power in all these new positions relative to the first, and draw the profile $f(\theta)$. Unfortunately, this approach leads to much wasted exploration. This is because the signal power does not change reliably when the robot moves. For example, if the robot moves for 5 or 10 centimeters, it is very likely that the resulting change in the signal power is below the variability in noise. Hence, measurements of power over short distances are likely to be marred by noise or phenomena that affect the signal strength locally such as deep fades [199] (due to reflections of the signal at the receiver interfering constructively or destructively). To obtain reliable measurements of changes in the signal power, the robot has to move significantly along potentially counter-productive paths.

To address this limitation, our approach relies on the channel phase as opposed to the power. Specifically, at any position the wireless channel can be expressed as a complex number $h(t)$ [141]. The magnitude of this complex channel captures the signal power (more accurately, its square-root). The phase of the channel has traditionally been ignored by robotic systems. However, the phase changes rapidly with motion. For Wi-Fi signals at a frequency of 5 GHz, the phase of the channel rotates by π every 3 cm. This far exceeds any rotation due to noise variability. Thus, by measuring channels as complex numbers and tracking changes in its phase as the robot moves, we reliably estimate signal variation without much exploration. In the next section, we explain how to use a technique called synthetic aperture radar (SAR) to extract the received signal strength along each direction from changes in channel phase. Note that SAR does not need exploration in all directions; the robot can move along its path without extra exploration or sampling. SAR uses the resulting variations in channel phase over distances of a few centimeters to find $f(\theta)$.

■ 9.2.1 Synthetic Aperture Radar (SAR)

Synthetic Aperture Radar (SAR) enables a single antenna mounted on a mobile device to estimate the strength of the signal received along every spatial direction. As explained in Section 9.7.2, SAR employs a single moving antenna to emulate a multi-antenna array and compute the directional profile of signal strength $f(\theta)$ (See Figure 9-4). Therefore, we can leverage the natural motion of a robotic router to implement SAR and measure $f(\theta)$ for each of its robotic clients using a single omni-directional antenna. To do so, the robotic router measures the channel $h(t)$ from its client as it moves along any straight line. The straight line path over which the router acquires data is on the order of half a wavelength (centimeters); assuming the source is stationary and the router either moves at a known constant velocity or its position is known for the traversal time window, then a sufficient amount of usable channel data can be collected. This means every few centimeters the router can have an updated measurement of $f(\theta)$, for all values of θ.

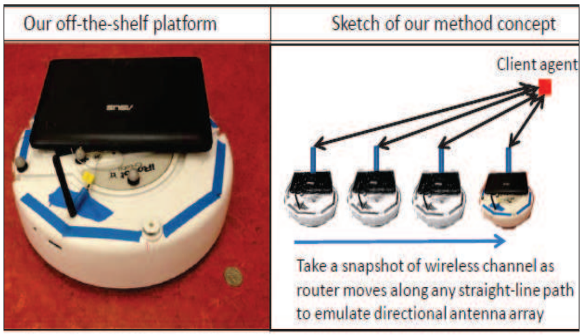

Our off-the-shelf platform
Sketch of our method concept
Client agent
Take a snapshot of wireless channel as router moves along any straight-line path to emulate directional antenna array

Figure 9-4: Emulating a Directional Array. Schematic representation of our method for emulating a directional antenna array with a single omni-directional antenna attached to a mobile off-the-shelf platform.

Specifically, Let $h(t)$ for $t \in \{t_0, \ldots, t_m\}$ be the $m+1$ most recent channel measurements, corresponding to the robot whose displacement from its initial position is $d(t_0), d(t_1), \ldots d(t_m)$. SAR computes the received signal strength across spatial directions $f(\theta)$ as:

$$f(\theta) = \left| \sum_t h(t) e^{-j\frac{2\pi}{\lambda} d(t) cos\theta} \right|^2, \tag{9.4}$$

where λ is the wavelength of the Wi-Fi signal. The analysis of this standard SAR equation may be found in [162]. At a high level, the terms $e^{-j\frac{2\pi}{\lambda} d(t) cos\theta}$ in Eqn. 9.4 project the channels $h(t)$ along the direction of interest θ by compensating for incremental phase rotations introduced by the robot's movement to any path of the signal arriving along θ.

Note that SAR finds the signal power from every angle θ simply by measuring the channels, without any prior tuning to the given direction. Of course, the resolution at which θ is available depends on the number of channel measurements. In fact, moving by around a wavelength (about 6 cm) is sufficient to measure the full profile of $f(\theta)$.

Therefore, SAR is a natural choice for autonomous robotic networks since it exploits the mobility of the robots to compute $f(\theta)$. Further, it only requires the robot to move along a small straight line along any arbitrary direction, and does not require it to explore directions counter-productive to the overall coordination goal. Note that SAR requires only the relative position of the robotic router $d(t)$ and both the magnitude and phase of the channel $h(t)$. It does not require the topology of the environment nor the exact location of the transmitter.

9.2.2 Algorithm for Performing SAR on Independent Wireless Devices

A key challenge in adapting SAR to multi-robot systems is that all past SAR-based solutions [50, 179, 9] are for radar-like applications, where a single device transmits a radar signal and receives its reflections off an imaged object, e.g., an airplane. However, in our scenario the transmitter and receiver are completely independent wireless devices (i.e., the robotic client and router, respectively). This means that the transmitter robot and the receiver robot have different frequency oscillators. In practice, there is always a small difference between the frequency of two independent oscillators. Unfortunately, even a small offset Δ_f in the frequency of the oscillators introduces a time varying phase to the wireless channel.

For instance, let $h(t_0)$, $h(t_1)$, ..., $h(t_m)$ be the actual wireless channel from the robotic client to the robotic router at times $t_0, t_1, \dots, t_m$. The channel observed by the router from its client $\hat{h}(t_0)$, $\hat{h}(t_1)$, ..., $\hat{h}(t_m)$ are given by:

$$\hat{h}(t_0) = h(t_0), \quad \hat{h}(t_1) = h(t_1)e^{-2\pi\Delta_f(t_1-t_0)}, \dots, \\ \hat{h}(t_m) = h(t_m)e^{-2\pi\Delta_f(t_m-t_0)}. \tag{9.5}$$

Hence, the phase of the channels are corrupted by time-varying values due to the frequency offset between the transmitter and the receiver. Fortunately, we can correct for this offset using the well-known concept of channel reciprocity [141]. Specifically, let $h^r(t)$ denote the reverse channel from the robotic router to its client, as shown in Fig. 9-3(f). Reciprocity states that the ratio of the forward and reverse channels stays constant over time, subject to frequency offset, i.e. $h^r(t) = \gamma h(t)$, where γ is constant. Further, the frequency offset in the reverse direction Δ_f^r is negative of the offset in the forward direction, i.e. $\Delta_f^r = -\Delta_f$. Thus, the observed reverse channels $\hat{h}^r(t_0)$, $\hat{h}^r(t_1)$, ..., $\hat{h}^r(t_m)$ are given by:

$$\hat{h}^r(t_0) = h^r(t_0), \;\; \hat{h}^r(t_1) = h^r(t_1)e^{2\pi\Delta_f(t_1-t_0)}, \;\; \dots, \\ \hat{h}^r(t_m) = h^r(t_m)e^{2\pi\Delta_f(t_m-t_0)}. \tag{9.6}$$

Multiplying Eqn. 9.5 and 9.6 and using $h^r(t) = \gamma h(t)$, we have $\hat{h}(t)\hat{h}^r(t) = h(t)h^r(t) = \gamma h(t)^2 \Rightarrow h(t) = \sqrt{\hat{h}(t)\hat{h}^r(t)/\gamma}$. Hence we re-write Eqn. 9.4 as:

$$f(\theta) = \left| \sum_t \sqrt{\hat{h}(t)\hat{h}^r(t)} e^{-j\frac{2\pi}{\lambda}d(t)cos\theta} \right|^2, \tag{9.7}$$

where the constant scaling γ is dropped for simplicity. Hence, to measure $f(\theta)$, the router and client simply need to measure their channels at both ends. In practice, the router and client transmit back-to-back packets with a small gap $\delta \approx 200\mu s$ to obtain $\hat{h}^r(t+\delta)$ and $\hat{h}(t)$, respectively. The router collects these values and approximates $\hat{h}(t)\hat{h}^r(t)$ as $\hat{h}(t)\hat{h}^r(t+\delta)e^{-j2\Delta_f\delta}$. The router computes this 10 times per second (an overhead of just 0.1%). Algorithm 4 summarizes our above approach to compute the signal strength profile $f_ij(\theta)$ for a general wireless link (i,j).

We note the following important points about Algorithm 4: 1) It requires as input the relative displacement of the robot router $d(t)$ from its initial position at $t=0$. In particular, if the robot moves at a known constant velocity v for the duration of SAR (i.e., corresponding to a total displacement of few cm), the algorithm only requires this velocity v, since it can readily compute the relative displacements as: $d(t)=vt$. 2) While the algorithm requires the client to be static, this requirement is only necessary for the duration that the router performs SAR (i.e., corresponding to a total router displacement of few cm). We note that i) the assumption of static channels is also necessary for stochastic sampling based methods since the channels and (thus sampled signal strengths) change otherwise and must be re-sampled and ii) the time scales are largely different between our proposed method and existing sampling methods; specifically, because our method allows for the attainment of rich channel data after a comparatively short measurement period, changes in the environment can be quickly adapted to.

In the following section, we explain how we leverage the signal strength profiles $f_{ij}(\theta)$ on each link (i,j) output by Algorithm 4 to control the position of multiple robotic routers to meet the clients' communication demands.

4 Algorithm for finding directional signal strength profile for a wireless link (i,j)

```
▷ Input: Wireless Channels on the forward link $h_{ij}(t)$, and reverse link $h^r_{ji}(t)$ and robotic router's
    displacement from its initial position $d_i(t)$ at times $t = t_0, \dots, t_m$ on link $(i,j)$
▷ Output: A vector of directional signal strength values $f_{ij} \in \mathbb{R}^l$ for $l$ discrete directionality angles
    in $[0,\pi]$
for $t \in \{t_0, \dots, t_m\}$ do
    $\tilde{h}_{ij}(t) \leftarrow \sqrt{h_{ij}(t)h^r_{ji}(t)}$
end for
for $\theta \in \{0, \frac{\pi}{l-1}, \frac{2\pi}{l-1}, \dots, \pi\}$ do
    $f_{ij}(\theta) \leftarrow \left|\sum_t \tilde{h}_{ij}(t)e^{-j\frac{2\pi}{\lambda}d_i(t)cos\theta}\right|^2$
end for
```

■ 9.3 Communication Coverage Controller

In this section, we target the problem of placing a team of mobile router vehicles at locations such that they provide wireless coverage to client vehicles, each with different communication demands. Specifically, using as input the channel feedback $f_{ij}(\theta)$ derived in the previous section, we aim to find a function $\tilde{g}$ that can be optimized over router positions such that:

$$C_{t+1} = \arg\min_{C}\{\max_{j}\min_{c_i \in C}\tilde{g}(c_i, C_t, w_{ij}, f_{ij})\}. \tag{9.8}$$

Where C_t are current router positions and w_{ij} are the current service discrepancies.

Our focus in this section is to find communication link costs $\tilde{g}$ that have the three desirable properties 1, 1, 1 from Section 10.1.

We show how to capitalize the rich spatial information provided by $f_{ij}(\theta)$, to derive a cost $\tilde{g}$ for each link possessing these three desired qualities. The resulting cost can then be optimized to complete our objective of robot router placement that best satisfies the communication demands of the clients.

■ 9.3.1 A Generalized Distance Metric

We turn attention to the derivation of a quadratic cost whose minimization will improve signal strength. We derive a generalized distance that encodes the direction of steepest descent and the confidence around this direction. We begin with the case where all positions are known and extend to the position independent case in Section 9.3.5.

Consider a single router-client pair (i, j) located at positions (c_i, p_j). A Euclidean disk model approach similarly assigns distance, in the Euclidean sense, to be the cost of each communication link in the network. However, this disk model approach does not use $f_{ij}(\theta)$ at all. Instead, it relates improving communication quality between the router and client to reducing the Euclidean distance between them, i.e. edges in the network take the cost $\tilde{g} := dist(p_j, c_i)$. The appeal of such a cost is in its simple quadratic form that can be easily optimized. Unfortunately, the cost is oblivious to the actual wireless channel at the client and fails to capture the current service discrepancy which can be large even at small distances (say, due to obstacles).

Our system avoids this pitfall, while retaining simplicity, by incorporating real-time channel feedback into a generalized distance metric. Intuitively, we employ a distance metric that effectively "warps" space so that the shortest distance for enabling better communication between two robots is not the straight line path between them, but rather the path along the θ_{max}, the direction of maximum signal strength from the mapping $f_{ij}(\theta)$. The advantage of using this distance metric as compared to a Euclidean distance metric becomes clear when an attenuating obstacle blocks the straight line communication path as shown in Figure 9-5.

Importantly, the recommended heading direction $\vec{v}_{\theta_{\text{max}}}$ may exhibit variation due to noise or multipath on the wireless link. To account for these effects, while not over-fitting to noise, we leverage the entire f_{ij} signal profile to design a *confidence* metric σ_{ij} in the recommended heading direction. The exact form of the confidence metric is derived in the following section. The purpose of this confidence metric is to incorporate second-order information from f_{ij} that captures the pres-

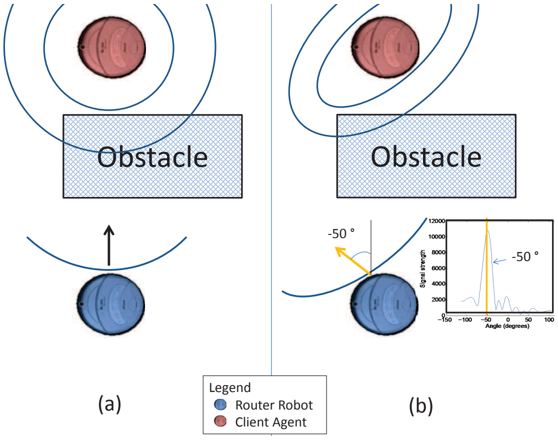


Figure 9-5: Euclidean vs. Mahalanobis Distance. Schematic depiction of the use of channel feedback for assigning cost to communication links in the network where edge cost is shown as circular contour lines. On the left, a Euclidean distance metric assigned lowest cost to the straight-line direction, whereas using the Mahalanobis distance (right) skews the distance contours to identify the direction about $\vec{v}_{\theta_{\max}}$ as the lowest cost. The amount of skew in the contour lines is determined by the confidence metric derived in Section 9.3.2.

ence of noise, or multipath, and can be used to alter the behavior of the controller accordingly (see Section 9.3.2). By using a *Mahalanobis* distance metric for assigning costs to each communication edge in the network, we can encode both the recommended heading direction and its confidence. The mathematical definition of the Mahalanobis distance is:

Mahalanobis Distance Given a positive definite matrix $M \in \mathbb{R}^{d \times d}$, a vector $x \in \mathbb{R}^d$, and a vector $y \in \mathbb{R}^d$, the Mahalanobis Distance between x and y is:

$$\mathrm{dist}_M(x, y) = \sqrt{(x-y)^T M (x-y)} \tag{9.9}$$

Euclidean distance is a special case of the Mahalanobis distance (see Fig. 9-8(a)) with $M = I$ where I is the identity matrix of appropriate dimension.

Here, $M = Q\Lambda Q^T$ is a positive-definite matrix, where Q consists of orthogonal eigen-vectors and Λ contains their corresponding eigen-values. By a careful construction of the matrix M, we can encode channel feedback as a quadratic Mahalanobis distance cost for each communication link in the network. This construction requires both the recommended descent direction $\vec{v}_{\theta_{\max}}$ from $f_{ij}(\theta)$, and the confidence metric σ_{ij} that is also computed from $f_{ij}(\theta)$ in the following section.

■ 9.3.2 Confidence Metric from Channel Feedback

We design a parameter σ_{ij} that is derived from the mapping $f_{ij}(\theta)$ and that we refer to as a *confidence* in the recommended heading direction $\vec{v}_{\theta_{\max}}$. Intuitively, σ_{ij} captures the"variance" of $f_{ij}(\theta)$

around θ_{max}. We define σ_{ij} mathematically as the ratio of two quantities, $\sigma_{f\,ij}$ and $\sigma_{N\,ij}$. We define

$$F = \sum_{\theta \in \{-90,\ldots,90\}} f(\theta) \tag{9.10}$$

$$\sigma_{f\,ij} = \sum_{\theta \in \{-90,\ldots,90\}} (\theta - \theta_{\text{max}})^2 \frac{f(\theta)}{F} \tag{9.11}$$

$$\sigma_{N\,ij} = \sum_{\theta \in \{-90,\ldots,90\}} (\theta - \theta_{\text{max}})^2 \frac{F}{L} \tag{9.12}$$

$$\sigma_{ij} = \frac{\sigma_{f\,ij}}{\sigma_{N\,ij}} \tag{9.13}$$

where L is the total number of θ values that make up the plot $f_{ij}(\theta)$. The term $\sigma_{f\,ij}$ is the variance of the plot f_{ij} around its maximum $\theta = \theta_{\text{max}}$ and $\sigma_{N\,ij}$ is a normalization factor (it is the variance around θ_{max} in the case that the mass under the $f_{ij}(\theta)$ curve was distributed evenly over the θ values). The ratio of these two quantities, $\sigma_{f\,ij}/\sigma_{N\,ij}$, characterizes the amount signal strength (mass under the $f_{ij}(\theta)$ curve) that is concentrated under the peak direction θ_{max} versus the remaining parts of the curve. A ratio of $\sigma_{f\,ij}/\sigma_{N\,ij} = 1$ would mean that the $f_{ij}(\theta)$ plot does not provide evidence that the max direction θ_{max} is of much significance and that indeed the plot is entirely noise. On the other hand a ratio $\sigma_{f\,ij}/\sigma_{N\,ij} < 1$ indicates that a significant portion of the signal strength curve in $f_{ij}(\theta)$ is concentrated around the max θ_{max} and thus this peak is considered to have "high confidence." Lastly, the case where $\sigma_{f\,ij}/\sigma_{N\,ij} > 1$ indicates the presence of high signal strength in other parts of the $f_{ij}(\theta)$ curve other than the θ_{max} direction which suggests the presence of multipath. These three scenarios are demonstrated empirically in Figure 9-6 where three actual $f_{ij}(\theta)$ plots are automatically identified as being single peak "high confidence", multiple peak "noise", and multiple peak "multipath" scenarios respectively, by computing the ratio σ_{ij} for each plot. The figure demonstrates a graphic depiction of this ratio where areas of the $f_{ij}(\theta)$ plot above and below the uniform variance line determine the confidence value (compare with Equations in (9.10)).

We define these three cases below for reference:

Confidence Confidence in the direction of highest signal strength θ_{max}. We define three cases captured by our confidence metric $\sigma_{ij} = \frac{\sigma_{f\,ij}}{\sigma_{N\,ij}}$:

- **High confidence peak:** $\sigma_{ij} < 1$
- **Noise:** $\sigma_{ij} \approx 1$
- **Multipath:** $\sigma_{ij} > 1$

See Figure 9-6 for examples of these regions identified automatically from actual experimental data.

Experimental results in the basement of the Stata Center building on the Massachusetts Institute of Technology campus show that the regions of high confidence, noise, and multipath defined above can be identified automatically from data using the confidence metric from Eq. (9.13) (see

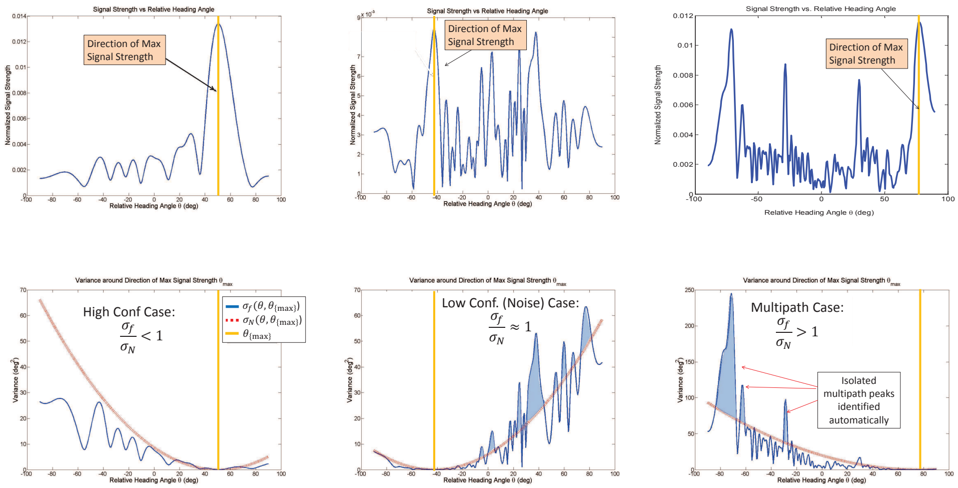


Figure 9-6: Directional Power Profiles. These plots show directional signal strength profiles from actual experiments. They demonstrate how the confidence metric identifies cases of high confidence, low confidence, and multipath automatically from the f_{ij} signal strength profile. The dotted red line is the variance,$\sigma_{N\,ij}$, of a uniform signal strength profile $f_{N\,ij}(\theta) = 1/L$ centered around $\theta_{\max}$. Comparing the variance (bottom row) $\sigma_{f\,ij}$ to $\sigma_{N\,ij}$ indicates which of the three cases are occurring in the f_{ij} plot (top row): high confidence $\theta_{\max}$ ($\sigma_{f\,ij} < \sigma_{N\,ij}$), low confidence (noise) $\theta_{\max}$ ($\sigma_{f\,ij} \approx \sigma_{N\,ij}$), or multipath around $\theta_{\max}$ ($\sigma_{f\,ij} > \sigma_{N\,ij}$).

Figure 9-7a). As expected, areas of the environment with no significant occlusions to the client agent show strong evidence of high confidence profiles. Areas such as corridors with potential occlusions due to walls and corners show a much higher incidence of multipath, about 90% in the worst case.

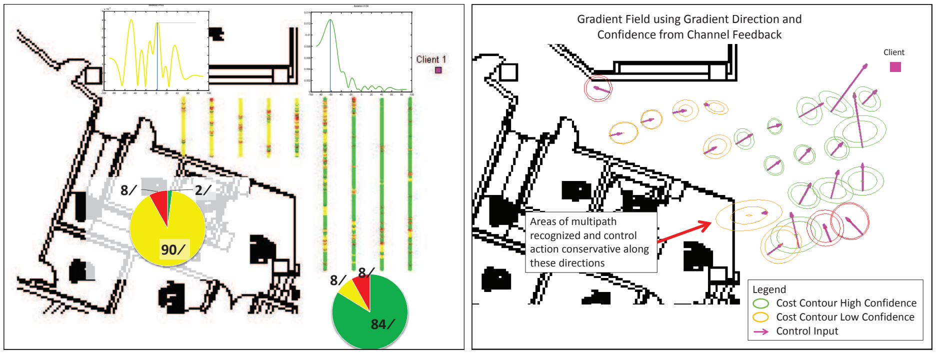
An important observation from the data in Figure 9-7a is that even in line-of-sight regions of the environment (relative to the position of the client) there may be significant multipath present due to reflections from nearby concrete walls and this may cause the direction profile to have peaks in heading directions that are non-intuitive. Therefore this data suggests that metrics relying solely on the geometry of the environment, including visibility graphs, do not adequately capture the complexities of wireless signal quality in general environments.

■ 9.3.3 Construction of Communication Link Costs

Our objective here is to construct the Mahalanobis distance matrix M_{ij} for each communication link (or edge) in the network using $f_{ij}(\theta)$. Specifically,

Problem 2 (Computation of M_{ij}) *For each communication link (i, j) in the network where $i \in [k]$ and $j \in [n]$, find a heading direction $\vec{v}_{\theta_{max}}$, a confidence metric σ, and a construction of M_{ij} such that setting the edge costs $\tilde{g}$ from Equation* (9.8) *to $\tilde{g} := dist^2_{M_{ij}}(p_i, c_j)$ satisfies Properties* 1*-*1*.*

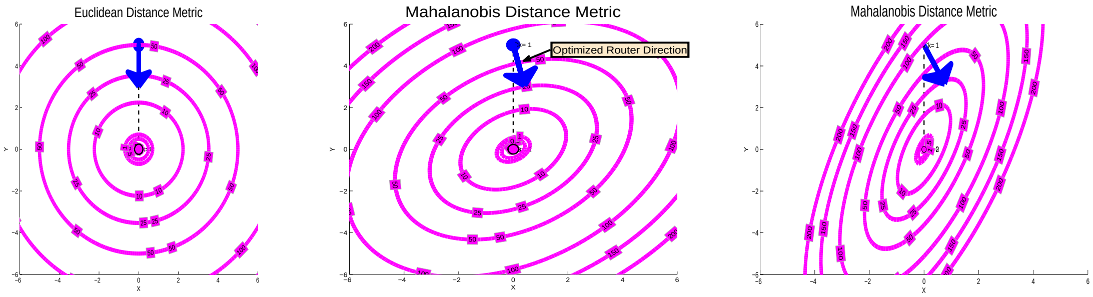
The direction along which the signal strength is maximum, $\theta_{\max}$, is characterized by a peak in the $f_{ij}(\theta)$ plot and we define $\vec{v}_{\theta_{\max}}$ to be the unit vector along this recommended heading direction



(a) Confidence Metric (b) Resulting Control Action

Figure 9-7: Confidence Metric and Control Actions. Figure (a) shows data collected for a one-link system of one router and one client where the client is stationary at the top right corner of a basement environment and a mobile router is driven in a lawn mower pattern throughout the environment through line-of-sight and non-line-of-sight regions. Each colored data point represents an acquired directional signal profile (two example profiles are shown) and the color of the data point is the result of automatic mode detection from the data using the confidence metric from Eq. (9.13) where red=noise, yellow=multipath, and green=high confidence peak. In (b) the resulting edge cost contours (Equation (9.14)) and actual control command at each point in the environment is shown. Confidence values have a direct effect on velocity (as indicated by arrow length) where confident directions are pursued more aggressively.

$\theta_{\max}$. Using this direction alone does not provide enough information for effective position control of the routers however, due to the fact that this direction may experience corruption due to noise or multipath. In the previous section we showed that the presence of multipath or noise in the $f_{ij}(\theta)$ plot can be identified via the computation of a confidence metric σ_{ij}. Now, we encode the quantity σ_{ij} into our controller such that $\vec{v}_{\theta_{\max}}$ directions of high confidence are followed more aggressively (larger displacements along these directions), and the opposite is true of $\vec{v}_{\theta_{\max}}$ directions with low confidence. Figure 9-7b shows the effect of the confidence value on the commanded displacements made by the controller in an actual implementation.



(a) Euclidean Distance (b) Mahalanobis (Low Conf) (c) Mahalanobis (High Conf)

Figure 9-8: Euclidean vs. Mahalanobis Level Sets. These plots show the level sets of a Euclidean distance function and a Mahalanobis distance function.

Specifically, for the three categories of σ_{ij} we desire the following behaviors for the routers: 1) $\sigma_{ij} < 1$: Indicates a high confidence in $\vec{v}_{\theta_{\max}}$ due to a sharp peak in f_{ij}. The robot is moved at higher speeds; 2) $\sigma_{ij} \approx 1$: Indicates that f_{ij} is noisy, so the robot moves slowly; 3) $\sigma_{ij} > 1$: Indicates that f_{ij} has multiple significant peaks owing to multi-path. We study this last case in Sec. 9.3.4., and particularly the opportunity it presents for making trade-offs between clients.

We use the heading direction and confidence to design a cost function $\tilde{g}$ that locally captures the cost of communication in the spatial domain. We express this cost as a Mahalanobis distance. The square of the Mahalanobis distance is a cost function (paraboloid) with ellipsoidal level sets (Fig. 9-8). We design our cost by orienting these level sets so that the direction of steepest descent is along $\vec{v}_{\theta_{\max}}$. We then skew the ellipsoidal level sets using the confidence σ_{ij}, so that a higher confidence translates to a steeper descent which leads to larger router displacements (speed) in the descent directions with high confidence.

Algorithm 5 provides a calculation of the matrix M_{ij} from Problem (2). We simply set one of the eigen-vectors of Q to the heading direction $\vec{v}_{\theta_{\max}}$. To skew the ellipsoid, we set the ratio of the eigen values $\{\lambda_1, \lambda_2\}$ in Λ to the confidence σ_{ij}^2, i.e. $\lambda_2/\lambda_1 = \sigma_{ij}^2$, where λ_1 is the eigen-value corresponding to $\vec{v}_{\theta_{\max}}$. For example, in Fig. 9-8(b), where $\sigma_{ij} \approx 1$ (i.e. poor confidence), the level sets are nearly circular, leading to a shallow descent in cost; while Fig. 9-8(c), where $\sigma_{ij} < 1$ (i.e high confidence), the level sets are skewed, leading to a steep descent in cost along $\vec{v}_{\theta_{\max}}$. In other words, the cost function has an elegant geometric interpretation, akin to Euclidean distance, but is derived directly from channel measurements. Further, the cost function $\tilde{g} := \text{dist}^2_{M_{ij}}(p_i, c_j)$ from Eqn. 9.9 is quadratic, a desirable property for optimizations.

5 Algorithm for constructing M_{ij} from channel feedback.

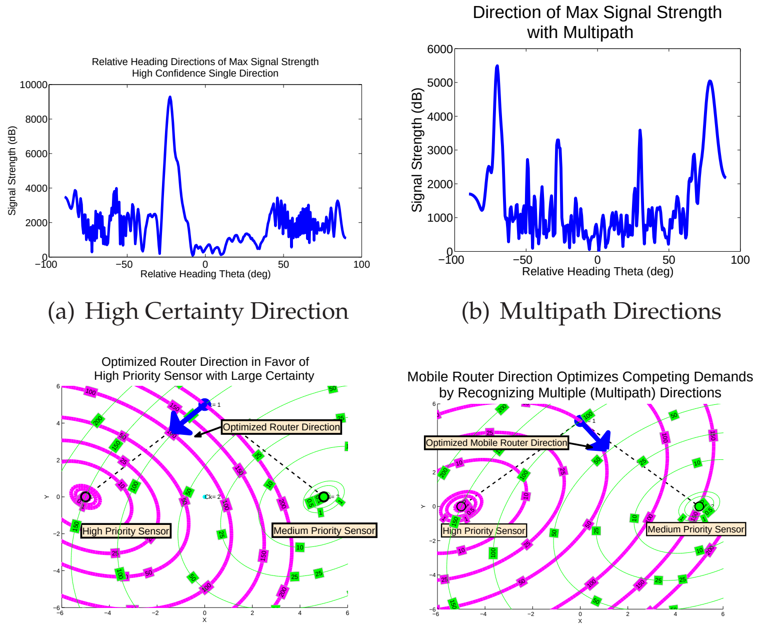
```
▷ Input: Directional signal strength map $f_{ij}$ for every link $(i, j)$ from Algorithm 4
▷ Output: A matrix $M_{ij}$ for defining communication edge costs in Equation (9.8) using Mahalanobis distance from Problem 2.
$Q_{ij} = [\vec{v}_{\theta_{\max}}, \vec{v}_{\theta_{\max\perp}}]$        ▷ a set of orthonormal basis vectors defined using $\vec{v}_{\theta_{\max}}$
$\sigma_{ij} = \frac{\sigma_{f\,ij}}{\sigma_{N\,ij}}$        ▷ confidence in the $\vec{v}_{\theta_{\max}}$ direction
$\Lambda = \text{diag}([\frac{1}{\sigma_{ij}^2}, 1])$        ▷ Construct a diagonal matrix using confidence
$M_{ij} = Q_{ij} \Lambda Q_{ij}^T$
```

9.3.4 Network Trade-offs

In this section, we show how our optimization framework readily extends to a multi-agent scenario and study the different trade-offs. We show that via the setting of two parameters, both set automatically from wireless channel data, the resulting positional controller can be made to greedily optimize one client's needs or alternatively, strike trade-offs between multiple clients. First, we focus on managing service discrepancies specified by w_{ij}. The quantity w_{ij} aims to bias the controller by assigning higher weight to users with larger service discrepancies. To do this, we scale the cost function $\tilde{g} = \text{dist}^2_{M_{ij}}(p_i, c_j)$ by the square of the discrepancy w^2_{ij} to optimize yield the network cost:

$$r_M(P, C) = \max_{p_j \in P} \min_{c_i \in C} \{w^2_{ij} \text{dist}^2_{M_{ij}}(p_i, c_j)\} \tag{9.14}$$

Second, we highlight the subtle role played by the confidence σ_{ij} in managing network trade-offs. For instance, consider a scenario with two clients: 1 and 2, where client-1 demands greater communication quality (as specified by w_{ij}'s). Suppose client-1 has a highly confident $\vec{v}_{\theta_{\max}}$ as shown in Fig. 9-9(a) (i.e $\sigma_{ij} < 1$). As expected, the robotic router is directed towards client-1 as shown in Fig. 9-9(c). In the more interesting scenario in Fig. 9-9(b), client-1's confidence is poor due to multiple peaks in the signal profile f_{ij} (i.e $\sigma_{ij} > 1$). Here, the router strikes a trade-off and services client-2 instead, as this may potentially benefit client-1 as well due to the multipath recognized in client-1's $f_{ij}(\theta)$ map. The intuition behind this is simple. Equation 9.14 above, scales the ellipsoidal cost function based on the discrepancies w_{ij}'s. However, recall that the ellipsoidal cost function is steep (or shallow) depending on whether the confidence is high (or low) and this is attained by setting the ratio of eigenvalues λ_2/λ_1 of M_{ij} (See Line 5 in Algorithm 5). In extremely low confidence scenarios such as Figure 9-9(b), the higher value of discrepancy of client-1 is masked by its low value of confidence. Hence, this balances the trade-off in favor of client-2, despite having a lower discrepancy.



(a) High Certainty Direction

(b) Multipath Directions

(c) Client Favored

(d) Client Tradeoff Multipath

Figure 9-9: Trade-offs between Clients. $(a) - (b)$ show the $f_{ij}(\theta)$ map for the high demand client; $(c) - (d)$ show the optimized router direction

Algorithm 6 demonstrates how the cost in Equation (9.14) can be used to find an updated set of router positions when both client and router positions are known at the current iteration.

The optimization in Equation (8) in Algorithm 6 is equivalent to a k-center optimization problem where the distance metric is a Mahalanobis distance. This is a generalized router placement problem similar to that studied in [57] for Euclidean distances. Thus the returned solution from this algorithm is the optimal placement of routers corresponding to the optimal assignment of routers to clients, given the channel feedback at the current iteration t.

6 Algorithm for router placement with known client positions.

```
▷ Input: Directional signal strength map $f_{ij}(\theta)$ for every link $(i, j)$, demand $q_j$, relative importance $\alpha_j > 0$ for client $j$, current quality of each link $\rho_{ij}$, and current router and client positions $P = \{p_1, \ldots, p_n\}$, $C = \{c_1, \ldots, c_n\}$
▷ Output: A configuration of optimal router positions $C^*$, $|C^*| = k$, given the current channel feedback for all links in the network.
for all links $(i, j)$ in the network with $\rho_{ij} > 0$, $i \in [k]$,$j \in [n]$ do
    $w_{ij} = \max(\alpha_j \frac{q_j - \rho_{ij}}{q_j}, 0)$
    $M_{ij} =$ result of Algorithm 5
end for
Compute:
$C^* = \arg\min_C \{\max_{p_j \in P} \min_{c_i \in C} w_{ij}^2 (p_i - c_j)^T M_{ij} (p_i - c_j)\}$
Return $C^*$
```

9.3.5 A Position-Independent Solution

A simple relaxation to the cost from the previous section frees the optimization of using client positions, while maintaining its simple structure and desirable properties developed above. Consider a user specified step-size $\gamma > 0$, that encodes the maximum permissible displacement for each router and denote $c_{i,t}$ to be the current router position. We replace client positions p_j in Equation (9.14) with "virtual" positions p'_{ij}:

$$p'{}_{ij} = c_{i,t} + \gamma w_{ij} \vec{v}_{\theta_{\max}}. \tag{9.15}$$

Intuitively, a client is no longer directly observed but rather estimated to be along the relative direction $\vec{v}_{\theta_{\max}}$ and at a distance of γw_{ij} with respect to the ith router. As before, $\vec{v}_{\theta_{\max}}$ is the heading direction associated with the maximum strength signal direction $\theta_{\max}$. As a client's demand is better satisfied by router i, the service discrepancy w_{ij} tends to 0 and the client is perceived as being closer to router i. The observation here is that routers better equipped to service a particular client as reflected by the w_{ij} term, will view the client as "closer" and those routers with a weaker signal to the same client will view this client as farther away. This results in a natural method of assigning client nodes to routers by effectively sensing over the wireless channels.

9.3.6 Controller for Router Positioning

We now present an algorithm for achieving router positions that minimize the edge costs $\tilde{g}$ derived in the previous sections. Particularly we formulate $\tilde{g}$ from Equation (9.3) to be

$$\tilde{g}(c_i, C_t, w_{ij}, f_{ij}) = (p'_{ij} - c_i)^T M_{ij}(p'_{ij} - c_i) \tag{9.16}$$

Where the dependence of $\tilde{g}$ on C_t, w_{ij} and f_{ij} are captured indirectly by p'_{ij} and M_{ij} via Equation (9.15) and Algorithm 5 respectively. This choice of edge costs satisfy Properties 1-1. Namely, having a quadratic form, allowing optimization over the entire network with competing demands, and being independent of client positions. As described in Section 10.1, minimization of these edge costs by Equation (9.3) results in the optimization of a network-wide metric, ie. minimizing the worst-case client service discrepancy.

The resulting optimization framework can be shown to exhibit other desirable properties relative to the instantaneous wireless channels over the network. An important remark is that we do not make assumptions on how the wireless channels may change over time, nor do we make assumptions on the underlying signal quality function in areas of the environment that are not currently being sensed by the routers. Unfortunately, this impairs our ability to prove certain desirable controller attributes such as convergence, that would require some additional assumptions on the signal quality such as a guarantee that this function is smooth, and can be strictly improved at every iteration. Such assumptions would be invalidated by small-scale fading alone [101, 60] , in real wireless systems. However, by relying solely on instantaneous channel feedback, we retain the important ability to adapt quickly to changes in the wireless environment due to dynamic obstacles, for example. Based on current channel feedback, we highlight our controller's network-wide properties. The following properties, and convergence to client demanded rates, are demonstrated extensively in actual implementations in the next section of the chapter:

Property 1 *The assignment of routers to clients is optimal based on the current feedback over wireless links in the network.*

This can be seen from the observation that Line 10 from Algorithm 7 is the classic k-center solution [57, 44] under the Mahalanobis distance metric. A k-center solution will assign clients to their closest routers. In this case "closest" is defined in the signal quality sense where routers serve the clients to whom their signal strength is greater than the signal strength between any other router in the network to the same client. An example of this property in an actual hardware implementation can be seen in Section 9.4.3-9.4.4 where routers choose clients based on the strengths of their relative wireless links.

Property 2 *Stability of router positions to solutions that satisfy client demands over the network.*

Our final cost takes the form:

$$r_M(C) = \max_{j\in\{1,\ldots,n\}} \min_{c_i\in C}\{\text{dist}^2_{M_{ij}}(c_{i,t} + \gamma w_{ij}\vec{v}_{\theta_{\max}}, c_i)\} \tag{9.17}$$

By expanding the squared per-link cost $\text{dist}^2_{M_{ij}}(c_i + \gamma w_{ij}\vec{v}_{\theta_{\max}}, c_i)$ from Eqn. 9.14:

$$(c_i - c_{i,t})^T M_{ij}(c_i - c_{i,t}) - 2\gamma w_{ij}\lambda_{\theta_{ij}}\vec{v}^T_{\theta_{\max}}(c_i - c_{i,t}) + \gamma^2 w^2_{ij}\lambda_{\theta ij} \tag{9.18}$$

we note that as $w_{ij} \to 0$ the first term in Eqn. (9.18) favors stable solutions where $c_i = c_{i,t}$, ie. the router reaches a static solution when all of its assigned clients have zero service discrepancy. In the case where it is not possible to satisfy all client demands, for example if there are not enough routers k to provide communication coverage to the clients, Algorithm 7 returns the solution with the lowest service discrepancy that is within a user specified tolerance of optimal. Extensive empirical validation of this property in actual hardware implementations is shown in Sections 9.4.3-9.4.5.

Property 3 *Adaptation of router positions to changes in the wireless channels and/or client demand.*

The Equation (9.18) shows that for nonzero discrepancy, ie. $w_{ij} \neq 0$, the cost for edge (i, j) is also nonzero. Thus a change in the client demands q_j, or in the quality of the wireless link ρ_{ij} due to moving occlusions or changes in the environment, will equally change the weighting w_{ij} on the link (see Equation (9.1)). If the change in link quality ρ_{ij} is sufficient, ie. if $w_{ij} > 0$, the routers following Algorithm 7 will update their positions until a new solution is found (see Property 2). Empirical validation of this property in an actual implementation is demonstrated in Section 9.4.6.

An algorithm for finding router placements in the most general case of unknown client positions is presented as Algorithm 7 below.

■ 9.4 Experimental Results with Motion Capture Support

We evaluated our system on a five-node testbed with two routers and three clients. Each node was an ASUS 1015PX netbook equipped with an Intel 5300 Wi-Fi card mounted on an iRobot Create robot. We implemented SAR by modifying the iwlwifi driver on Ubuntu 10.04. We used the 802.11 CSI tool[67] to obtain channel information ($\hat{h}(t)$ in Eqn. 9.7). The routers communicated with a central laptop emulating the base for control information and human input. Our first set of experiments were performed in a room with a Vicon motion capture system to measure the relative displacement of the robotic routers. Sec. 9.5 describes results in complex indoor environments without motion capture support. Our testbed contains obstacles to simulate both line-of-sight and non-line-of-sight scenarios.

■ 9.4.1 Measuring the Direction of Maximum Signal Strength

We first provide a microbenchmark to demonstrate that our system indeed provides the direction θ_{max} that results in maximum improvement in client service quality. We consider two representa-

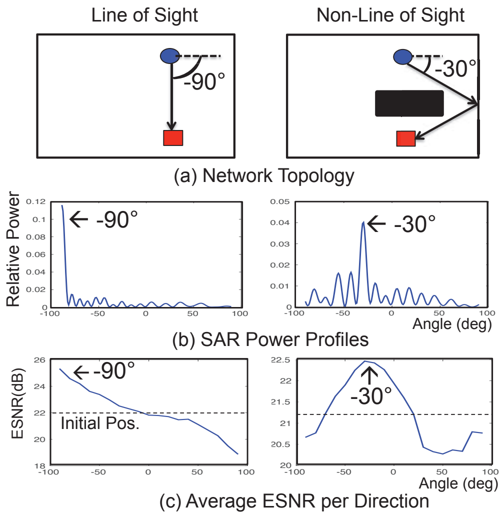
7 Algorithm for router placement with unknown client positions.

```
▷ Input: Directional signal strength map $f_{ij}(\theta)$ for every link $(i,j)$, demand $q_j$, relative importance $\alpha_j > 0$ for client $j$, current quality of each link $\rho_{ij}$, step-size $\gamma > 0$, tolerance $tol > 0$ and current router positions $C = \{c_1, \ldots, c_n\}$
▷ Output: A configuration of router positions $C^*$, $|C^*| = k$, and the achieved service discrepancy $w^*$.
$\delta = \inf$
while $\delta > tol$ do

    for all links $(i,j)$ in the network with $\rho_{ij} > 0$, $i \in [k]$,$j \in [n]$ do
        $p'_{ij} = c_{i,t} + \gamma w_{ij} \vec{v}_{\theta_{\max}}$    ▷ compute virtual client $j$ position as perceived by router $i$
        $w_{ij} = \max(\alpha_j \frac{q_j - \rho_{ij}}{q_j}, 0)$
        $M_{ij}$ = result of Algorithm 5
        Compute:
        $C^* = \arg\min_C \{\max_{p_j \in P} \min_{c_i \in C} (p'_{ij} - c_i)^T M_{ij} (p'_{ij} - c_i)\}$
        $C = C^*$    ▷ Update router positions
        for all links $(i,j)$, $i \in [k]$,$j \in [n]$ do
            $w_{ij} = \max(\alpha_j \frac{q_j - \rho_{ij}}{q_j}, 0)$    ▷ Compute updated $w_{ij}$
        end for
        $\delta = \max_{(i,j)}(w_{ij})$    ▷ Store max
    end for
end while
$w^* = \min_{j \in [n]} \max_{i \in [k]} w_{ij}$
Return $C^*, w^*$
```

tive examples of a single robot router-client pair placed in i) line-of-sight configuration where the strongest signal path is also the shortest Euclidean distance path, and ii) non-line-of-sight configuration where the shortest Euclidean path between the router and client is obstructed by a cement column. These configurations are depicted in Figure 9-10(a). We measure the power profiles of signals from the robot router from different directions using the solution described in Sec. 9.2. We also compute the average service quality of the client (measured in terms of Effective Signal-to-Noise Ratio or ESNR) along various spatial directions by iteratively moving the robot router and exhaustively sampling the signal quality along each physical direction, at a total of 1800 samples (100 samples, about 1m, along each ten degree arc).

Results: Fig. 9-10(b) and (c) plots the power profile obtained by our system, as well as the service quality observed when moving along the different spatial directions. We note that the direction of maximum signal power measured by our system actually leads to maximum increase in service quality in both line-of-sight and non-line-of-sight settings. Notice that while the plots in both Fig. 9-10(b) and (c) capture similar trends, they are not identical. Specifically, the profiles output by our system isolate signal power arriving from individual spatial directions, and therefore have sharp peaks that are easy to discern and less prone to error. In contrast, the average service quality of the client varies much more gradually along different directions, and therefore needs to be sampled much more extensively to obtain accurate trends (for more details, see Sec. 9.2). This



(a) Network Topology

(b) SAR Power Profiles

(c) Average ESNR per Direction

Figure 9-10: Direction of Maximum Signal Strength. We validate our computed direction of signal strength for two representative configurations of (a) line-of-sight and non-line-of-sight settings. (b) Power profiles indicating direction of maximum signal strength. (c) Service quality (average ESNR) measured along each spatial direction.The horizontal dotted line indicates the ESNR at the initial position.

demonstrates that our system captures the direction of maximum signal strength with a higher accuracy, and without the need for exhaustive exploration, when compared to pure sampling-based approach.

■ 9.4.2 Visualizing the Gradient Field of Signal Strength

In this experiment, we visualize the gradient field of the directions of maximum signal strength θ_{max}, on a wireless link. We consider a single client, serviced by a robot router that is: 1) In direct line-of-sight (LOS) as shown in Fig. 9-11(a). 2) In possible non-line-of sight (NLOS) scenarios due to obstacles as shown in Fig. 9-11(b). We drive the robot router in a lawn-mover pattern and get θ_{max} at regular intervals.

Results: Fig. 9-11(a) and 9-11(b) depict the gradient field with the arrows indicating θ_{max} in LOS and NLOS, respectively. The gradient field in LOS accurately directs the robot router towards the client regardless of its initial position. In NLOS, the robot is directed away from obstacles so that controller can route around obstacles to improve signal strength. We stress that θ_{max} is found locally at the router purely via wireless channels and its own position, *without* prior knowledge of the environment. Further, the plots are not static and naturally *change over time*, especially in dynamic settings. Thus our system obtains instantaneous θ_{max} values locally in real-time.

Fig. 9-11(c) and 9-11(d) plot $f_{ij}(\theta)$, the power profile of the signal along different directions, for a candidate location in line-of-sight and non-line-of-sight scenarios, respectively. Clearly, the power profile in line-of-sight is dominated by a single peak at θ_{max}, directed along the line-of-sight path to the client. In contrast, the power profile in non-line-of-sight close to an obstacle has two significant peaks, each corresponding to reflected paths along walls or other objects in the environment.

■ 9.4.3 Controlling Router Trajectory to satisfy Client Demands

We evaluate how a single robotic router finds a trajectory to satisfy the demands of three clients (specified in terms of effective signal-to-noise ratio or ESNR) using θ_{max} on each link. We consider the candidate non-line-of-sight setting in Fig. 9-12(a). The router is unaware of exact client positions or the layout of the environment.

Results: Fig. 9-12(a) depicts the trajectory of the robotic router in blue. The colored arrows denote the recommended $\vec{v}_{\theta_{\text{max}}}$ directions for each client at every control point. The figure shows how the robot performs non-zero control actions until it eventually satisfies network demands. Fig. 9-12(b) tracks the ESNR of the clients across time (dotted lines). The plot shows that the ESNR demands of each client (solid lines) are satisfied upon convergence. Note that the whenever the robot decides to follow the $\vec{v}_{\theta_{\text{max}}}$ of a client at a control point (vertical line), the client's ESNR increases. This validates our claim that following a heading direction based on $\vec{v}_{\theta_{\text{max}}}$ indeed improves the ESNR of the corresponding client.

■ 9.4.4 Aggregate System Results

We evaluate our full system with two robot routers serving three clients with different ESNR demands. We perform the experiment in line-of-sight (LOS) and non-line-of-sight (NLOS) settings as shown in the inset maps of Fig. 9-13(b) and 9-13(d) respectively. We repeat the experiment five times in each setting and plot the results.

Results: Fig. 9-13(a) and 9-13(b) plot the mean and variance of ESNR over time across experiments for each client (dotted colored lines) in LOS and NLOS. Clearly, each client's ESNR demand (solid lines) is satisfied at the converged position across experiments. Fig. 9-13(c) and 9-13(d) plot the corresponding aggregate link rate across time, which follows the same trend as the ESNR [67].[3] The inset plots in Fig. 9-13(c) and 9-13(d) depict the final converged position of the routers (blue dots) in LOS and NLOS. The results show that our system consistently satisfies client demands while adapting to real-time changes in wireless channels, even in the presence of obstacles.

■ 9.4.5 Comparison with Existing Schemes

We test our method against two other popular approaches to the communication problem in robotics: 1) Euclidean Disk Model as used in[33, 80], where communication constraints are in terms of Euclidean distance; 2) Stochastic Gradient Approach, where we implement the Simultaneous Perturbation method (SPSA)[159] for estimating the gradient of signal power by sampling

[3] Note that the data-rate is capped by 60 Mb/s causing the plot to appear flat at times unlike ESNR.

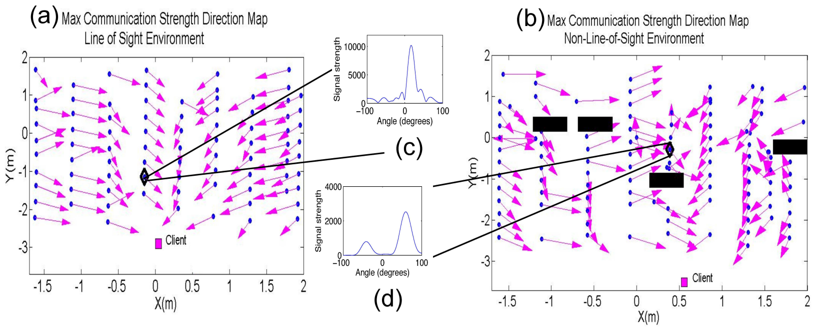


Figure 9-11: Gradient Field Visualization. Gradient field of θ_{max} and power profile for (a) Line-of-sight and (b) Non-Line-of-Sight.

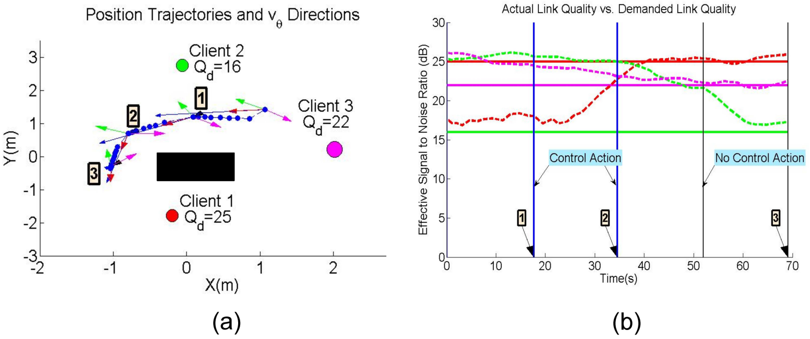


Figure 9-12: Non-Line-of-Sight Robot Router Trajectory. (a) Depicts testbed with robot router servicing three clients in a candidate non-line-of-sight setting. The blue line depicts the trajectory, and colored arrows indicate instantaneous θ_{max} for the corresponding clients. (b) Plots the ESNR across time (as dotted lines) for each client through the experiment. Solid lines denote client demands.

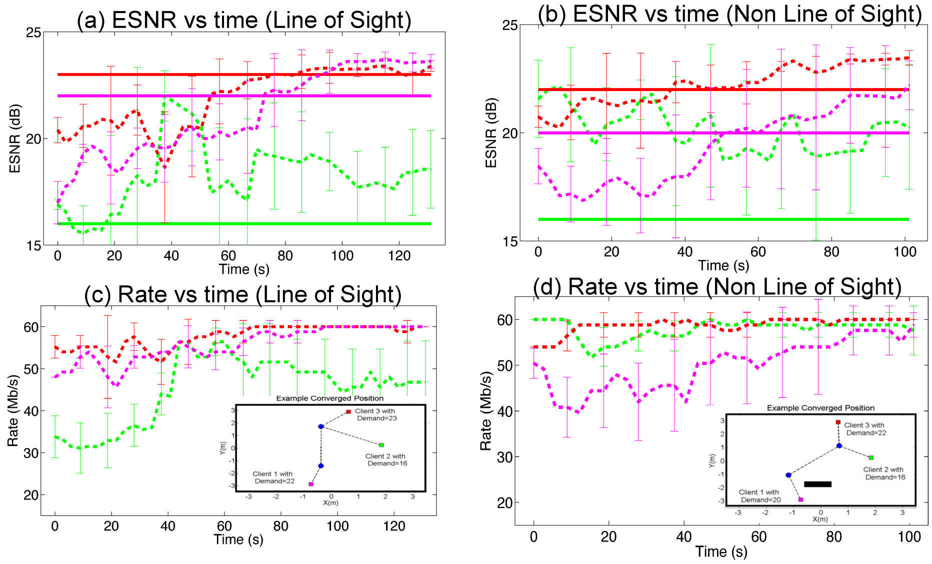


Figure 9-13: Aggregate ESNR and Rate Measurements. Aggregate results obtained over 5 runs show demands are consistently met even in the presence of obstacles as demonstrated by the candidate converged solutions.

the ESNR (which provides greater granularity than RSSI), along randomized directions, similar to the approach utilized by [95]. For the generation of each direction in the SPSA method we use a Bernoulli random variable (as in [159]) and diminishing step sizes satisfying the conditions stated in [159] for convergence. Our largest step size was allowed to be the same maximum vehicle velocity of v_c for all experiments. We consider a robotic router and three clients, each with an ESNR demand of 20 dB. We repeat the experiment five times in the non-line-of-sight environment in Fig. 9-14(b)-(d). In each instance, we measure r_{max}, the maximum ratio of ESNR demand versus the ESNR achieved among all three clients. In particular, r_{max} is below one at the converged position (i.e. all client demands are satisfied), and above one otherwise.

Results: Fig. 9-14(a) plots the aggregate mean and variance of r_{max} across time, for all the three approaches. Fig. 9-14(b)-(d) show a candidate trajectory adopted by the robotic router for the three schemes. The plots demonstrate while the disk model converges quickly to a solution, ignorance of the wireless channels leads to solutions not meeting client demands; especially in non-line-of-sight settings. In contrast, the stochastic gradient approach (in blue), which sample the instantaneous ESNR, eventually satisfies network demands. However, the convergence is often laborious as the router often traverses counter-productive directions (see Fig. 9-14 (c)). Indeed such techniques are noisy at low signal power, as even a large change in distance translates to a small change in signal power (a well-studied problem, e.g. in [27, 188, 82]). Fig. 9-14(c) shows that this leads to areas at non-line-of-sight or far distances from the client, where the robot easily gets lost.

Our method leverages full channel information, including signal power and phase, to find the signal direction as opposed to just its magnitude. The result is an algorithm that converges to positions that satisfy network demands without the counter-productive exploration of a pure sampling approach.

■ 9.4.6 Robustness to Dynamic Obstacle Positions

We evaluate how our system adapts to changes in the environment without an *a priori* known map. Consider two robotic routers and three clients in an environment with an obstacle located initially as shown in Fig. 9-15(a). We allow the robot routers to navigate to their converged positions. At $t = 120$ sec, we move the obstacle to a different location as in Fig. 9-15(c), and let the routers reconverge.

Results: Fig. 9-15(b) and Fig. 9-15(c) depict the converged position of the routers before and after the obstacle was moved. Fig. 9-15(d) plots the data-rate across time for each client. The plot shows that our system satisfies client demands at the initial position. It also recovers from the sharp fall in rate at one client and successfully re-converges after the obstacle is moved.

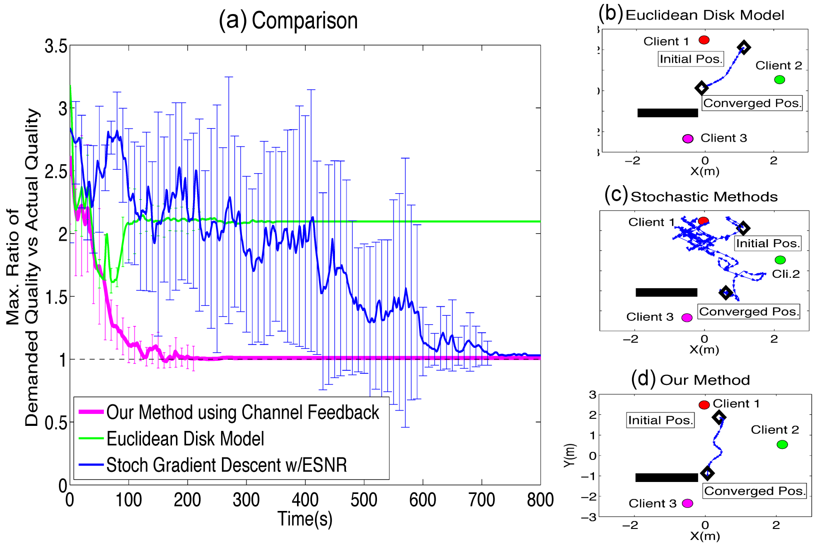


Figure 9-14: Comparison Plots comparing our method against the Euclidean disk model and a stochastic gradient descent method based on ESNR. Our method both converges to a position that meets communication demands, and converges quickly along an efficient path.

9.5 Experimental Results without Motion Capture Support

In this section, we evaluate our system in a large complex indoor environment with concrete walls and columns without any motion capture support (see Fig. 9-16). Instead, we use a constant velocity assumption to infer the relative displacements, $d(t)$ (see Algorithm 4) , of the Wi-Fi antenna on the router. The requirements for obtaining $d(t)$ as described in Section 9.2, are that the robot router moves at a constant known velocity over the time window required for computing SAR. Thus in our experiment we command the iRobot Create platform to move with a known constant velocity between control actions.

9.5.1 Gradient Field in Complex Environments

In this experiment, we measure the gradient field capturing the direction of maximum signal strength across spatial locations in the above testbed without Vicon support. We place a robotic router and client and line-of-sight (LOS) and non-line-of-sight (NLOS) as in Fig. 9-16. We trace the router's gradient field towards the client starting from multiple initial positions.

Results: Fig. 9-16 (a) and (b) plot of candidate trajectories (from gradient field) in LOS and NLOS across initial locations. The plots show that our system successfully navigates towards the client to satisfy its demands, without knowledge of the environment or client location.

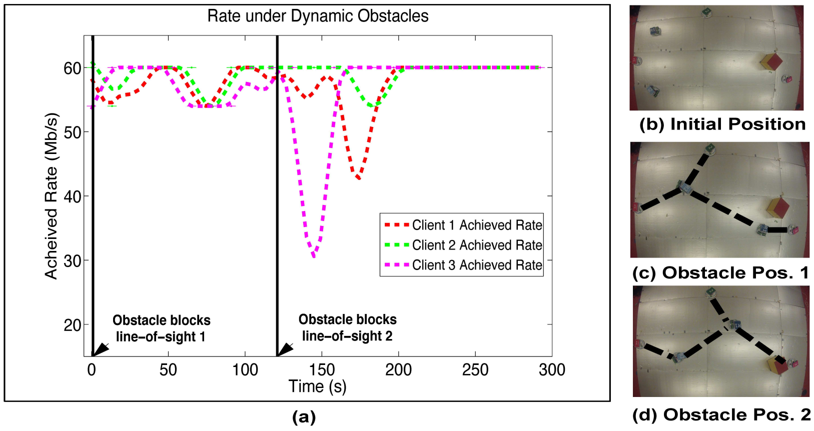


Figure 9-15: Effect of Moving an Obstacle. These plots show the result of disturbing the wireless channels via movement of a line-of-sight obstructing obstacle. Actual testbed snapshots are shown on the right.

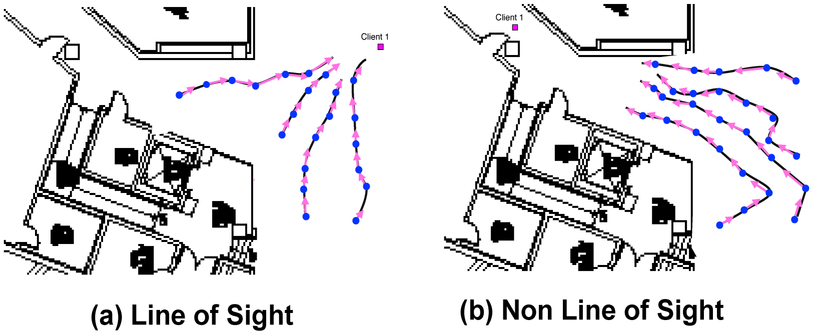


Figure 9-16: Complex Environments. Trajectories using measured $\vec{v}_{\theta_{\max}}$ directions satisfy a client's demand in line-of-sight and non-line-of-sight settings in complex indoor environments.

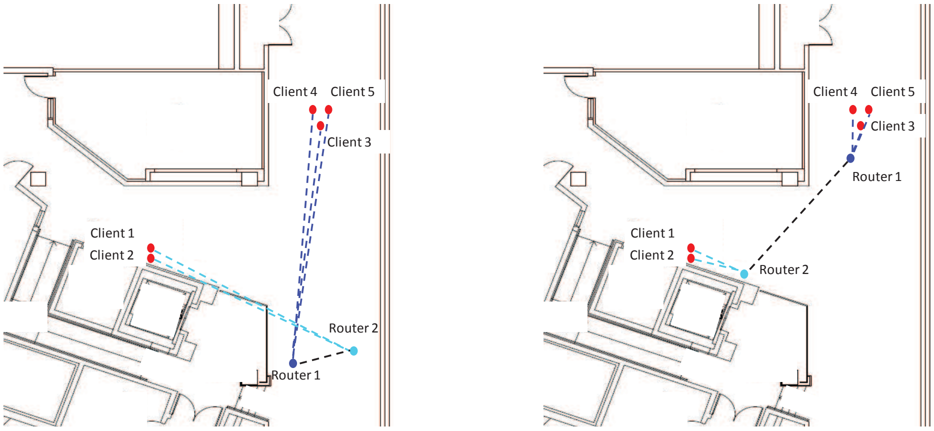
9.5.2 Full-Scale Experiment in Complex Environments

We implement a full-scale experiment of two routers and five clients in the complex indoor environment described in Sec. 9.5.1 above with no motion capture support. Clients in this case are static Asus EEE Seashell series netbooks and routers are AscTec Atom boards mounted on mobile iRobot Create platforms.

Clients are positioned in two clusters along orthogonal hallways, ie. a non-convex environment. Routers are placed in the initial positions as shown in the floorplan in Figure 9-17(a). For these initial router positions, Client 1 and Client 2 are both out of direct line-of-sight as they are obstructed by a concrete wall.

The relative displacement of the Wi-Fi antenna, required by Algorithm 4 to obtain a directional signal strength profile, is measured by assuming the router moves at a constant velocity (see Sec. 9.2.2).

Before calculating the next waypoint, each router is commanded to move at constant velocity



(a) Initial Positions (b) Final Positions

Figure 9-17: Full-scale Experiment in Complex Indoor Environments. Full-scale experimental setup with initial (a) and final (b) configurations for a two-router five-client network configuration.

for a period of 24 seconds which is equivalent to two wavelengths in displacement. The commanded waypoint from the control Algorithm 7 is then provided as a heading/distance pair which is actuated by the router using dead reckoning.

Results: Figure 9-17(a) depicts the initial configuration of the network of routers and clients. The dotted lines indicate which cluster of clients each router is assigned to by the controller. These assignments are optimized by the controller based on the observed ESNR values as described in Algorithm 7. Figure 9-17(b) show the converged positions of the routers indicating that all client demands are satisfied at these positions. In particular, Figure 9-18 demonstrates the trajectories traversed by Router 1 and Router 2 (left top and left bottom respectively) and the corresponding ESNR curves for each router's assigned clients on the right column. The ESNR curves are averaged over a window of 24 seconds as the router moves along its trajectory, and the solid blue squares indicate the times where a control action was given. As shown by ESNR curves in Figure 9-18, all client demands are satisfied at the final router configuration.

■ 9.6 Trade-off: Local vs. Global Optimality

Our primary focus in the body of this chapter has been on developing a closed-loop controller that uses instantaneous feedback on wireless channels to position routers. This real-time feedback allows for routers to repair communication links on the fly as needed, for example in the case of dynamic obstacles that may occlude a link. However, here we point out that it is also possible to use the methods presented in this chapter to obtain a static directional map of signal strength, or gradient field, throughout the environment. In fact, the richness of directional profiles derived here would allow mapping to a level of accuracy that was previously unattainable for small mobile

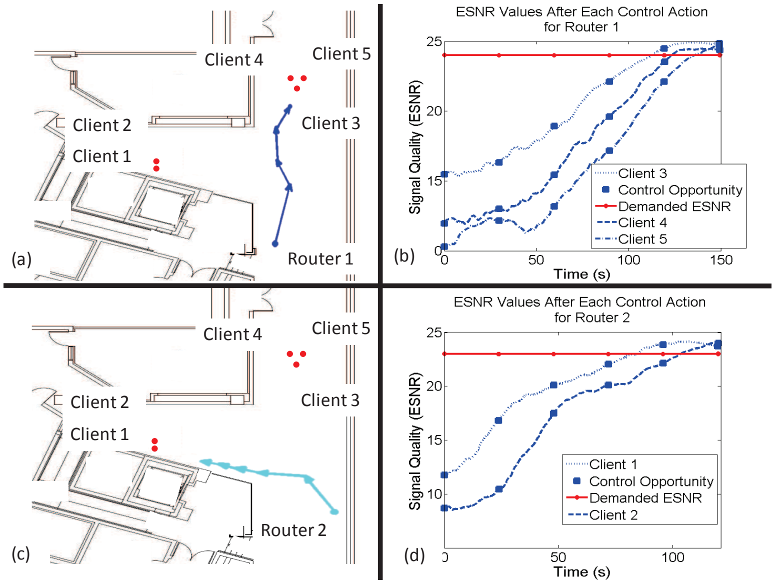


Figure 9-18: Router Trajectories. Router trajectories resulting from execution of commanded waypoints from Algorithm 7. These paths were executed via dead reckoning for Router 1 (a) and Router 2 (c). Corresponding measured ESNR curves for Router 1's clients (b) and Router 2's clients (d) respectively.

platforms. Such a gradient field (as in Figure 9-11) can be used to plan router placements that are globally optimal, in contrast to the local solutions provided here. However, it is important to point out that in this case the ability to adapt to changes in the environment, for example if obstacles or clients move in the environment), is lost since this would invalidate a static map. Therefore this is a trade-off that would have to be evaluated carefully for each situation.

■ 9.7 Related Work

Related work falls under two broad categories.

■ 9.7.1 Multi-Robot Coordination

Our work is related to past chapters on multi-robot coordination to achieve a collaborative task while supporting specific communication demands [95, 191, 113, 47]. Past work on this topic fall under two classes of approaches.

Euclidean Disk Model: The first class employs Euclidean disk assumptions where signal quality is assumed to be deterministic and mapped perfectly to the Euclidean distance between the communication nodes. A Euclidean metric allows for quadratic cost for the edges of the network and enables a geometric treatment of an otherwise complex problem. In reality, signal strength suffers from large variations over small displacements [101, 60] that these models simply do not capture. Yet, the simplicity afforded by these models has led to significant contributions including i) multi-agent coordination for coverage and flocking [147, 114], ii) assignment of routers to clients for attaining a prescribed level of connectivity [57, 44] or throughput [35], and iii) connectivity maintenance based on graph theoretic approaches [117, 37].

Stochastic Sampling Methods: Recently, efforts have focused on giving the communication quality over each link in the network a more realistic treatment by sampling the signal strength and building closed-loop controllers using this feedback. Such stochastic sampling methods either supplement theoretical models for signal strength with a stochastic component based on the collected samples [113, 101], or, use the collected samples to design stochastic gradient controllers [173, 95]. These chapters have studied stochastic sampling patterns for i) acquiring sufficient signal strength (RSS) samples [102, 101], ii) co-optimizing communication quality and other higher level tasks like motion planning or message routing [191, 48], iii) used router mobility to escape "deep fades" or null points where connectivity may be lost [175] or to map out the signal strength and resulting connectivity regions of the environment [173]. Unfortunately these works necessitate at least one of the following prohibitive requirements: i) motion of the routers along counter-productive paths to collect sufficient RSS samples, ii) assumptions of a known environment map, static surroundings, and known positions of communicating agents, or iii) previously acquired signal strength maps.

In comparison to these papers, we introduce a system that captures the magnitude of the signal arriving from different *directions*, as opposed to only its total magnitude at a particular position. This allows us to combine the best of both the disk model and stochastic sampling methods: Like the disk model, we do not require prior knowledge of the environment and its obstacles, or a model of channel's distribution. Like the stochastic methods, our approach accurately captures actual signal characteristics and hence can help multi-robot systems satisfy their desired communication demands in real-world environments. Figure 9-19 provides an illustrative example of how our system out-performs the Euclidean disk model and stochastic sampling, particularly in the presence of obstacles.

■ 9.7.2 Angle of Arrival Systems

Our method builds upon a rich body of literature in wireless networking that estimates actual angle-of-arrival of each of the reflected paths of a signal at a receiving device. Past work has employed two classes of hardware to estimate angle-of-arrival:

Antenna Arrays: Past literature has leveraged arrays of antennas to estimate angle-of-arrival for localization [179, 82, 188] and tracking [137]. These use stationary multi-antenna receivers to locate the transmitter with sub-meter accuracy. Unfortunate for the robotics community, many of these techniques require bulky, specialized hardware such as customized software radios, and are thus difficult to place on small, agile, mobile platforms that are ubiquitous for robotics applications.

Synthetic Aperture Radar: Understanding how to attain this directional information using a mov-

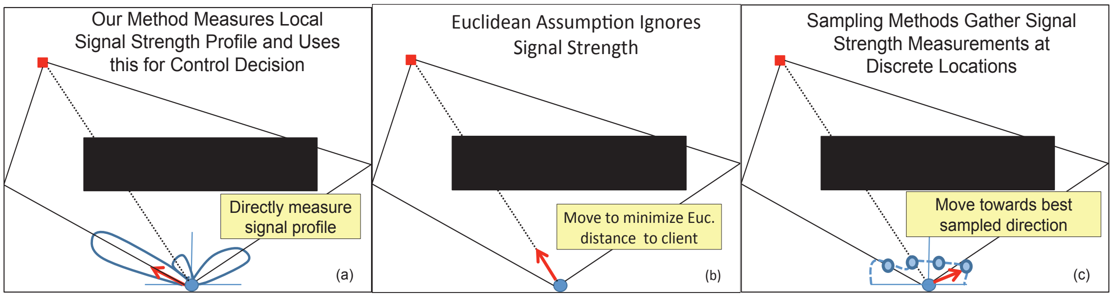

Our Method Measures Local Signal Strength Profile and Uses this for Control Decision
Directly measure signal profile
(a)
Euclidean Assumption Ignores Signal Strength
Move to minimize Euc. distance to client
(b)
Sampling Methods Gather Signal Strength Measurements at Discrete Locations
Move towards best sampled direction
(c)

Figure 9-19: Comparison with Past Work. Compares our method against the Euclidean disk model and a stochastic sampling. The figures depict the actual router (blue) and client (red) separated by an obstacle (black). The black lines indicate the different paths of the signal. (a) Our method estimates the actual signal power arriving from different angular directions, much like a high-resolution directional antenna would. This provides a sharpened peak in the direction of maximum signal strength. (b) The Euclidean Disk model guides the router along the shortest Euclidean path, which is greatly attenuated by the obstacle. (c) Stochastic methods measure the signal strength by moving the router and sampling at various positions (blue circles). The signal strength does not vary significantly between locations due to the lack of spatial resolution, ie. at each sample location the signal strength is a combination of signals arriving from all angular directions. This leads to much less discernible peaks when contrasted with (a) (Note that the polar plot of signal power, shown here as dotted lines, have peaks in the same angular directions as our method though less sharp). This method guides the router towards the direction of the best sample, which often may not be the actual direction of maximum signal strength, as shown.

ing platform is the subject of Synthetic Aperture Radar (SAR) [50]. SAR allows even a single-antenna mounted on a flying aircraft or satellite to emulate a multi-antenna array. Unfortunately, most SAR applications [50, 179, 178] are geared towards radar-type problems (eg. imaging, RFID applications) where signals are transmitted and processed by the same node. Therefore, they cannot be used to analyze the direction of arrival of the signal from a distinct transmitter (e.g. Wi-Fi devices).

For an adaptive communication network of small router robots, we need a light-weight, single-antenna system that can perform SAR using two-way transmissions (unlike radar) on off-the-shelf Wi-Fi devices. In this regard, we develop a system that builds upon synthetic aperture radar meant for robotic routers and clients equipped with standard Wi-Fi cards.

■ 9.8 Discussion

In this chapter, we present a framework to satisfy real-time variable communication demands for a changing network. We develop a solution enabling a robotic receiver to find the profile of signal strength across spatial directions for each sender of interest. While our technique retrieves these spatial signal profiles in real time, we note that it faces an important limitation: it assumes access to wireless channels from both the transmitter and the receiver. Developing a system that can work with unmodified transmitters remains an open challenge. Our system integrates the signal profiles with a controller that optimizes communication quality while maintaining quadratic edge costs, and thus has natural extensions to many communication-aware coordination problems such as coverage[33], consensus[129], formation control[80], etc. We believe our system provides the necessary robustness to bring the benefits of these important contributions to practical robotic systems.

CHAPTER 10
Guaranteeing Spoof-Resilient Multi-Robot Networks

Multi-robot networks rely on wireless communication to enable a wide range of tasks and applications: coverage [132, 33, 148], disaster management [110], surveillance [18], and consensus [130] to name a few. The future promises an increasing trend in this direction, such as delivery drones which transport goods (e.g., Amazon Prime Air [1]) or traffic rerouting algorithms (e.g., Google Maps Navigation) that rely on broadcasted user locations to achieve their goals. Effective coordination, however, requires trust. In order for these multi-robot systems to perform their tasks optimally, transmitted data is often assumed to be accurate and trustworthy; an assumption that is easy to break. A particularly challenging attack on this assumption is the so-called "Sybil attack."

In a Sybil attack a malicious agent generates (or spoofs) a large number of false identities to gain a disproportionate influence in the network.[1] These attacks are notoriously easy to implement [152] and can be detrimental to multi-robot networks. An example of this is coverage, where an adversarial client can spoof a cluster of clients in its vicinity in order to create a high local demand, in turn denying service to legitimate clients (Figure 10-1). Although a vast body of literature is dedicated to cybersecurity in general multi-node networks (e.g., a wired LAN), the same is not true for multi-robot networks [72, 146], leaving them largely vulnerable to attack. This is because many characteristics unique to robotic networks make security more challenging; for example, traditional key passing or cryptographic authentication is difficult to maintain due to the highly dynamic and distributed nature of multi-robot teams where clients often enter and exit the network.

This chapter addresses the challenge of guarding against Sybil attacks in multi-robot networks. We focus on the general class of problems where a group of server robots coordinate to provide some service using the broadcasted locations of a group of client robots. Our core contribution is a novel algorithm that analyzes the received wireless signals to detect the presence of spoofed

[1] Please refer to [40, 126] for a detailed treatment of this class of cyber attacks.

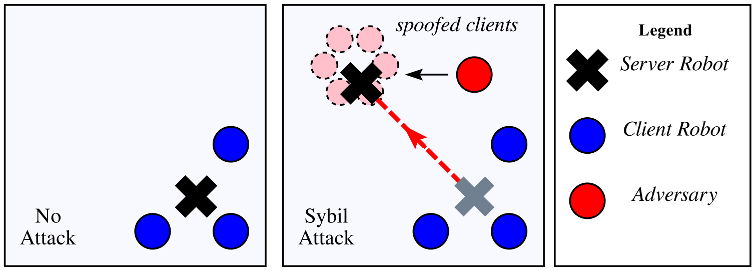


Figure 10-1: Sybil Attack on Coverage. A server robot provides locational coverage to legitimate clients when no attack is present. In a Sybil attack, an adversary spoofs many fake clients to draw away coverage from the legitimate clients.

clients spawned by adversaries. We call this a "virtual spoofer sensor" as we do not use specialized hardware nor encrypted key exchange, but rather a commercial Wi-Fi card and software to implement our solution. Our virtual sensor leverages the rich physical information already present in wireless signals. At a high level, as wireless signals propagate, they interact with the environment via scattering and absorption from objects along the traversed paths. Carefully processed, these signals can provide a unique signature or "spatial fingerprint" for each client, measuring the power of the signal received along each spatial direction (Fig. 10-2). Unlike message contents such as reported IDs or locations which adversaries can manipulate, spatial fingerprints rely on physical signal interactions that cannot be exactly predicted [60, 113].

Using these derived fingerprints, we show that a confidence weight, $\alpha \in (0,1)$ can be obtained for each client in the network. We prove that these confidence weights have a desirable property where legitimate clients have an expected confidence weight close to one, while spoofed clients will have an expected confidence weight close to zero. A particularly attractive feature of confidence weight α is that it can be readily integrated as a per-client weighting function into a wide variety of multi-robot controllers. More importantly, the analytical bounds on these weights can provably limit the ill-effects of spoofers on the performance of these controllers. This chapter demonstrates this capability in the context of the well-known locational coverage algorithm [33, 148].

We provide an extensive experimental evaluation of our theoretical claims using a heterogeneous team of air/ground robots consisting of two AscTec Hummingbird platforms and ten iRobot Create platforms. We conduct our experiments in general indoor settings with randomly placed clients and demonstrate a spoofer detection rate of 96%. For the case of coverage we find that the converged positions of the service robots is on average $3\ cm$ from optimal even when more than 75% of total clients in the network are spoofed.

Contributions: We develop a virtual sensor for spoofing detection which provides performance guarantees in the presence of Sybil attacks and is applicable to a broad class of problems in dis-

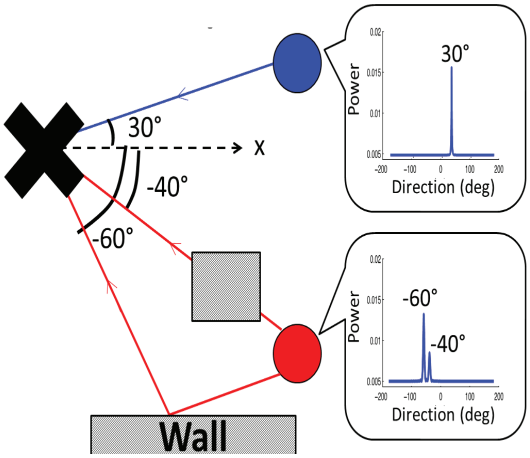


Figure 10-2: Spatial Fingerprints: A quadrotor server measures the directional signal strength of each client (here, simplified to 2-D). The blue client has one line-of-sight peak; the other, 2 signal paths.

tributed robotics. We show that the influence of spoofers is analytically bounded under our system in a coverage context, where each robotic node providing coverage remains within a radius of its position in the absence of an attack. Our theoretical results are validated extensively through experiments in diverse settings.

■ 10.1 Problem Statement

This chapter focuses on problems where the knowledge of agent positions facilitates some collaborative task. Specifically, it assumes two groups of agents, "clients" requiring some type of location-based service such as coverage or goods delivery and "servers" whose positions are optimized in order to provide the service to its clients. Let $P := \{p_1, \dots, p_c\}$ denote the client positions in $\mathbb{R}^3$. Let $X := \{x_1, \dots, x_m\}$ be the positions of the servers in $\mathbb{R}^3$ and the notation $[m] = \{1, \dots, m\}$ denote their indices. We consider the case where a subset of the clients, $S \subset P$ (with $s := |S|$) are "spoofed" clients.

Spoofed Client A single malicious client may generate multiple unique identities, each with a fabricated position. Each generated, or "spawned" identity is considered a *spoofed client*. By spoofing multiple clients, the malicious client gains a disproportionate influence in the network. All clients which are not spoofed are considered *legitimate clients*.

Threat Model: Our threat model considers one or more adversarial robot clients with one Wi-Fi antenna each. The adversaries can be mobile and scale power on a per-packet basis. We only consider adversarial clients.[2] Adversarial clients perform the "Sybil Attack" to forge packets em-

[2] The case of adversarial server robots is left for future work although many of the concepts in the current chapter are extensible to this case as well.

ulating s non-existent clients, where s can exceed the number of legitimate clients. More formally:

Sybil Attack Define a network of client and server positions as $P \cup X$, where a subset S of the clients are spoofed, such that $P = S \cup \tilde{S}$. We assume that set P is known but knowledge of which clients are spoofed (i.e., in S) is unknown. This attack is called a "Sybil Attack."

To counter the Sybil attack, this chapter has two objectives. First, we find a relation capturing directional signal strength between a client i and a server l. We seek a mapping $F_{il} : [0, \frac{\pi}{2}] \times [0, 2\pi] \mapsto \mathbb{R}$ such that for any 3D direction (θ, ϕ) defined in Fig. 10-4, the value $F_{il}(\theta, \phi)$ is the power of the received signal from client i along that direction. Using this mapping, or "fingerprint", our first problem is to derive a *confidence weight* whose expectation is provably bounded near 1 for legitimate clients and near 0 for spoofed clients. Further, we wish to find these bounds analytically from problem parameters like the signal-to-noise ratio of the received wireless signal. We summarize this objective as Problem 1 below:

Problem 1: Spoofer Detection. *Let $\mathcal{F}_i$ be the set of fingerprints measured from all clients $j \in [c]$ and servers $l \in [m]$ in the neighborhood, $\mathcal{N}_i$, of client i.*[3] *Here, a neighborhood of client i, $\mathcal{N}_i$, are all agents that can receive Wi-Fi transmissions sent by client i. Using $\mathcal{F}_i$, derive a confidence weight $\alpha_i(\mathcal{F}_i) \in (0, 1)$ and a threshold $\omega_i(\sigma_i^2) > 0$ where σ_i^2 represents error variances such as the signal-to-noise ratio that are assumed to be given. Find $\omega_i(\cdot)$ to have the provable property of differentiating spoofed clients whereby spoofed clients are bounded below this threshold, i.e., $E[\alpha_i] \leq \omega$, and legitimate clients are bounded above this threshold $E[\alpha_i] \geq 1 - \omega$.*

Our second objective is to apply our spoofer detection method to multi-robot control problems. We consider the well-known coverage problem in [33, 148]. We show that by integrating the confidence weight from Problem 1, we can analytically bound the error in performance caused by spoofed clients in the network. We consider the coverage problem where an importance function is defined over an environment and where the positions of the clients correspond to peaks in the importance function. Here, servers position themselves to maximize their proximity to these peaks, to improve their coverage over client robots. If $C_V = \{x_1^*, \ldots, x_m^*\}$ is the set of server positions optimized by the coverage controller with zero spoofers, we wish to guarantee that server positions optimized with spoofers present, C_{V_α}, is "close" to C_V. We state this second objective more specifically as Problem 2 below:

Problem 2: Sybil-resillience in Multi-Robot Coverage. *Consider a locational coverage problem where an importance function $\rho(q) > 0$ is defined over an environment $\mathcal{Q} \subset \mathbb{R}^3$ and $q \in \mathcal{Q}$. Specifically, consider an importance function that can be decomposed into terms, $\rho_i(q)$, depending on each client's position, $i \in [c]$ (for*

[3] Detecting if a client i is spoofed becomes easier given more servers communicating with i (i.e., a larger neighborhood $\mathcal{N}_i$). But even with a single server, this determination can be made. A theoretical treatment of this point is given in Sec. §10.3 and experimental results (§10.5.1) use as little as one server.

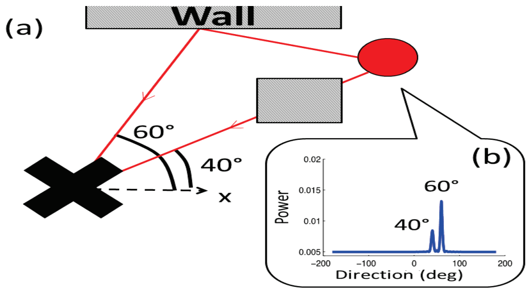


Figure 10-3: Example Signal Fingerprint: (a) A server (×) receives a client signal on 2 paths: direct along 40° attenuated by an obstacle (shaded) and reflected by a wall along 60°. (b) is a corresponding fingerprint: peak heights at 40° and 60° correspond to their relative attenuations.

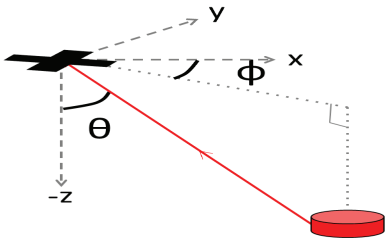


Figure 10-4: 3-D Angles: The figure depicts the notation for the azimuthal angle ϕ and polar angle θ for the direct path from a ground client to aerial server robot (×) in 3 dimensions. More generally, the set of all angles between client i and server l are denoted as Φ_{il}, Θ_{il} respectively.

example, each client position corresponds to a peak), i.e., $\rho(q) = \rho_1(q) + \ldots + \rho_c(q)$*. Let* $C_V = \{x_1^*, \ldots, x_m^*\}$ *be the set of server positions returned by an optimization of* $\rho(q)$ *over* X*, where there are zero spoofed clients in the network. Under a Sybil attack, let* $C_{V_\alpha} = \{x_1, \ldots, x_m\}$ *be the set of server positions returned by an optimization of an* α-modified *importance function* $\rho(q) = \alpha_1\rho_1(q) + \ldots + \alpha_c\rho_c(q)$ *where the importance weight terms* α_i *satisfy the bounds stated in Problem 1. We wish to find an* $\varepsilon(\mathcal{P}) > 0$ *such that the set* C_{V_α} *is within a distance* $\varepsilon(\mathcal{P})$ *to* C_V*.* C_{V_α} *is within a distance* $\varepsilon(\mathcal{P})$ *to* C_V *if* $\forall x \in C_{V_\alpha}$ *there exists a unique* $y \in C_V$ *where* $dist(x, y) < \varepsilon(\mathcal{P})$*. Here,* $\mathcal{P}$ *is a set of problem parameters that we wish to find.*

Intuitively, solutions to Problem 2 guarantee that under a Sybil attack, all server positions computed using an α-modified coverage controller are within a computable distance $\varepsilon(\mathcal{P})$ from their optimal positions (i.e., in the absence of spoofers). Sec. §10.4 derives a closed-form for $\varepsilon(\mathcal{P})$ and shows the set $\mathcal{P}$ of problem parameters to be the number of spoofers, the footprint of the environment covered, and signal noise.

■ 10.2 Fingerprints to Detect Malicious Clients

Here we construct a *fingerprint*, a directional signal strength profile for a communicating server-client pair. Our choice of signal fingerprints have many desirable properties that enable us to derive a robust spoof-detection metric: they *1)* capture directional information of the transmitted signal source and thus are well-suited for flagging falsely reported client positions, *2)* can be obtained for a single server-client pair, unlike location estimation techniques such as triangulation which require multiple servers to coordinate, *3)* cannot be manipulated by the client, since the occurrence of each signal path is due to environment reflections, *4)* are applicable in complex multipath environments where a transmitted signal is scattered off of walls and objects;since these scattered signals manifest themselves as measurable peaks in the fingerprint, complex multipath contributes significantly to fingerprint uniqueness.

We construct fingerprints using wireless channels h, complex numbers measurable on any wireless device characterizing the attenuation in power and the phase rotation that signals experience as they propagate over the air. These channels also capture the fact that wireless signals are scattered by the environment, arriving at the receiver over (potentially) several different paths [199]. Fig. 10-3 is an example 2D schematic of a wireless signal traversing from a client robot to a server robot arriving along two separate paths: one attenuated direct path at $40°$ and one reflected at $60°$. If the server robot had a directional antenna, it could obtain a full 3D profile of power of the received signal (i.e., $|h|^2$) along *every* spatial direction. We use such a 3-D profile as a "spatial fingerprint" that can help distinguish between different clients.

Unfortunately directional antennas are composed of large arrays of many antennas that are too bulky for small agile robot platforms. Luckily, a well-known technique called Synthetic Aperture Radar [50] (SAR) can be used to emulate such an antenna using a commodity Wi-Fi radio. Its key idea is to use small local robotic motion, such as spinning in-place, to obtain multiple snapshots of the wireless channel that are then processed like a directional array of antennas. SAR can be implemented using a well-studied signal processing algorithm called MUSIC [71] to obtain spatial fingerprints at each server robot.

Mathematically, we obtain a spatial fingerprint for each wireless link between a server l and client i as a matrix $F_{il} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$. For each spatial path represented as (θ, ϕ) (see Fig. 10-4), F_{ij} maps to a scalar value representing the signal power received along that path. More formally:

$$F_{il}(\phi, \theta) = 1/|Eig_n(\hat{\mathbf{h}}_{\mathbf{il}}\hat{\mathbf{h}}_{\mathbf{il}}^{\dagger})e^{\sqrt{-1}\mathbf{S}_{\mathbf{il}}(\phi,\theta)}|^2 \tag{10.1}$$

Where $\hat{\mathbf{h}}_{\mathbf{il}}$ is a vector of the ratio of wireless channel snapshots between two antennas mounted on the body of the server l and $\mathbf{S}_{\mathbf{il}}(\phi, \theta) = \frac{2\pi r}{\lambda}\cos(\phi - \mathbf{B}_{\mathbf{l}})\sin(\theta - \mathbf{G}_{\mathbf{l}})$, λ is the wavelength of the signal and r is the distance between the antennas, $\mathbf{B}_{\mathbf{l}}, \mathbf{G}_{\mathbf{l}}$ are the server's angular orientation, $Eig_n(\cdot)$ are noise eigenvectors, $(\cdot)^{\dagger}$ is conjugate transpose, and k is the number of signal eigenvectors, equal to the number of paths.

While our above formulation is derived from MUSIC [71], it varies in one important way: while MUSIC uses a single-antenna channel snapshot h_{il}, we use the channel ratio $\hat{h}_{il} = h_{1_{il}}/h_{2_{il}}$ between two antennas. This modification provides resilience to intentional power scaling by the sender since scaling his transmit power by χ yields a measured ratio $\hat{h}_{il} = \chi h_{1_{il}}/(\chi h_{2_{il}})$; a value unaffected by power scaling.

■ 10.3 Constructing a Client Confidence Weight

Given a client fingerprint $F_{il}(\phi, \theta)$ for each client i relative to a robotic server l, we wish to generate a confidence weight $\alpha_i \in [0, 1]$ that approaches 1 for legitimate clients, and 0 otherwise. We achieve

Symbol	Meaning
m, c, s	No. of servers, clients, spoofers
p_i, x_l	Position of client i / server l
F_{il}, k	Fingerprint of i at l, k peaks
$\hat{\mathbf{h}}_{il}$	$M \times 1$ channel ratios of i to l
$f(\cdot\,;\mu,\sigma^2)$	PDF of normal distribution
$g(\cdot\,;\mu,\sigma^2)$	$\min(1, \sqrt{2\pi}f(x;\mu,\sigma^2))$
κ	Constant = $((\sqrt{2}+\sqrt{\pi})/\pi)^2$
α_i, β_i	confidence, honesty metric of i
γ_{ij}	Similarity metric of client i, j
SNR	Signal-to-noise ratio
RSSI	Received Signal Strength
σ_θ^2, σ_ϕ^2	Variance in peak shifts of F_{il}
$\hat{\sigma}_\theta^2$, $\hat{\sigma}_\phi^2$	σ_θ^2, σ_ϕ^2 plus measurement error
C_{V_L}, C_{V_α}	Coverage centroid of optimal, our system; error $\vec{e}$ within ε
$L(Q), \rho(q)$	Footprint, Mass function

Figure 10-5: Table of Most Common Notations

this by defining α_i as the product of two terms β_i and γ_{ij} that go to 0 if a client reports a falsified location or has the same fingerprint as another client j respectively. In particular, β_i is termed the *honesty* metric and is the likelihood (Eq. (10.2)) that client i is indeed along its reported direction (ϕ_{il}, θ_{il}) with respect to each server l in its neighborhood. The second term γ_{ij} is the *similarity* metric - the likelihood that client i's fingerprint as seen by server l is not unique compared to that of a different client j of server l. Finally, α_i is the product of 1) β_i and 2) $(1-\gamma_{ij})$ over all $j \neq i$, which compares client i's fingerprint with all other clients in its neighborhood and approaches 0 if client i's profile is not unique. Therefore if either the honesty term or similarity term goes to 0, the weight α_i for client i also approaches zero.

$$\alpha_i = \beta_i \prod_{j \neq i} (1-\gamma_{ij}) \ \text{ where, } \ \beta_i = \prod_{l \in \mathcal{N}_i} \mathcal{L}(i \text{ is at } (\phi_{il}, \theta_{il}) | F_{il})$$

$$\gamma_{ij} = \prod_{l \in \mathcal{N}_i} \mathcal{L}(i \text{ spoofs } j | F_{il}, F_{jl}) \tag{10.2}$$

Here, $\mathcal{L}(\cdot)$ denotes an event likelihood, (ϕ_{il}, θ_{il}) is the reported direction of client i with respect to server l, and the neighborhood $\mathcal{N}_i$ are servers communicating with client i.

Defining Honesty and Similarity Metrics: The honesty metric β_i and similarity metric γ_{ij} are derived using peak locations in client fingerprints. In practice however, peaks may have slight shifts owing to noise. Thus, any comparison between peak locations must permit some variance due to these shifts. Fortunately, noise in wireless environments can be modeled closely as additive white-Gaussian [199]. As the following lemma shows, this results in peak shifts that are also Gaussian, meaning that their variance is easy to model and account for. More formally, the lemma states that shifts are normally distributed with zero mean and well-defined variance, based on the wireless medium's signal-to-noise ratio (SNR):

Lemma 10.1 *Let $\Delta\theta_i, \Delta\phi_i$ denote the error between the azimuthal and polar angle of the uncorrelated i^{th} path of a (potentially multipath) source and the corresponding angles of the (local) maximum in the fingerprint $F(\phi, \theta)$, over several uniformly gathered packets (i.e., SAR snapshots) for $\theta \in (10°, 80°)$. Then $\Delta\theta_i$ and $\Delta\phi_i$ are normally distributed with a mean 0, and expected variance σ_ϕ^2 and σ_θ^2:*

$$\sigma_\theta^2 = \sigma_\phi^2 = 9\lambda^2/(8M\pi^2r^2SNR)$$

Where, λ is the wavelength of the signal, SNR is the signal-to-noise ratio in the network[4], M is the number of packets per-rotation, and r is the distance between the antennas. □

Proof of Lemma 10.1: Cramer-Rao bounds calculate estimated performance for given geometries of antenna trajectories in an algorithm-independent manner. It is well-known that algorithms such as MUSIC achieve the Cramer-Rao bound (CRB) [54, 55, 161]. Using the result in [161], for a sufficiently large number of packets M, the error in angle for the uncorrelated path i using the Cramer-Rao bound has a mean 0 and variance σ^2 where:

$$\sigma^2 = \frac{1}{2SNR}\mathrm{Re}\left[D^*D - D^*A(A^*A)^{-1}A^*D\right]^{-1} \tag{10.3}$$

Where SNR is the signal to noise ratio in the network, A is the steering vector of the MUSIC algorithm [71] and D is its derivative with respect to the angle. Let us consider the quadrotor retrieves samples at a set of uniform angles $\{0, \Delta, 2\Delta, \ldots, (M-1)\Delta\}$, where $\Delta = \frac{2\pi}{M}$. For the azimuthal angle, these values are given by:

$$\begin{aligned}
A &= [1, e^{j\frac{2\pi r \cos\phi_i \sin\theta_i}{\lambda}}, \ldots, e^{j\frac{2\pi r \cos((M-1)\Delta - \phi_i)\sin\theta_i}{\lambda}}]^T \\
D &= \left[0, \ldots, \frac{2\pi r \sin\theta_i \sin((M-1)\Delta - \phi_i)}{\lambda} e^{j\frac{2\pi r \cos((M-1)\Delta - \phi_i)\sin\theta_i}{\lambda}}\right]^T \\
A^*A &= M \\
D^*A &= \sum_{j=1}^{M} \frac{2\pi r \sin\theta_i}{\lambda} \sin((j-1)\Delta - \phi_i) \to 0\text{, as } M \to \infty \\
\frac{1}{M}D^*D &= \left[\frac{2\pi r \sin\theta_i}{\lambda}\right]^2 \sum_{j=1}^{M} \sin^2((j-1)\Delta - \phi_i) \to \frac{2\pi^2 r^2 \sin^2\theta_i}{\lambda^2}\text{, as } M \to \infty
\end{aligned}$$

Substituting these values in Eqn. 10.3, we have:

$$\sigma_\phi^2 = \frac{(2\pi^2 M r^2 \sin^2\theta_i/\lambda^2)^{-1}}{2SNR} = \frac{\lambda^2}{4\pi^2 r^2 M \sin^2\theta_i SNR} \tag{10.4}$$

Similarly, for the polar angle, we can write:

$$\begin{aligned}
D &= \left[0, \ldots, \frac{2\pi r \cos\theta_i \cos((M-1)\Delta - \phi_i)}{\lambda} e^{j\frac{2\pi r \cos((M-1)\Delta - \phi_i)\sin\theta_i}{\lambda}}\right]^T \\
D^*A &= \sum_{j=1}^{M} \frac{2\pi r \cos\theta_i}{\lambda} \cos((j-1)\Delta - \phi_i) \to 0\text{, as } M \to \infty \\
\frac{1}{M}D^*D &= \left[\frac{2\pi r \cos\theta_i}{\lambda}\right]^2 \sum_{j=1}^{M} \cos^2((j-1)\Delta - \phi_i) \to \frac{2\pi^2 r^2 \cos^2\theta_i}{\lambda^2}\text{, as } M \to \infty
\end{aligned}$$

Again, substituting these values in Eqn. 10.3, we have:

$$\sigma_\theta^2 = \frac{(2\pi^2 M r^2 \cos^2\theta_i/\lambda^2)^{-1}}{2SNR} = \frac{\lambda^2}{4\pi^2 r^2 M \cos^2\theta_i SNR} \tag{10.5}$$

[4] For clarity, we drop dependence on i, l for SNR, σ_θ and σ_ϕ

Note that these values for σ_ϕ^2 and σ_θ^2 are defined for large values of M. Applying the central-limit theorem, it is well known that the average of any distribution over a large number of samples is asymptotically normal [19]. As a result, the distributions of ϕ_i and θ_i are asymptotically normally distributed with a mean of zero and variance given by Eqn. 10.4 and Eqn. 10.5. Assuming θ is uniformly in $(10°, 80°)$, and using the corresponding expected values of $1/\sin^2\theta_i$ and $1/\cos^2\theta_i$ to be 4.5, we have:

$$\begin{aligned}\sigma_\phi^2 =& \frac{9\lambda^2}{8M\pi^2r^2SNR}\\ \sigma_\theta^2 =& \frac{9\lambda^2}{8M\pi^2r^2SNR}\end{aligned}$$

This proves the required lemma. □

The above lemma follows from well-known Cramer-Rao bounds [115, 55, 54] shown previously for linear antenna movements in SAR [161] but readily extensible to circular rotations. Using this lemma, we can define the honesty metric β_i as the likelihood that the client is at its reported location, subject to this Gaussian error and additional measurement error in reported locations.

Definition *(β_i)* Let $\phi_{F_{il}}$ and $\theta_{F_{il}}$ denote the closest maximum in $F_{il}(\phi, \theta)$ to (ϕ_{il}, θ_{il}). We denote $\hat{\sigma}_\phi^2$ and $\hat{\sigma}_\theta^2$ as the variances in angles σ_ϕ^2 and σ_θ^2 plus any variance due to measurement error of reported locations that can be calibrated from device hardware. We define β_i for client i as:

$$\beta_i = \prod_l g(\phi_{il} - \phi_{F_{il}}; 0, \hat{\sigma}_\phi^2) \times g(\theta_{il} - \theta_{F_{il}}; 0, \hat{\sigma}_\theta^2) \tag{10.6}$$

Where $g(x; \mu, \sigma^2) = \min(1, \sqrt{2\pi} f(x; \mu, \sigma^2))$ is a normalized Gaussian PDF $f(x; \mu, \sigma^2)$ with mean μ and variance σ^2. □

In practice, reported client locations are subject to measurement errors due to position sensor inaccuracies. Our definition of β_i above accounts for this by using the effective variances $\hat{\sigma}_\phi^2$ and $\hat{\sigma}_\theta^2$ that are the sum of the variance in angles, σ_ϕ^2 and σ_θ^2, in addition to the variances due to measurement error.

Using Lemma 10.1 we define the similarity metric γ_{ij} as the likelihood that two client fingerprints share identical peaks:

Definition (γ_{ij}) Let (Φ_{il}, Θ_{il}) and (Φ_{jl}, Θ_{jl}) denote the set of local maxima, ordered by non-decreasing angle values, in fingerprints F_{il} and F_{jl}. We define γ_{ij} for client i relative to client j as:

$$\gamma_{ij} = \prod_{\phi_i \in \Phi_{il}, \phi_j \in \Phi_{jl}} g(\phi_i - \phi_j; 0, 2\sigma_\phi^2) \prod_{\theta_i \in \Theta_{il}, \theta_j \in \Theta_{jl}} g(\theta_i - \theta_j; 0, 2\sigma_\theta^2) \tag{10.7}$$

Where $g(\cdot;\mu,\sigma^2)$ is from Definition. 10.3, and the factor of 2 in the variance accounts for computing the difference of two normally distributed values. □

Defining the Confidence Weight: . We notice that Eqn. 10.2, 10.6 and 10.7 fully define α_i for each client i. In summary, the confidence weight is computed in three steps: (1) Obtain the client fingerprint using SAR on wireless signal snapshots. (2) Measure the variance of peak locations of these client fingerprints using their Signal-to-Noise Ratio. (3) Compute the similarity and honesty metrics using their above definitions to obtain the confidence weight. Algorithm 8 below summarizes the steps to construct α_i for a given client i.

8 Algorithm to Compute Client Confidence Weight

```
▷ Input: Ratio of Channels $\hat{\mathbf{h}}_{\mathbf{il}}$ and SNR
▷ Output: Confidence Weight, $\alpha_i$ for client $i$
▷ Step (1): Measure fingerprints for client $i$
for $l = 1, \ldots, m$ do
    for $\phi \in \{0^\circ, \ldots, 360^\circ\}; \theta \in \{0^\circ, \ldots, 360^\circ\}$ do
        Find $F_{il}(\phi, \theta)$ using a single spin to get $\hat{\mathbf{h}}_{\mathbf{il}}$ (Eqn. 10.1)
    end for
end for
▷ Step (2): Measure variances in peak locations using SNR
$\sigma_\theta^2$ = $\sigma_\phi^2$ = Apply Lemma 10.1 SNR
▷ Step (3): Find honesty, similarity and confidence weight
$\beta_i$ = Apply Defn. 10.3 using $\sigma_\theta^2$, $\sigma_\phi^2$, peaks of $F_{il}$
for $j = \{1, \ldots, c\} \setminus \{i\}$ do
    $\gamma_{ij}$ = Apply Defn. 10.3 using $\sigma_\theta^2$, $\sigma_\phi^2$, peaks of $F_{il}$, $F_{jl}$
end for
$\alpha_i = \beta_i \prod_{j \neq i}(1 - \gamma_{ij})$
```

We now present our main result that solves Problem 1 in the problem statement (Sec. §10.1). The following theorem 10.3.1 and associated lemmas state that the expected α_i's of legitimate nodes approach 1, while those of spoofers approach 0, allowing us to discern them under well-defined assumptions: (A.1) The signal paths are independent. (A.2) Errors in azimuth and polar angles are independent. (A.3) The clients transmit enough packets to emulate a large antenna array (in practice, $25 - 30$ packets per second).[5]

Lemma 10.2 *Let us define* $g(x;0,\sigma^2)$ *as*

$$g(x;0,\sigma^2) = \begin{cases} \sqrt{2\pi} f(x;0,\sigma^2), & \text{if } f(x;0,\sigma^2) < \frac{1}{\sqrt{2\pi}}. \\ 1, & \text{otherwise.} \end{cases} \tag{10.8}$$

Where $f(x;0,\sigma^2)$ *is density function of the Gaussian distribution with mean* 0 *and variance* σ^2*. Then if the random variable* u *is uniformly distributed between and* $(0,2\pi)$ *and the random variable* n *is normally*

[5] This is a mild requirement since $25 - 30$ packets can be transmitted in tens of milliseconds, even at the lowest data rate of 6Mb/s of 802.11n Wi-Fi.

distributed with mean 0 *and variance* σ^2*, then their expectations obey the following:*

$$E[g(u)] \leq \sqrt{\sigma}\left(\frac{\sqrt{2}+\sqrt{\pi}}{\pi}\right) \tag{10.9}$$

$$E[g(n)] \geq 1-\sigma \tag{10.10}$$

Proof of Lemma 10.2: Notice that by the definition of the gaussian distribution $\sqrt{2\pi}f(x;0,\sigma^2) < 1$ occurs only if:

$$\sqrt{2\pi}\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{x^2}{2\sigma^2}} < 1 \tag{10.11}$$

$$\text{i.e., } e^{-\frac{x^2}{2\sigma^2}} < \sigma \tag{10.12}$$

$$x > \sqrt{2}\sigma\sqrt{\ln\frac{1}{\sigma}} := S \tag{10.13}$$

Let us first evaluate $E[g(n)]$. We can write this as:

$$\begin{aligned}
E[g(n)] &= \int_{-S}^{S} f(x;0,\sigma^2)dx + 2\sqrt{2\pi}\int_{-\infty}^{-S}[f(x;0;\sigma^2)]^2dx \\
&\geq \int_{-S}^{S} f(x;0,\sigma^2)dx = \text{erf}\left(\frac{S}{\sigma\sqrt{2}}\right) && (10.14)\\
&\geq 1-\text{erfc}\left(\frac{S}{\sigma\sqrt{2}}\right) && (10.15)\\
&\geq 1-\sigma \quad \text{(From erfc}(x) < e^{-x^2}\text{, Eqn. 10.13)} && (10.16)
\end{aligned}$$

Where erf$(\cdot)$ is the well known Error function and erfc$(\cdot)$ is its complement. The error function is related to the CDF of the normal distribution $F(x;0;\sigma^2)$ as: $F(x) = \frac{1}{2}\left(1-\text{erfc}(\frac{-x}{\sigma\sqrt{2}})\right)$. Similarly, we can evaluate $E[u(n)]$ as:

$$\begin{aligned}
E[u(n)] &= \int_{-S}^{S}\frac{1}{2\pi}dx + 2\sqrt{2\pi}\int_{-2\pi}^{-S}\frac{1}{2\pi}f(x;0;\sigma^2)dx \\
&\leq \frac{2S}{2\pi} + \frac{1}{\sqrt{2\pi}}F(-S,0;\sigma^2) \\
&\leq \frac{S}{\pi} + \frac{1}{\sqrt{2\pi}}\text{erfc}(\frac{S}{\sigma\sqrt{2}}) \\
&\leq \frac{S}{\pi} + \frac{1}{\sqrt{2\pi}}e^{\frac{-S^2}{2\sigma^2}} \quad \text{(from erfc}(x) < e^{-x^2}) \\
&\leq \frac{\sigma}{\pi}\left[\sqrt{2\ln\frac{1}{\sigma}} + \frac{1}{\sqrt{2}}\right] \quad \text{(using Eqn. 10.13)} && (10.17)\\
&\leq \sqrt{\sigma}\left(\frac{\sqrt{2}+\sqrt{\pi}}{\pi}\right) \quad \text{(if } \sigma^2 < 2\pi) && (10.18)
\end{aligned}$$

Eqn. 10.16 and 10.18 together prove the lemma. □

Lemma 10.3 *Assuming a network of* m *routers, and a client* i *which reports either: 1) Its legitimate location given by* $\{(\theta_{il}, \phi_{ri})\}$*; or 2) A fake (spoofer) location chosen uniformly at random. Then the corresponding* *values of β_i, i.e. β_{legit} and β_{spoof} have an expectation given by:*

$$E[\beta_{spoof}] \leq \left[\sqrt{\hat{\sigma}_\theta \hat{\sigma}_\phi} \left(\frac{\sqrt{2} + \sqrt{\pi}}{\pi} \right)^2 \right]^m \tag{10.19}$$

$$E[\beta_{legit}] \geq 1 - m\hat{\sigma}_\theta \hat{\sigma}_\phi \tag{10.20}$$

Where, $\hat{\sigma}_\theta$, $\hat{\sigma}_\phi$ are the variances defined in Lemma 10.1.

Proof of Lemma 10.3: Let us first evaluate the component of β_{legit} the confidence metric of a genuine client-i. Let's assume the direction of this client closest to its true location from the profile is $(\phi_{F_{il}}, \theta_{F_{il}})$. Then, the following results hold:

$$\begin{aligned}
&E[\mathcal{L}_{legit}(F_{1r}|(\phi_{il}, \theta_{il})] \\
&= E[g(\phi_{F_{il}} - \phi_{il}; 0, \hat{\sigma}_\phi^2)]E[g(\theta_{F_{il}} - \theta_{il}; 0, \hat{\sigma}_\theta^2)] \\
&= E[g(m_{il}; 0, \hat{\sigma}_\phi^2)]E[g(n_{il}; 0, \hat{\sigma}_\theta^2)] \\
&\geq (1 - \hat{\sigma}_\theta)(1 - \hat{\sigma}_\phi) \geq 1 - \hat{\sigma}_theta\hat{\sigma}_\phi
\end{aligned} \tag{10.21}$$

Where m_{il}, n_{il} are normally distributed. We can write the analogous metric if the client is an adversary as:

$$\begin{aligned}
&E[\mathcal{L}_{spoof}(F_{1r}|(\phi_{il}, \theta_{il})] \\
&= E[g(\phi_{F_{il}} - \phi_{il}; 0, \hat{\sigma}_\phi^2)]E[g(\theta_{F_{il}} - \theta_{il}; 0, \hat{\sigma}_\theta^2)] \\
&= E[g(u_{il}; 0, \hat{\sigma}_\phi^2)]E[g(v_{il}; 0, \hat{\sigma}_\theta^2)] \\
&\leq \sqrt{\hat{\sigma}_\theta \hat{\sigma}_\phi} \left(\frac{\sqrt{2} + \sqrt{\pi}}{\pi} \right)^2
\end{aligned} \tag{10.22}$$

Where u_{il}, v_{il} are uniformly distributed. Generalizing to m independent routers, we can write:

$$E[\beta_{spoof}] \leq \left[\sqrt{\hat{\sigma}_\theta \hat{\sigma}_\phi} \left(\frac{\sqrt{2} + \sqrt{\pi}}{\pi} \right)^2 \right]^m \tag{10.23}$$

$$E[\beta_{legit}] \geq (1 - \hat{\sigma}_\theta \hat{\sigma}_\phi)^m \geq 1 - m\hat{\sigma}_\theta \hat{\sigma}_\phi \tag{10.24}$$

Eqn. 10.23 and 10.24 prove the above Lemma 10.3. □

Lemma 10.4 *Assume a network of m routers serving a client j. We assume a new client i joins the network, which does not report its location. Without loss of generality, this client: 1) either spoofs client-j, potentially scaling power, or; 2) a randomly located legitimate client. Let γ_{spoof}, γ_{legit} denote the second component γ_{ij} of the confidence metrics measured in either cases. Then the expected value of these confidence metrics are* *bounded by:*

$$E[\gamma_{spoof}] \geq 1 - 2mk\sigma_\theta\sigma_\phi \tag{10.25}$$

$$E[\gamma_{legit}] \leq \left[\sqrt{2\sigma_\theta\sigma_\phi}\left(\frac{\sqrt{2}+\sqrt{\pi}}{\pi}\right)^2\right]^{mk} \tag{10.26}$$

Where, σ_θ, σ_ϕ are the variances defined in Lemma 10.1 that depend on signal-to-noise ratio and k is the number of maxima in the fingerprint of client i.

Proof of Lemma 10.4: Let us first evaluate the component of γ_{legit} the confidence metric of a genuine client-i, in relation to router l and client-j. Let's assume set of k directions of this client $\Phi_{legit,r} = \{\phi_{legit,1}, \ldots, \phi_{legit,k}\}$, $\Theta_{legit,r} = \{\theta_{legit,1}, \ldots, \theta_{legit,k}\}$ are uniformly at random. For the moment, let's consider γ_{legit} pertaining to a legitimate client-i. Then, the following results hold:

$$\begin{aligned}
&E[\mathcal{L}_{legit}(2 \text{ spoofs } 1|F_{1r}, F_{2r})] \\
&= \prod_{p=1}^{k} E[g(\phi_{legit,p} - \phi_{1,p}; 0, 2\sigma_\phi^2)] E[g(\phi_{legit,p} - \phi_{1,p}; 0, 2\sigma_\theta^2)] \\
&= \prod_{p=1}^{k} E[g(u_p; 0, 2\sigma_\phi^2)] \prod_{p=1}^{k} E[g(v_p; 0, 2\sigma_\theta^2)] \\
&\leq \sqrt{2\sigma_\theta\sigma_\phi}^k \left(\frac{\sqrt{2}+\sqrt{\pi}}{\pi}\right)^{2k}
\end{aligned} \tag{10.27}$$

Where u_p, v_p are uniformly at random as well, and κ is a constant normalization factor to ensure the likelihood is at most 1. We can write the analogous metric if client-i is an adversary as:

$$\begin{aligned}
&E[\mathcal{L}_{spoof}(2 \text{ spoofs } 1|F_{1r}, F_{2r}] \\
&= \prod_{p=1}^{k} E[f(\phi_{spoof,p} - \phi_{1,p}; 0, 2\sigma_\phi^2)] E[f(\phi_{spoof,p} - \phi_{1,p}; 0, 2\sigma_\theta^2)] \\
&= \prod_{p=1}^{k} E[f(m_p; 0, 2\sigma_\phi^2)] \prod_{p=1}^{k} E[f(n_p; 0, 2\sigma_\theta^2)] \\
&\geq (1 - \sqrt{2}\sigma_\theta)^k (1 - \sqrt{2}\sigma_\phi)^k \geq 1 - 2k\sigma_\theta\sigma_\phi
\end{aligned} \tag{10.28}$$

Where m_p, n_p are normally distributed. Assuming fingerprints from routers are independent, we can write the expectations across routers in the two cases as:

$$E[\gamma_{spoof}] \geq (1 - 2k\sigma_\theta\sigma_\phi)^m \geq 1 - 2mk\sigma_\theta\sigma_\phi \tag{10.29}$$

$$E[\gamma_{legit}] \leq \left[\sqrt{2\sigma_\theta\sigma_\phi}\left(\frac{\sqrt{2}+\sqrt{\pi}}{\pi}\right)^2\right]^{mk} \tag{10.30}$$

Eqn. 10.30 and 10.29 prove the above Lemma 10.4. □

Theorem 10.3.1 *Consider a network with m servers and c clients. A new client i either: 1) spoofs s clients reporting a random location, potentially scaling power, or; 2) is a uniformly randomly located legitimate* *client. Let α_{spoof}, α_{legit} be the confidence weights in either case. Assume that the client obtains its signals from servers along k paths (where the number of paths k is defined by Eqn. §10.1 in Sec. §10.2). Under A.1-A.3, the expected $\alpha_{spoof}, \alpha_{legit}$ are bounded by:*

$$\begin{aligned} E[\alpha_{spoof}] &\leq \left[\sqrt{\hat{\sigma}_\theta \hat{\sigma}_\phi}\kappa\right]^m [2mk\sigma_\theta\sigma_\phi]^s \\ E[\alpha_{legit}] &\geq 1 - cm\hat{\sigma}_\theta\hat{\sigma}_\phi \left[\sqrt{2\sigma_\theta\sigma_\phi}\kappa\right]^{mk} \end{aligned} \tag{10.31}$$

Where $\kappa = \left((\sqrt{2}+\sqrt{\pi})/\pi\right)^2$, σ_θ, σ_ϕ, $\hat{\sigma}_\theta$, $\hat{\sigma}_\phi$ are the variances defined in Lemma 10.1 that depend on signal-to-noise ratio (the latter include measurement error in reported locations).

Proof of Theorem 10.3.1: Using Lemma 10.4, we can write:

$$\begin{aligned} E[1-\gamma_{spoof}] &\leq 2mk\sigma_\theta\sigma_\phi \\ E[1-\gamma_{legit}] &\geq 1 - \left[\sqrt{2\sigma_\theta\sigma_\phi}\kappa\right]^{mk} \end{aligned}$$

Combining over s spoofers and $c-s$ legitimate clients, we can write:

$$\begin{aligned} E[\prod_{i\neq j} 1-\gamma_{spoof}] &\leq [2mk\sigma_\theta\sigma_\phi]^s \left(1-(c-s)\left[\sqrt{2\sigma_\theta\sigma_\phi}\kappa\right]^{mk}\right) \\ &\leq [2mk\sigma_\theta\sigma_\phi]^s \\ E[\prod_{i\neq j} 1-\gamma_{legit}] &\geq 1 - c\left[\sqrt{2\sigma_\theta\sigma_\phi}\kappa\right]^{mk} \end{aligned}$$

Combining the above equations with Lemma 10.3, we can write:

$$\begin{aligned} E[\alpha_{spoof}] &\leq \left[\sqrt{\hat{\sigma}_\theta \hat{\sigma}_\phi}\kappa\right]^m [2mk\sigma_\theta\sigma_\phi]^s \\ E[\alpha_{legit}] &\geq 1 - cm\hat{\sigma}_\theta\hat{\sigma}_\phi \left[\sqrt{2\sigma_\theta\sigma_\phi}\kappa\right]^{mk} \end{aligned}$$

Which proves the required lemma. □

A natural question one might ask is if the above lemma holds in general environments, where its assumptions A.1-A.3 may be too stringent. Our extensive experimental results in Sec. 10.5 show that our bounds on α approximately predict performance in general environments. Further, Sec. §10.5.1 shows that results from an anechoic chamber, which emulate free-space conditions where the lemma's assumptions can be directly enforced, tightly follow the bounds of Lemma 10.1.

In sum, one can adopt the above lemma to distinguish adversarial nodes from legitimate nodes, purely based on α. However, an interesting alternative is to incorporate α directly into multi-robot controllers to give provable service guarantees to legitimate nodes. The next section show how α_i readily integrates with robotic coverage controllers, in particular.

10.4 Threat-Resistant Distributed Control

This section describes how our spoof detection method from Sec. §10.3 integrates with well-known coverage controllers from [33, 148, 149]. The area coverage problem deals with positioning server

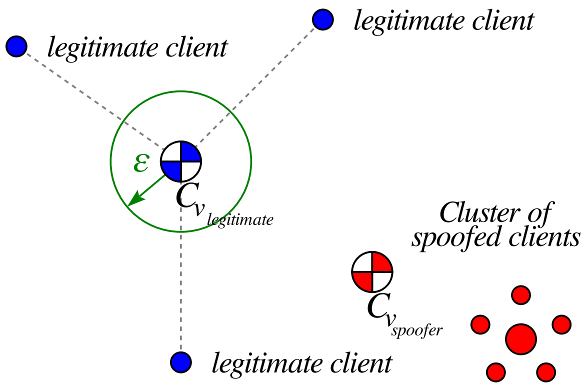


Figure 10-6: Coverage guarantee. An ε ball around the ground-truth centroid, $C_{V_{\text{legitimate}}}$, is shown in green. Theorem 10.4.1 finds $\varepsilon(\mathcal{P})$ so that server positions remain in this ball in the presence of spoofed clients.

robots to minimize their Euclidean distance to certain areas of interest in the environment. These areas are determined by an importance function $\rho(q)$ that is defined over the environment $\mathcal{Q} \subset \mathbb{R}^3$ of size $L(\mathcal{Q})$. For our coverage problem, the peaks of the importance are determined by client positions P, e.g., $\rho(q, P) = \rho_1(q) + \ldots + \rho_c(q)$ where $\rho_i(q)$ quantifies the influence of client i's position on the importance function. Using [33, 148, 149], server robot positions optimizing coverage over $\rho(q, P)$ will minimize their distance to clients.

To account for spoofed clients, we modify the importance function $\rho(q, P)$ using the α_i for each client $i \in [c]$ that is computed by Algorithm 8. E.g., we can multiply each client-term in $\rho(q, P)$ by its corresponding confidence weight: $\rho(q, P)_\alpha = \alpha_1\rho_1(q) + \ldots + \alpha_c\rho_c(q)$. Given the properties of these weights derived in Theorem 10.3.1, i.e., α_i is bounded near zero for a spoofed client and near one for a legitimate client, the effect of multiplication by the α's is that terms corresponding to spoofed clients will be bounded to a small value (see Fig. 10-6); providing resilience to the spoofing attack.

For simplicity, we assume the importance function $\rho(q)$ is static (from [33]) and α's from Algorithm 8 are computed once, at the beginning of the coverage algorithm. We note that our approach readily extends to the adaptive case in [148, 149] when the importance function (and location of clients) change, by having the service robots exchange their learned importance function. This in turn can trigger a re-calculation of α values.

We now show that computed server positions are impacted by spoofers to within a closed-form bound, that depends on problem parameters like signal-to-noise ratio. Theorem 10.4.1 below solves Problem 2 of our problem statement (Sec. §10.1).

Theorem 10.4.1 *Let X be a set of server robot positions and $P = S \cup \tilde{S}$ be a set of client positions where S is the set of spoofed client positions, and $\tilde{S}$ is the set of legitimate clients. The identities of the clients being spoofed is assumed unknown. Let $\{\alpha_1, \ldots, \alpha_c\}$ be a set of confidence weights satisfying Theorem 10.3.1 and assume a known importance function $\rho(q, P) = \rho_1(q) + \ldots + \rho_c(q)$ that is defined over the environment $\mathcal{Q} \subset \mathbb{R}^3$ of size $L(\mathcal{Q})$. Define $C_V = \{x_1^*, \ldots, x_m^*\}$ to be the set of server positions optimized over $\rho(q, \tilde{S})$, i.e., where there are zero spoofed clients and C_{V_α} to be the set of server positions optimized over $\rho(q, P)_\alpha = \alpha_1\rho_1(q) + \ldots + \alpha_c\rho_c(q)$ where there is at least one spoofed client, ie. $|S| \geq 1$. If $\{\alpha_1, \ldots, \alpha_c\}$ satisfy*

Theorem 10.3.1, we have that $\forall x \in C_{V_\alpha}$ there exists a unique $y \in C_V$, where in the expected case dist$(x, y) \leq \varepsilon(m, s, \sigma_\phi, \sigma_\theta, \kappa)$

$$\varepsilon = \max\left\{[\sqrt{\hat{\sigma}_\theta\hat{\sigma}_\phi}\kappa]^m[2mk\sigma_\theta\sigma_\phi]^s, cm\hat{\sigma}_\theta\hat{\sigma}_\phi[\sqrt{2\sigma_\theta\sigma_\phi}\kappa]^{mk}\right\} L(Q)$$

and $m, s, \sigma_\phi, \sigma_\theta, \kappa$ are problem parameters as in Theorem 10.3.1.

Proof: We make an important observation that $E[\alpha_i] \leq a$ if client i is a spoofed node, and $E[\alpha_i] \geq b$ otherwise; hence:

$$\rho(q, P)_\alpha = a(\rho_1(q) + \ldots + \rho_s(q)) + b(\rho_{s+1}(q) + \ldots + \rho_c(q))$$

is the maximal effect that the presence of spoofed clients can have on the importance function. Intuitively, all spoofed clients have a weight of *at maximum* a and all legitimate clients have a reduced weight of *at minimum* b. Using this observation we can bound the influence of the spoofed clients on computed server control inputs (see Fig. 10-6). Specifically, recall from [33] that the position control for each server is: $u_l = -2M_V(C_V - c_l)$, where $M_V = \int_V \rho(q)dq$, $C_V = \frac{1}{M_v}\int_V q\rho(q)dq$ and V is the voronoi partition for server l defined as all points $q \in \mathcal{Q}$ with dist$(q, x_l) <$ dist(q, x_g) where $g \neq l$. Using the importance function from above we can write $C_{V_\alpha} = \frac{1}{M_{V_\alpha}}(aC_{V_S} + bC_{V_L})$ where C_{V_S} is the component of the centroid computed over spoofed nodes and C_{V_L} is the component of the centroid computed over legitimate nodes and M_{V_α} is defined shortly. We rewrite C_{V_S} as a perturbation of the centroid over legitimate nodes as $C_{V_S} = C_{V_L} + \vec{v}\|\vec{e}\|$ where $\vec{v}$ is an arbitrary unit vector and the magnitude of $\vec{e}$ can be as large as the length of the operative environment,$\|\vec{e}\| \leq L(Q)$. Let the total mass be $T = M_{V_s} + M_{V_L}$. We can write a similar expression for the mass M_{V_α} using the bounds a and b as $M_{V_\alpha} = bT + (a-b)M_{V_L}$. Substituting these expressions into C_{V_α} and simplifying gives $C_{V_\alpha} = \frac{C_{V_L} + b\vec{v}\|\vec{e}\|}{bT + (a-b)M_{V_L}}$. Combining this expression with the server control input:

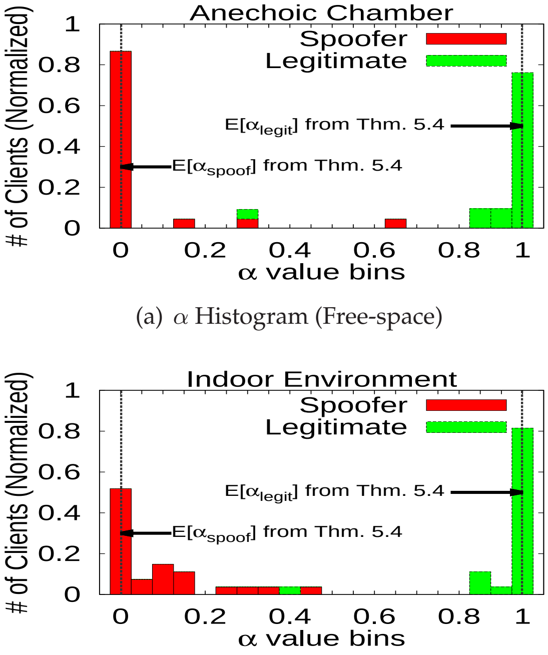
$$u_l = k\,(\;[(a+b)C_{V_L} - p_l] + b\|\vec{e}\|\vec{v}\;) \tag{10.32}$$

Where $k = -2(bT + aM_{V_L})$. If $(a+b) = 1$, this control input drives the server robot l to a neighborhood of size $\varepsilon = b\|\vec{e}\| \leq bL(\mathcal{Q})$ centered around the centroid C_L defined over the legitimate clients. So if $b = \max\{[\sqrt{\hat{\sigma}_\theta\hat{\sigma}_\phi}\kappa]^m[2mk\sigma_\theta\sigma_\phi]^s, cm\hat{\sigma}_\theta\hat{\sigma}_\phi[\sqrt{2\sigma_\theta\sigma_\phi}\kappa]^{mk}\}$ from Theorem 10.3.1 Equation (10.31), then:

$$\varepsilon = \max\left\{[\sqrt{\hat{\sigma}_\theta\hat{\sigma}_\phi}\kappa]^m[2mk\sigma_\theta\sigma_\phi]^s, cm\hat{\sigma}_\theta\hat{\sigma}_\phi[\sqrt{2\sigma_\theta\sigma_\phi}\kappa]^{mk}\right\} L(\mathcal{Q})$$

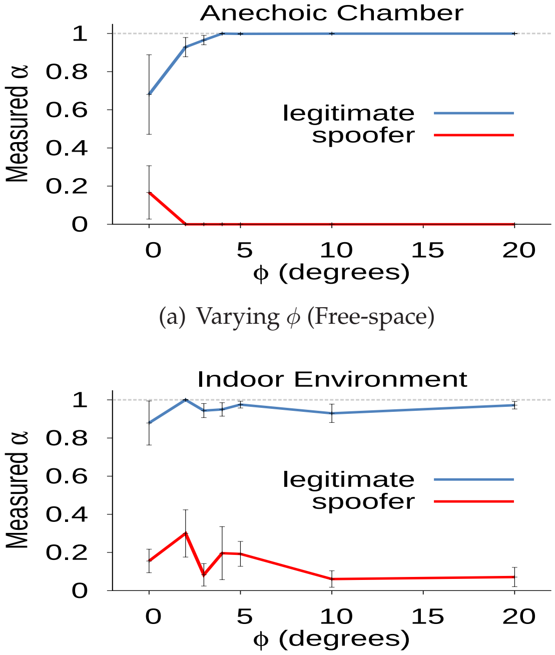
then we have $(a+b) = 1$ as desired, proving the lemma. □

10.5 Experimental Results



(a) α Histogram (Free-space)

(b) α Histogram (Multipath)



(a) Varying ϕ (Free-space)

(b) Varying ϕ (Multipath)

Figure 10-8: Experimental Evaluation of α: (a) In an anechoic chamber approximating our assumptions A.1-A.3 (§10.3.1), α largely agrees with theory. (b) In a typical multipath environment, experimental results closely follow theoretical predictions. Data shows that $\alpha = 0.5$ is a good threshold value.

Figure 10-9: Co-Aligned Clients: We vary the angle ϕ between a legitimate and malicious client, relative to a single server, and plot α in (a) an anechoic chamber and (b) an indoor environment. The minimum ϕ needed to distinguish the clients is only: (a) 3° in freespace, (b) 0° in multipath settings.

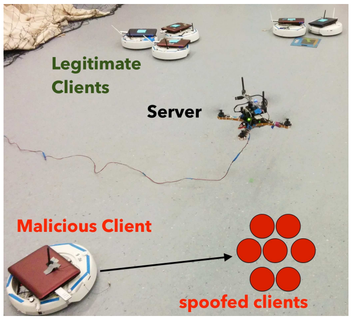
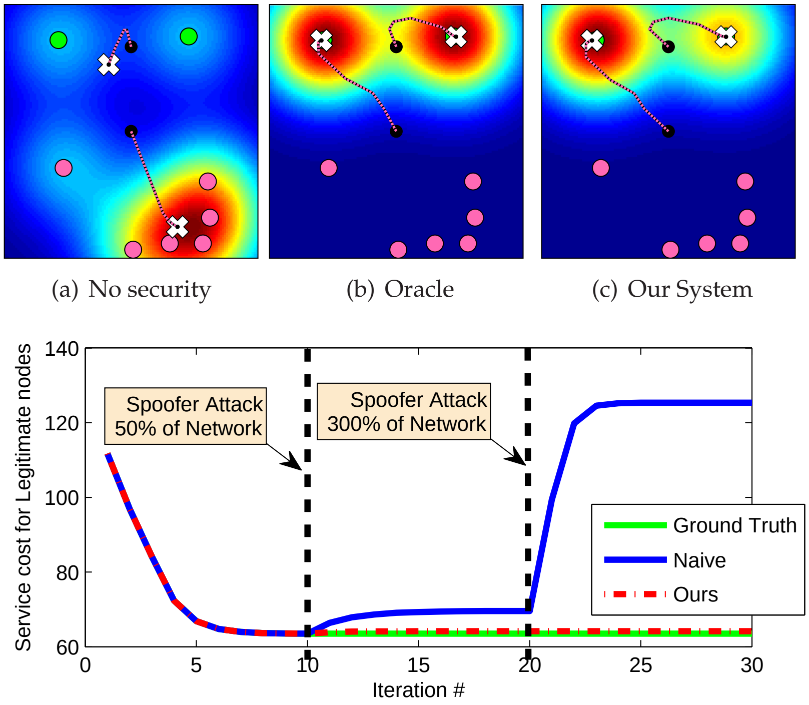


Figure 10-7: Testbed Snapshot

This section describes our results from an experimental evaluation of our theoretical claims. Our aerial servers were implemented on two AscTec Atomboard computing platforms equipped with Intel 5300 Wi-Fi cards with two antennas each, mounted on two AscTec Hummingbird quadrotors. Our clients were ten iRobot Create robots, each equipped with Asus EEPC netbooks and single-antenna Wi-Fi cards. An adversarial client forged multiple identities by spawning multiple packets containing different identities (up to 75% of the total number of legitimate clients in the system), and could use a different transmit power for each identity. The adversary advertised identities by modifying the Wi-Fi MAC field, a common technique for faking multiple identities [152].

Evaluation: . We evaluate our system in two environments: *1)* An indoor multipath-rich environment with walls and obstacles equipped with a Vicon motion capture system to aid quadrotor



(a) No security (b) Oracle (c) Our System

(d) Cost

Figure 10-10: Experimental Results for Sybil Attack in Multi-Agent Coverage. Depicts the total distance of converged quadrotor server positions (white ×) to two legitimate clients and six spoofed clients. We consider: (a) an insecure system where each spoofed client creates a false peak in the importance function, (b) a ground truth importance function, and (c) our system where applying α weights from Algorithm 8 recovers the true importance function. (d) Depicts a ground-truth cost computed with respect to legitimate clients as Sybil nodes dynamically enter the network. Our system (red dotted line) performs near-optimal even when spoofed clients comprise more than twice the network.

navigation; *2)* An anechoic chamber to emulate a free-space setting that is particularly challenging to our system. We estimated the average theoretical expected standard deviation to be $\sigma_\theta, \sigma_\phi$ of $0.7°$ (Lemma 10.1). After including the standard deviation in reported location, based on the known errors of our localization framework, this increased the average $\hat{\sigma}_\theta, \hat{\sigma}_\phi$ by $2°$(variances in each experiment depend on measured SNR) We compare our system against a baseline that uses a Received Signal Strength (RSSI) comparison (akin to [138]).

Roadmap: . We conduct three classes of experiments: (1) Microbenchmarks to validate our client confidence metric, both in free-space and multipath indoor environments (Sec. §10.5.1). (2) Experiments applying this confidence metric to quarantine adversaries (Sec. §10.5.2). (3) Application of our system to secure the coverage problem against Sybil attacks (Sec. §10.5.3).

■ 10.5.1 Microbenchmarks on the Confidence Metric

This experiment studies the correctness of our system's confidence metric α. Recall from theory in §10.3 that α's measured by a server robot distinguish between unique clients based on their diverse physical directions and the presence of multipath reflections. Thus, a free-space environment (i.e., with no multipath) is particularly challenging to our system.

Method. To approximate free-space, we measured α values in a radio-frequency anechoic chamber which attenuates reflected paths by about 60 dB, for a legitimate and malicious client from one server robot 12 m away. Next, in a 10 m x 8 m indoor room (a typical multipath case), we measured α's from one server for up to ten legitimate clients and ten spoofed clients.

Results. In Fig. 10-8, the values of α in the anechoic chamber tightly follow our theoretical bounds in Theorem 10.3.1 (Fig. 10-9(c)). As expected, our results in indoor multipath environments exhibit a larger variance but follow the trend suggested by theory. Further, we stress our confidence metric by isolating the case of colinearity in both environments. In Fig. 10-9, we consider a spoofing adversary initially co-aligned with a legitimate client, and measure α as the angle of separation, ϕ, is increased from $0°$ to $20°$ relative to the server robot. In the anechoic chamber at ϕ close to $0°$, the fingerprints of both the legitimate and adversarial nodes are virtually identical, each with precisely one peak at $0°$. Consequently, α for the legitimate node is much below 1, indicating that is believed to be adversarial (i.e., the term $1 - \gamma$ in α approaches 0 in Eqn. 10.2). However, α for the legitimate client quickly approaches 1, even if $\phi = 3°$ in the anechoic chamber. In fact, α is virtually identical to 1 beyond $10°$, indicating that a single server robot can distinguish closely aligned legitimate and adversarial clients even in free-space. Fig. 10-9(b) shows that multipath can distinguish clients even at $\phi = 0°$, due to additional reflected paths that help disambiguate these clients.

■ 10.5.2 Performance of Sybil Attack Detection

In this experiment, we measure our system's classification performance on legitimate and spoofed clients, in the presence of static, mobile, and power-scaling adversaries.

Method. This experiment was performed in the multipath-rich indoor testbed with walls and obstacles. Each run consisted of one quadrotor server, and (randomly positioned) ten control clients, or nine legitimate clients with an adversary reporting two to nine spoofed clients. Each Sybil attack was performed under three modalities: (1) a stationary attacker with a fixed transmission power, (2) a mobile attacker (random-walk and linear movements), and (3) an attacker scaling the per-packet power by a different amount for each spoofed client, from 1 to 31 mW. The quadrotor server classifies clients with an $\alpha < 0.5$ as spoofed (see Fig. 10-8). The baseline RSSI classifier uses a $2\ dB$ thresholded minimum dissimilarity, a technique previously applied in static networks [138, 180].

Results. For each modality, our performance against an RSSI baseline over multiple network topologies is summarized here as true positive rates (TPR) and false positive rates (FPR). In particular, our classifier is robust to power-scaling Sybil attacks (where RSSI performs poorly) since we use the ratio of wireless channels in computing α (Sec. §10.2). Our client classifier exhibits consistent performance in both power-scaling and mobile scenarios with a TPR $\approx 96\%$ and FPR $\approx 4\%$.

	Our System		RSSI	
	TPR	FPR	TPR	FPR
Static	96.3	3.0	81.5	9.1
Mobile	96.3	6.1	85.2	6.1
Δ mW	100.0	3.0	74.1	27.3

■ 10.5.3 Application to Multi-Agent Coverage

We implement the multi-agent coverage problem from [33], where a team of aerial servers position themselves to minimize their distance to client robots at reported positions p_i,$i \in [c]$. We use an importance function $\rho(q, P) = \rho_1(q) + \ldots + \rho_c(q)$ defined in Sec. §10.4 where each client term is a Gaussian-shaped function $\rho_i(q) = \exp(-\frac{1}{2}(q - p_i)^T(q - p_i))$ (Fig. 10-10(b)). An α-modified importance function is implemented as $\rho(q, P)_\alpha = \alpha_1\rho_1(q) + \ldots + \alpha_c\rho_c(q)$ where the α terms are computed using Algorithm 8 (Fig. 10-10(c)).

Method. This experiment was performed in the multipath-rich indoor testbed. For each experiment we randomly place three clients in an 8 m x 10 m room with two AscTec quadrotor servers. Fig. 10-10(a)-(c) shows one client-server topology where an adversary spoofs six Sybil clients. Upon convergence, we measure the distance of each server from an optimal location in 3 scenarios: *1)* a naive system with no security, *2)* an oracle which discards Sybil clients *a priori*, and *3)* our system.

Results. Fig. 10-10(a)-(c) depicts the converged locations for a candidate topology in the above three scenarios. We observe that by incorporating α weights in our controller, our system approximates oracle performance. Fig. 10-10(d) demonstrates the ability of our system to bound the service cost to near optimal even as spoofers enter the network (comprising up to 300%).

Aggregate Results. Across multiple topologies and 12 runs, with no security the maximum distance from each quadrotor to an oracle solution is on average 3.77 m (stdev: 0.86). Our system achieves a 0.02 m (stdev: 0.02) average from oracle.

■ 10.6 Related Work

The problem of Sybil attacks has been studied in general multi-node, often static, networks, and many tools have been developed for these settings. Past work falls under three categories: (1) Cryptographic authentication schemes can be used to prevent Sybil attacks (Table 7 in [182]). These require trusted central authorities and computationally expensive distributed key management, to account for dynamic clients that enter and leave the network [182]. (2) Non-cryptographic techniques in the wireless networking community leverage wireless physical-layer information to detect spoofed client identities or falsified locations [81, 194, 189, 192]. These rely on bulky and expensive hardware like large multi-antenna arrays, that cannot be mounted on small robotic platforms. (3) Recent techniques have attempted to use wireless signal information like received signal strength (RSSI) [180, 138] and channel state information [105]. Such techniques need clients to remain static, since mobility can cause wireless channels to fluctuate rapidly [10]. In addition, they are susceptible to power-scaling attacks, where clients scale power differently to imitate different users. In sum, the above systems share one or more of the following characteristics making them ill-suited to multi-robot networks: (1) require computationally-intensive key management; (2) rely on bulky and expensive hardware; (3) assume static networks. Indeed past work has highlighted the gravity and apparent sparsity of solutions to cyber-security threats in multi-robot networks [72, 146, 25].

Unlike past work, our solution has three attributes that particularly suit multi-robot networks. (1) It captures physical properties of wireless signals and therefore does not require distributed key management. (2) It relies on cheap commodity Wi-Fi radios, unlike hardware-based solutions [189, 194]. (3) It is robust to client mobility and power-scaling attacks.

Finally, our system builds on Synthetic Aperture Radar (SAR) to construct signal fingerprints [50]. SAR has been widely used for radar imaging [50, 85] and indoor positioning [89, 88, 179, 58]. In contrast, this chapter builds upon SAR to provide cyber-security to multi-robot networks. In doing so, it provides theoretical security guarantees that are validated experimentally. These integrate readily with performance guarantees of existing multi-robot controllers, like the well-known robotic coverage controllers [33, 148] as shown in Sec. §10.4.

■ 10.7 Discussion

In this chapter, we develop a new system to guard against the Sybil attack in multi-robot networks. We derive theoretical guarantees on the performance of our system, which are validated experimentally. While this chapter has focused on coverage, it can be readily extended to secure other multi-robot controllers against Sybil attacks, e.g., unmanned delivery [90], search-and-rescue [100], and formation control [181]. We note for future work that our method of detecting spoofed clients is applicable to servers as well, since they also communicate wirelessly. Since our approach is based on the fundamental physics of wireless signals, we believe that it will easily generalize beyond Sybil attacks to other Wi-Fi based security issues in robot-swarms such as packet path validation [106] and detecting packet injection attacks to name a few.

Chapter 11
Conclusion, Lessons Learned and Future Work

In conclusion, this dissertation address the key challenges and opportunities of modern wireless networks: combating wireless interference and providing a new indoor positioning service. Unfortunately, past work that has that looked into these challenges either require complete overhaul of wireless infrastructure, or do not provide satisfying performance. This dissertation addresses this dilemma, presenting multiple systems that both combat interference and provide accurate indoor positioning without needing overhaul of the infrastructure. It develops systems that manage wireless interference by manipulating wireless transmissions at the signal-level, even on commodity Wi-Fi radios. Further, it enables novel indoor positioning services for both Wi-Fi and LTE networks, that achieve high accuracy while requiring low deployment effort. Finally, it leverages these tools to open up new connections between wireless networking and robotics, for improved communication and security in multi-robot networks. Specifically, we make the following contributions:

- *Bringing Multi-Antenna Interference Management to Today's Wireless LANs*: OpenRFis the first system that enables the deployment of physical-layer MIMO techniques on commodity Wi-Fi cards. It is also the first cross-layer design that is demonstrated using a fully operational network stack with real applications.

- *Interference Alignment by motion:* We describe MoMIMO, a technique that demonstrated, for the first time, that interference alignment and nulling can be achieved, even with single-antenna transmitters, by simply moving the receive antenna. We also showed that the amount of antenna displacement needed is fairly small ($\sim$ one inch); hence, MoMIMO can be achieved by sliding antennas.

- *LTE Radio Analytics made Easy and Accessible:* We present LTEye, the first open platform to provide fine-grained temporal and spatial analytics on LTE radio performance, without private user information or provider support. LTEye employs a novel extension of synthetic aperture radar to communication signals to accurately localize mobile users, despite the presence of multipath. We empirically evaluate LTEye on software radios and provide deep insights on the LTE PHY and highlight shortcomings such as inter-cell interference and inefficient spectrum utilization.

- *Accurate indoor positioning with Zero Startup Cost:* We design Ubicarse, an indoor localization system to achieves tens of centimeters of accuracy on commodity mobile devices, without requiring specialized infrastructure or fingerprinting. Ubicarse's provides a new formulation of Synthetic Aperture Radar that allows mobile devices to emulate an antenna array; where this formulation is tolerant to unknown translation of the antenna center of up to half a meter. We implement Ubicarse on a commodity tablet and demonstrate tens of centimeter accuracy in both device localization and object geotagging in complex indoor settings.
- *Adaptive Communication in Multi-Robot Systems Using Directionality of Signal Strength:* We present a novel method to position a team of robotic routers to satisfy the heterogeneous communication demands of a network of robotic clients, while adapting to real-time environmental changes. At the heart of this system is a new approach for routers to determine the direction of movement that best improves its signal to any given client in the network. We implement our design and demonstrate its empirical gains in a testbed of iRobot Create robots equipped with commodity Wi-Fi radios.
- *Guaranteeing Spoof-Resilient Multi-Robot Networks:* We present a new approach that secures multi-robot networks against the Sybil attacks that is applicable to a broad class of problems in distributed robotics. We prove that the influence of spoofers is analytically bounded under our system in a coverage context, where each robotic node providing coverage remains within a bounded radius of its position in the absence of an attack. Our theoretical results are validated extensively through experiments using aerial and ground robots in diverse settings.

■ 11.1 Lessons Learned

This dissertation presents contributions that span multiple topics in the field of wireless communication ranging from improving communication quality, new RF-based services and connections with the field of robotics. Developing the systems in this thesis involved collaborations with experts in diverse fields: security, robotics, control theory, and signal processing. Such inter-disciplinary collaboration led to both important research contributions as well as several lessons learned:

Look at the Network as a Whole. Often, networking research tends to focus on layers – either the physical layer, the MAC, congestion control, etc. However, looking at how layers inter-operate to achieve network objectives as a whole can be extremely valuable. For instance, prior to OpenRF, there were several important solutions to combat interference that operated purely at the physical layer. Yet, these failed to answer the questions that operators truly care about – Will TCP continue to work? How will these systems impact the delays of video traffic? It is indeed these questions that led us to design OpenRF, which demonstrated for the first time that one can combat interference to achieve performance gains for the network as a whole: for real applications such as TCP file transfers and video streaming.

A similar example can be seen in the context of cellular networks. LTEye observed inefficiencies at the physical layer that may perhaps make sense in isolation, but not in the context of higher layers. For instance, the physical layer allots equal resources for uplink and downlink traffic. Yet in reality, downlink traffic far exceeds uplink traffic by an order of magnitude, leading to much spectrum that lies wasted.

Bring solutions to real networks. Wireless research has benefited immensely from innovative signal processing algorithms. The traditional approach has been to demonstrate these solutions on software radios. Yet, bringing these solutions to real networks requires operating with the constraints of today's Wi-Fi radios and standards.

OpenRF develops innovations in the context of bringing MIMO interference management techniques to commercial wireless networks. Ubicarse, on the other hand, brings radar technologies to commercial handheld devices. In both cases, new algorithms needed to be designed. OpenRF required operation with existing Wi-Fi standards and the full Wi-Fi stack. Ubicarse needed new formulations of radar algorithms that can handle trajectories of the user devices that are not mechanically controlled. Bringing solutions to real networks can be challenging, yet make high impact in bridging wireless research with the networks of today.

Wireless radio as a "sensor". Traditionally, wireless networking research has focused on communication. Yet, wireless radios can do much more – their signals capture physical properties of the transmitter as well as the environment around them. In other words, wireless radios are effectively "sensors".

The research in this dissertation has exploited this observation in a variety of contexts: Robots can sense how wireless signals reflect around obstacles to navigate effectively. Or they one can sense a malicious transmitter that pretends to be multiple fake nodes. Indeed, both Ubicarse and LTEye are essentially sensing different paths that wireless signals take through the environment to isolate the true location of a wireless transmitter.

The enormous potential of wireless networking and robotics: . Wireless networking does not live in an island – it serves real applications in a variety of contexts. However, the traditional approach to design wireless protocols is to be generic – serve as many applications as possible while functioning efficiently. Yet, knowledge of the application concerned can lead to much more efficient networks.

We discovered a particularly huge opportunity to do this in the context of robotic networks. Robotic networks are by definition highly mobile with this mobility being highly controllable. This allowed us to design new wireless communication protocols to exploit this mobility – both to improve communication speeds, and open up new services. We believe we have only scratched the surface in this endeavor and there are several more opportunities for wireless networking protocols to be re-designed in the context of the applications they serve.

■ 11.2 Future Work

Wireless communication is at an exciting time, expanding to a variety of applications creating new challenges and opportunities. Indeed there remain several opportunities to draw on a deep understanding of the wireless physical layer to advance next-generation mobile systems and applications. A particularly important direction that builds upon the work in this dissertation are in the area of the Internet of Things and machine-to-machine communication. As diverse devices and platforms connect to the Internet, the vision of Internet of Things is upon us. Wireless technologies will play a crucial role in this vision, given that they are cheap, low-power, and most importantly - mobile. Yet, building the Internet of Things requires innovation in several areas that we can explore:

■ 11.2.1 Location as a Context

In the Internet of Things, device locations are important to identify context. For example, a television can turn on automatically when the user sits on the couch; or, a factory robot can find different parts based on their location. The main challenge is to find simple and cheap solutions that leverage existing infrastructure, without requiring extensive fingerprinting of the deployment space.

Solutions in this space, much like Ubicarse, must enlist wireless channels. Fortunately, in the Internet of Things, wireless devices will be everywhere - many at fixed locations, like lamps and thermostats. This creates new opportunities for collaborative indoor positioning. For instance, consider systems that synchronize many distributed wireless devices to emulate large antenna arrays. Antenna arrays are widely used in accurate positioning systems, but are restricted to be small owing to practical limits on size and cost. In contrast, many low-cost wireless devices in the Internet of Things can collaborate to form large antenna arrays. However, unlike typical antenna arrays, such devices may not be arranged in regular shapes like lines or circles, which necessitates new antenna array algorithms. Further, these devices may not be located at well-known positions, which calls for means to "auto-calibrate" the system.

■ 11.2.2 Built-in Security

For any system to be truly secure, security must be an inherent aspect of system design and not merely an after-thought. Finding robust and scalable security solutions for the Internet of Things is challenging, given the sizable number of devices involved. A unique challenge of interest is spoofing, where a single entity pretending to be a large number of devices can alter system behavior. For example, consider Google's Project Loon that plans to expand the Internet to a large number of devices in remote areas using aerial routers that gravitate towards network demand. Malicious clients can spawn thousands of devices, drawing away network coverage.

A promising solution to guard against such attacks relies on wireless signal characteristics. LTEye demonstrates that wireless signals from a source contain information about its physical location, as well as objects in the environment. When processed carefully, this information can serve as unique fingerprints that map solely to a given device. Such a system would be invaluable in detecting large groups of fake spawned devices, since these devices will map to the same physical fingerprint.

■ 11.2.3 Architecture Integrating Diverse Platforms

Diverse wireless technologies are point-solutions targeting specific applications. Seamless communication between them requires new standards, much like what TCP/IP did for the Internet. Designing a cohesive architecture that integrates different wireless platforms at the physical layer for the Internet of Things is a major problem I plan to attack in the future.

Such an architecture must remain simple and scalable, given that the Internet of Things will primarily contain a large number of small devices. Much like OpenRF, it must allow for easy coordination, so that several simple devices can collaborate to perform complex tasks. For example, low-power sensors can synchronize their transmissions and relay data to higher-power Wi-Fi access points. The architecture must also be programmable at the level of sub-systems, as opposed to individual entities. For instance, one should be able to program the heating system of an entire room as opposed to commanding vents individually.

References

[1] Amazon prime air. (Cited on page 231.)

[2] IEEE 802.11n-2009 (Section 9.10). http://standards.ieee.org/findstds/standard/802.11n-2009.html. (Cited on pages 33, 34, 45, 69, 72 and 76.)

[3] Linux Advanced Routing and Traffic Control. lartc.org. (Cited on page 152.)

[4] VICON T-Series. http://www.vicon.com. VICON. (Cited on page 183.)

[5] 3gpp. Radio Measurement Collection for Minimization of Drive Tests (MDT). http://www.etsi.org/d eliver/etsi_ts/137300_137399/137320/11.03.00_60/ts_137320v110300p.pdf. (Cited on pages 34 and 157.)

[6] 3rd Generation Partnership Project. E-UTRA, Physical Channels and Modulation, TS 36.211, v8.9.0. 2010. (Cited on pages 31, 132 and 151.)

[7] 3rd Generation Partnership Project. E-UTRA, Radio Resource Control, TS 36.331, v8.9.0. Apr 2010. (Cited on pages 133 and 135.)

[8] 3rd Generation Partnership Project. E-UTRA, v8.8.0, Multiplexing and Channel Coding, TS 36.212. 2010. (Cited on page 136.)

[9] F. Adib and D. Katabi. See through walls with wi-fi. In *SIGCOMM*, 2013. (Cited on page 206.)

[10] F. Adib, S. Kumar, O. Aryan, S. Gollakota, and D. Katabi. Interference Alignment by Motion. MOBICOM, 2013. (Cited on pages 7, 11, 46 and 251.)

[11] M. Agrawal and K. Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive gps. ICPR. 2006. (Cited on pages 180 and 189.)

[12] Apple. iBeacon. support.apple.com/kb/HT6048. (Cited on page 125.)

[13] R. Arkin and T. Balch. Line-of-sight constrained exploration for reactive multiagent robotic team. *AMC*, July 2002. (Cited on pages 49 and 117.)

[14] E. Aryafar, N. Anand, T. Salonidis, and E. Knightly. Design and experimental evaluation of multi-user beamforming in wireless LANs. In *MOBICOM*, 2010. (Cited on pages 48 and 82.)

[15] B. Ayvazian. LTE Deployment Strategies: Network Overlay vs. Single RAN. *White Paper: UBM Tech*, 2013. (Cited on page 135.)

[16] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: Mobile phone localization via ambience fingerprinting. MobiCom, 2009. (Cited on pages 124 and 189.)

[17] M. Bansal et al. Openradio: A programmable wireless dataplane. In *HOTSDN*, 2012. (Cited on pages 49 and 83.)

[18] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson. Decentralized cooperative aerial surveillance using fixed-wing miniature uavs. *Proceedings of the IEEE*, 94(7):1306–1324, July 2006. (Cited on pages 194 and 231.)

[19] D. Bertsekas and J. Tsitsiklis. *Introduction to Probability*. Athena Scientific books. Athena Scientific, 2002. 5.4: The Central Limit Theorem. (Cited on page 239.)

[20] F. Bowman. *Introduction to Bessel Functions*. Dover Publications, 2012. (Cited on pages 170, 171 and 174.)

[21] V. Cadambe and S. Jafar. Interference alignment and spatial degrees of freedom for the k user interference channel. In *IEEE ICC*, 2008. (Cited on page 116.)

[22] Samsung wireless smartcam. http://www.samsungsv.com/Model/Detail/26/SNH-1011N. (Cited on page 91.)

[23] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: taking control of the enterprise. SIGCOMM, 2007. (Cited on pages 44 and 82.)

[24] R. Chandra et al. A case for adapting channel width in wireless networks. SIGCOMM, 2008. (Cited on pages 48 and 83.)

[25] A. Chapman, M. Nabi-Abdolyousefi, and M. Mesbahi. Identification and infiltration in consensus-type networks. *1st IFAC Workshop on Estimation and Control of Networked Systems*, 2009. (Cited on page 251.)

[26] N. Chatzipanagiotis, Y. Liu, A. Petropulu, and M. M. Zavlanos. Controlling groups of mobile beamformers. In *IEEE CDC*, 2012. (Cited on pages 49 and 117.)

[27] H.-C. Chen, T.-H. Lin, H. Kung, C.-K. Lin, and Y. Gwon. Determining rf angle of arrival using cots antenna arrays: A field evaluation. In *MILCOM*, 2012. (Cited on page 222.)

[28] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. MobiCom, 2010. (Cited on pages 125, 159 and 189.)

[29] J. Choi et al. Achieving single channel, full duplex wireless communication. MobiCom, 2010. (Cited on pages 48, 51 and 82.)

[30] S. D. Co. Space data skysat: balloon powered communication for military and rural areas, 2014. (Cited on page 197.)

[31] B. Conley. M2M Implications of the 4G LTE Buildout. *Remote Magazine*, 2013. (Cited on page 128.)

[32] I. Constandache, R. Roy, and C. I. Rhee. Compacc: Using mobile phone compasses and accelerometers for localization. In *In INFOCOM*, 2010. (Cited on page 189.)

[33] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. In *IEEE Transactions of Robotics and Automation*, 2004. (Cited on pages 40, 192, 193, 194, 195, 197, 198, 200, 220, 229, 231, 232, 234, 244, 245, 246, 250 and 251.)

[34] J. Cortes et al. Spatially-distributed coverage optimization and control with limited-range interactions. In *ESAIM*, 2004. (Cited on page 197.)

[35] E. Craparo, J. How, and E. Modiano. Throughput optimization in mobile backbone networks. *Mobile Computing, IEEE Transactions on*, 1, 2011. (Cited on page 227.)

[36] M. Crovella et al. Heavy-tailed probability distributions in the world wide web. *A practical guide to heavy tails*, 1:3–26, 1998. (Cited on page 75.)

[37] M. De Gennaro and A. Jadbabaie. Decentralized control of connectivity for multi-agent systems. In *Decision and Control, 2006 45th IEEE Conference on*, 2006. (Cited on page 227.)

[38] L. Deek et al. The impact of channel bonding on 802.11n network management. CoNEXT, 2011. (Cited on page 83.)

[39] P. Djukic and P. Mohapatra. Soft-tdmac: A software tdma-based mac over commodity 802.11 hardware. In *INFOCOM*, 2009. (Cited on pages 49 and 83.)

[40] J. Douceur. The sybil attack. In P. Druschel, F. Kaashoek, and A. Rowstron, editors, *Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer Berlin Heidelberg, 2002. (Cited on page 231.)

[41] USRP N210. http://www.ettus.com. Ettus Inc. (Cited on pages 107, 128 and 144.)

[42] Eurecom. OpenAirInterface. http://www.openairinterface.org/. (Cited on pages 125 and 156.)

[43] Federal Communications Commission. Enabling Innovative Small Cell Use In 3.5 GHZ Band NPRM & Order. FCC 12-148, 2012. (Cited on page 128.)

[44] D. Feldman, S. Gil, R. Knepper, B. Julian, and D. Rus. K-robots clustering of moving sensors using coresets. In *ICRA*, 2013. (Cited on pages 216 and 227.)

[45] R. Fenby. Limitations on directional patterns of phase-compensated circular arrays. *REE*, 1965. (Cited on pages 140, 163 and 176.)

[46] B. Ferris et al. WiFi-SLAM using gaussian process latent variable models. IJCAI, 2007. (Cited on page 189.)

[47] J. Fink, A. Ribeiro, and V. Kumar. Robust control for mobility and wireless communication in cyber-physical systems with application to robot teams. *Proceedings of the IEEE*, 1, 2012. (Cited on pages 193, 198, 199 and 226.)

[48] J. Fink, A. Ribeiro, and V. Kumar. Robust control of mobility and communications in autonomous robot teams. *Access, IEEE*, 1, 2013. (Cited on page 227.)

[49] J. Fink, A. Ribeiro, V. Kumar, and B. Sadler. Optimal robust multihop routing for wireless networks of mobile micro autonomous systems. In *MILCOM*, 2010. (Cited on page 203.)

[50] P. J. Fitch. *Synthetic Aperture Radar*. Springer, 1988. (Cited on pages 121, 122, 123, 125, 130, 139, 140, 163, 168, 173, 189, 193, 199, 206, 228, 236 and 251.)

[51] FlexNets Group. Open-Source Long-Term Evolution (LTE) Deployment (OSLD). https://sites.google.com/site/osldproject/. (Cited on pages 125 and 156.)

[52] S. M. for Wireless Week. Skysites move closer up to reality firm's weather balloons rise to fill gaps in rural areas, 2014. (Cited on page 197.)

[53] H. Foroosh, J. Zerubia, and M. Berthod. Extension of phase correlation to subpixel registration. *Image Processing, IEEE Transactions on*, 2002. (Cited on page 177.)

[54] H. Gazzah and S. Marcos. Directive antenna arrays for 3d source localization. In *Signal Processing Advances in Wireless Communications, 2003. SPAWC 2003. 4th IEEE Workshop on*, pages 619–623, June 2003. (Cited on pages 238 and 239.)

[55] H. Gazzah and S. Marcos. Cramer-Rao bounds for antenna array design. *IEEE Transactions on Signal Processing*, 54:336–345, 2006. (Cited on pages 238 and 239.)

[56] J. Georgy et al. Modeling the stochastic drift of a mems-based gyroscope in gyro/odometer/gps integrated navigation. *IEEE ITS*, 2010. (Cited on pages 165 and 176.)

[57] S. Gil, D. Feldman, and D. Rus. Communication coverage for independently moving robots. In *IROS*, 2012. (Cited on pages 215, 216 and 227.)

[58] S. Gil, S. Kumar, D. Katabi, and D. Rus. Adaptive Communication in Multi-Robot Systems Using Directionality of Signal Strength. IJRR, 2015. (Cited on pages 7, 12 and 251.)

[59] S. Gil, S. Kumar, M. Mazumder, D. Katabi, and D. Rus. Guaranteeing spoof-resilient multi-robot networks. *RSS*, 2015. (Cited on pages 7 and 12.)

[60] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005. (Cited on pages 195, 198, 216, 226 and 232.)

[61] S. Gollakota, F. Adib, D. Katabi, and S. Seshan. Clearing the rf smog: making 802.11n robust to cross-technology interference. In *SIGCOMM*, 2011. (Cited on pages 48, 51, 70, 82 and 116.)

[62] S. Gollakota and D. Katabi. ZigZag Decoding: Combating Hidden Terminals in Wireless Networks. In *SIGCOMM*, 2008. (Cited on page 34.)

[63] S. Gollakota, S. Perli, and D. Katabi. Interference Alignment and Cancelation. In *ACM SIGCOMM*, 2009. (Cited on pages 34, 35, 36, 45, 48, 51, 52, 56, 57, 70, 71, 75, 82, 85, 89, 106 and 116.)

[64] K. Gomadam, V. R. Cadambe, and S. A. Jafar. Approaching the capacity of wireless networks through distributed interference alignment. In *IEEE GLOBECOM*, 2008. (Cited on page 95.)

[65] A. Goswami, L. Ortiz, and S. Das. Wigem: A learning-based approach for indoor localization. CoNEXT, 2011. (Cited on pages 125 and 189.)

[66] M. Grossglauser and D. N. C. Tse. Mobility increase the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 2002. (Cited on pages 49 and 117.)

[67] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with channel state information. *ACM SIGCOMM CCR*, 2011. (Cited on pages 72, 162, 182, 201, 217 and 220.)

[68] D. Halperin, S. Kandula, J. Padhye, P. Bahl, and D. Wetherall. Augmenting data center networks with multi-gigabit wireless links. In *ACM SIGCOMM*, 2011. (Cited on page 91.)

[69] D. Halperin et al. Predictable 802.11 packet delivery from wireless channel measurements. In *CCR*, 2010. (Cited on pages 65, 76, 107, 108 and 115.)

[70] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom, 1999. (Cited on page 125.)

[71] M. H. Hayes. *Statistical Digital Signal Processing and Modeling*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1996. (Cited on pages 236 and 238.)

[72] F. Higgins, A. Tomlinson, and K. Martin. Threats to the swarm: Security considerations for swarm robotics. *International Journal on Advances in Security*, 2, 2009. (Cited on pages 194, 195, 231 and 251.)

[73] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor. Towards the deployment of a mobile robot network with end-to-end performance guarantees. 2006. (Cited on pages 49 and 117.)

[74] C. Hsieh-Chung et al. Determining RF Angle of Arrival using COTS Antenna Arrays: A Field Evaluation. In *MILCOM*, 2012. (Cited on page 157.)

[75] H. Hu. Openwifi : a consumer wifi sharing system. MEng. Thesis, MIT, 2007. (Cited on pages 49 and 83.)

[76] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. A Close Examination of Performance and Power Characteristics of 4G LTE Networks. MobiSys, 2012. (Cited on pages 125, 127, 133 and 157.)

[77] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck. An In-depth Study of LTE: Effect of Network Protocol and Application Behavior on Performance. SIGCOMM, 2013. (Cited on pages 125, 127 and 157.)

[78] Huawei. SingleSON Whitepaper. http://www.huawei.com/en/static/SingleSON-75714-1-127574.pdf, 2012. (Cited on page 157.)

[79] M. Hlzl, R. Neumeier, and G. Ostermayer. Analysis of compass sensor accuracy on several mobile devices in an industrial environment. In *EUROCAST*. 2013. (Cited on page 179.)

[80] A. Jadbabaie, J. Lin, and A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 1, 2003. (Cited on pages 40, 192, 193, 197, 198, 200, 220 and 229.)

[81] D. Jin and J. Song. A traffic flow theory aided physical measurement-based sybil nodes detection mechanism in vehicular ad-hoc networks. In *Computer and Information Science (ICIS), 2014 IEEE/ACIS 13th International Conference on*, pages 281–286, June 2014. (Cited on pages 195 and 251.)

[82] K. Joshi, S. Hong, and S. Katti. PinPoint: Localizing Interfering Radios. NSDI, 2013. (Cited on pages 33, 34, 37, 124, 130, 157, 158, 159, 189, 222 and 227.)

[83] Y. Kakishima, T. Kawamura, Y. Kishiyama, H. Taoka, and T. Nakamura. Experimental Evaluation on Throughput Performance of Asymmetric Carrier Aggregation in LTE-Advanced. In *VTC Spring*, 2011. (Cited on pages 129 and 147.)

[84] C. Kaplinsky. Tightvnc related software. april 2008. (Cited on page 78.)

[85] H. Klausing. Feasibility of a sar with rotating antennas (rosar). In *Microwave Conference, 1989*, 1989. (Cited on page 251.)

[86] H. W. Kuhn. The Hungarian Method for the Assignment Problem. *NRL Quarterly*, 1955. (Cited on page 136.)

[87] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi. Bringing cross-layer MIMO to today's wireless LANs. In *ACM SIGCOMM*, 2013. (Cited on pages 7, 11, 44, 85 and 116.)

[88] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate Indoor Localization with Zero Startup Cost. MOBICOM, 2014. (Cited on pages 7, 11, 122 and 251.)

[89] S. Kumar, E. Hamed, D. Katabi, and L. E. Li. LTE Radio Analytics Made Easy and Accessible. SIGCOMM, 2014. (Cited on pages 7, 11, 120 and 251.)

[90] A. Kushleyev, B. MacAllister, and M. Likhachev. Planning for landing site selection in the aerial supply delivery. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1146–1153, Sept 2011. (Cited on page 251.)

[91] F. C. Lab. Announcing connectivity lab at facebook, 2014. (Cited on page 197.)

[92] G. X. R. Labs. Loon for all, 2014. (Cited on page 197.)

[93] O. Landron, M. Feuerstein, and T. Rappaport. A comparison of theoretical and empirical reflection coefficients for typical exterior wall surfaces in a mobile radio environment. *IEEE Transactions on Antennas and Propagation*, 1996. (Cited on page 101.)

[94] C. Laurendeau and M. Barbeau. Centroid Localization of Uncooperative Nodes in Wireless Networks Using a Relative Span Weighting Method. *EURASIP Journal on Wireless Communications and Networking*, 2010. (Cited on page 157.)

[95] J. Le Ny, A. Ribeiro, and G. Pappas. Adaptive communication-constrained deployment of mobile robotic networks. In *ACC*, 2012. (Cited on pages 40, 192, 200, 204, 222, 226 and 227.)

[96] Z. Li et al. Robust statistical methods for wireless localization in sensor networks. In *IPSN*, 2005. (Cited on page 178.)

[97] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo. Zero-configuration indoor localization over 802.11 wireless infrastructure. *IEEE Wirel. Netw.*, 2010. (Cited on pages 125 and 189.)

[98] K. Lin, S. Gollakota, and D. Katabi. Random access heterogeneous mimo networks. In *SIGCOMM*, 2011. (Cited on pages 48, 51, 52, 55, 56, 57, 71, 75 and 82.)

[99] K. C.-J. Lin, S. Gollakota, and D. Katabi. Random access heterogeneous MIMO networks. In *ACM SIGCOMM*, 2011. (Cited on pages 35, 36, 45, 85, 89, 94, 110, 111 and 116.)

[100] L. Lin and M. A. Goodrich. Uav intelligent path planning for wilderness search and rescue. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 709–714. IEEE, 2009. (Cited on page 251.)

[101] M. Lindhe, K. Johansson, and A. Bicchi. An experimental study of exploiting multipath fading for robot communications. In *RSS*, 2007. (Cited on pages 193, 198, 216, 226 and 227.)

[102] M. Lindhé and K. H. Johansson. Adaptive exploitation of multipath fading for mobile sensors. In *ICRA*, 2010. (Cited on pages 198 and 227.)

[103] M. Lindhe, K. H. Johansson, and A. Bicchi. An experimental study of exploiting multipath fading for robot communications. In *Robotics: Science and Systems*, 2007. (Cited on pages 49 and 117.)

[104] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Push the limit of WiFi based localization for smartphones. Mobicom, 2012. (Cited on pages 34, 37, 125, 159 and 189.)

[105] H. Liu, Y. Wang, J. Liu, J. Yang, and Y. Chen. Practical user authentication leveraging channel state information (csi). In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '14, pages 389–400, New York, NY, USA, 2014. ACM. (Cited on page 251.)

[106] X. Liu, A. Li, X. Yang, and D. Wetherall. Passport: Secure and adoptable source authentication. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pages 365–378, Berkeley, CA, USA, 2008. USENIX Association. (Cited on page 251.)

[107] K. Liu et al. Guoguo: Enabling fine-grained indoor localization via smartphone. MobiSys, 2013. (Cited on pages 124, 159 and 189.)

[108] X. Liu et al. Pushing the envelope of indoor wireless spatial reuse using directional APs and clients. MobiCom, 2010. (Cited on pages 49 and 83.)

[109] V. Loscri et al. Carrier Independent Localization Techniques for GSM Terminals. In *PIMRC*, 2008. (Cited on pages 34 and 157.)

[110] G. Loukas, S. Timotheou, and E. Gelenbe. Robotic wireless network connection of civilians for emergency response operations. *ISCIS '08*, oct. 2008. (Cited on pages 194 and 231.)

[111] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, IPSN '15, pages 178–189, New York, NY, USA, 2015. ACM. (Cited on page 34.)

[112] M. Maddah-Ali, A. Motahari, and A. Khandani. Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis. *Information Theory, IEEE Transactions on*, 2008. (Cited on page 116.)

[113] M. MalmirChegini and Y. Mostofi. On the spatial predictability of communication channels. *Wireless Communications, IEEE Transactions on*, 11(3):964–978, 2012. (Cited on pages 40, 192, 193, 195, 198, 226, 227 and 232.)

[114] S. Martinez, J. Cortes, and F. Bullo. Motion coordination with distributed information. *Control Systems Magazine, IEEE*, 1, 2007. (Cited on page 226.)

[115] C. P. Mathews and M. D. Zoltowsk. Signal subspace techniques for source localization with circular sensor arrays. Purdue University TechReport, 1994. (Cited on page 239.)

[116] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *CCR*, 2008. (Cited on pages 44 and 82.)

[117] N. Michael, M. Zavlanos, V. Kumar, and G. Pappas. Maintaining connectivity in mobile robot networks. In *Experimental Robotics*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 117–126. Springer Berlin/Heidelberg, 2009. (Cited on page 227.)

[118] Microsoft. Kinect. http://www.kinect.com. (Cited on page 189.)

[119] K. Miller, A. Sanne, K. Srinivasan, and S. Vishwanath. Enabling real-time interference alignment: promises and challenges. MobiHoc, 2012. (Cited on page 116.)

[120] G. M.Knoblach and E. A.Frische. Airborne constellation of communications platforms and method. *U.S. Patent 6,628,941*, 1, 1999. (Cited on page 197.)

[121] L. Moreau. Stability of continuous-time distributed consensus algorithms. In *Decision and Control, International Conference on*, 2004. (Cited on page 197.)

[122] A. S. Motahari, S. O. Gharan, M. A. Maddah-Ali, and A. K. Khandani. Forming pseudo-MIMO by embedding infinite rational dimensions along a single real line: Removing barriers in achieving the dofs of single antenna systems. *CoRR*, 2009. (Cited on page 116.)

[123] R. Murty, J. Padhye, R. Chandra, A. Wolman, and B. Zill. Designing high performance enterprise wi-fi networks. NSDI, 2008. (Cited on pages 48 and 83.)

[124] R. Murty, J. Padhye, A. Wolman, and M. Welsh. Dyson: an architecture for extensible wireless lans. USENIXATC, 2010. (Cited on pages 48 and 83.)

[125] A. Networks. Outdoor antennas and rf coverage strategies, 2014. (Cited on pages 193 and 199.)

[126] J. Newsome, E. Shi, D. Song, and A. Perrig. The sybil attack in sensor networks: analysis defenses. In *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, pages 259–268, April 2004. (Cited on page 231.)

[127] M. Niessner, A. Dai, and M. Fisher. Combining Inertial Navigation and ICP for Real-time 3D Surface Reconstruction. Eurographics, 2014. (Cited on pages 123, 160 and 165.)

[128] H. Nyqvist et al. A high-performance tracking system based on camera and IMU. In *FUSION*, 2013. (Cited on page 176.)

[129] R. Olfati-Saber, J. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 1, 2007. (Cited on pages 192, 193, 197, 198 and 229.)

[130] R. Olfati-Saber and R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, Sept 2004. (Cited on pages 194 and 231.)

[131] S. J. Orfanidis. Electromagnetic waves and antennas. http://www.ece.rutgers.edu/ orfanidi/ewa/. (Cited on pages 140, 163 and 185.)

[132] L. E. Parker. Distributed algorithms for multi-robot observation of multiple moving targets. *Autonomous Robots*, 12, 2002. (Cited on pages 194 and 231.)

[133] V. Paxson and S. Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM ToN*, 1995. (Cited on page 75.)

[134] V. Pejovic and E. Belding. A context-aware approach to wireless transmission adaptation. In *SECON*, 2011. (Cited on pages 48, 51 and 82.)

[135] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: A high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys, 2007. (Cited on page 125.)

[136] S. W. Peters and R. W. Heath. Cooperative algorithms for MIMO interference channels. *IEEE Transactions on Vehicular Technology*, 2011. (Cited on page 95.)

[137] T. Pham and B. Sadler. Wideband array processing algorithms for acoustic tracking of ground vehicles. *US Army Research Laboratory Report*, 1, 1997. (Cited on page 227.)

[138] J. Pires, W.R., T. de Paula Figueiredo, H. Wong, and A. Loureiro. Malicious node detection in wireless sensor networks. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, pages 24–, April 2004. (Cited on pages 248, 249 and 251.)

[139] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *MOBICOM*, August 2000. (Cited on page 125.)

[140] Qualcomm. Gimbal Proximity Beacons. http://www.qualcomm.com/solutions/gimbal/beacons. (Cited on page 125.)

[141] H. Rahul, S. S. Kumar, and D. Katabi. Megamimo: Scaling wireless capacity with user demand. In *SIGCOMM*, 2012. (Cited on pages 33, 34, 35, 48, 51, 52, 56, 70, 75, 82, 116, 176, 204 and 206.)

[142] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. Mobicom, 2012. (Cited on pages 33, 34, 37, 125, 159 and 189.)

[143] E. Rozner, Y. Mehta, A. Akella, and L. Qiu. Traffic-aware channel assignment in enterprise wireless lans. In *ICNP'07*. (Cited on pages 48 and 83.)

[144] S. Salvador and P. Chan. Toward Accurate Dynamic Time Warping in Linear Time and Space. *IDA*, 2007. (Cited on page 143.)

[145] Sanjole. The WaveJudge Wireless Test System. http://www.sanjole.com/our-products/wavejudge-test-system/. (Cited on pages 125 and 156.)

[146] I. Sargeant and A. Tomlinson. Modelling malicious entities in a robotic swarm. In *Digital Avionics Systems Conference (DASC), 2013 IEEE/AIAA 32nd*, Oct 2013. (Cited on pages 194, 195, 231 and 251.)

[147] M. Schuresko and J. Cortes. Distributed tree rearrangements for reachability and robust connectivity. *Intell. Robot. Syst.*, 1, 2009. (Cited on page 226.)

[148] M. Schwager, B. J. Julian, and D. Rus. Optimal coverage for multiple hovering robots with downward facing cameras. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3515–3522, May 2009. (Cited on pages 194, 195, 231, 232, 234, 244, 245 and 251.)

[149] M. Schwager, D. Rus, and J.-J. Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009. (Cited on pages 244 and 245.)

[150] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the mona lisa: Spot localization using phy layer information. MobiSys, 2012. (Cited on pages 124 and 189.)

[151] W.-L. Shen, Y.-C. Tung, K.-C. Lee, K. C.-J. Lin, S. Gollakota, D. Katabi, and M.-S. Chen. Rate adaptation for 802.11 multiuser MIMO networks. In *ACM MobiCom*, 2012. (Cited on page 116.)

[152] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell. Detecting 802.11 mac layer spoofing using received signal strength. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages –, April 2008. (Cited on pages 194, 231 and 247.)

[153] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong. Argos: Practical many-antenna base stations. In *ACM MobiCom*, 2012. (Cited on page 116.)

[154] V. Shrivastava et al. CENTAUR: Ralizing the full potential of centralized wlans through hybrid data path. MobiCom, 2009. (Cited on pages 49, 83, 124, 125 and 189.)

[155] S. Singh et al. One strategy does not serve all: tailoring wireless transmission strategies to user profiles. HotNets, 2012. (Cited on page 83.)

[156] M. Solutions. White Paper: Femtocells - The Gateway to the Home. 2010. (Cited on page 127.)

[157] J. Sommers and P. Barford. Cell vs. WiFi: On the Performance of Metro Area Mobile Connections. IMC, 2012. (Cited on pages 125 and 157.)

[158] Sony bravia wireless link module. http://www.lg.com/us/tv-accessories/lg-AN-WL100W-wireless-media-kit. (Cited on page 91.)

[159] J. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *Automatic Control, IEEE Transactions on*, 1, 2000. (Cited on pages 40, 192, 200, 204, 220 and 222.)

[160] D. Spanos and R. Murray. Motion planning with wireless network constraints. In *Proceedings of the ACC*, 2005. (Cited on page 197.)

[161] P. Stoica and N. Arye. Music, maximum likelihood, and cramer-rao bound. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(5):720–741, May 1989. (Cited on pages 238 and 239.)

[162] P. Stoica and R. L. Moses. *Spectral Analysis of Signals*. Prentice Hall, 2005. (Cited on page 205.)

[163] C. Suh and D. Tse. Interference alignment for cellular networks. In *Allerton*, 2008. (Cited on page 116.)

[164] L. Suresh et al. Towards programmable enterprise wlans with odin. HotSDN, 2012. (Cited on pages 49 and 83.)

[165] A. Tahbaz-Salehi and A. Jadbabaie. On recurrence of graph connectivity in vicsek's model of motion coordination for mobile autonomous agents. In *ACC*, pages 699 –704, july 2007. (Cited on page 197.)

[166] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang, M. Chen, and G. M. Voelker. SAM: enabling practical spatial multiple access in wireless LAN. In *ACM MobiCom*, 2009. (Cited on pages 45, 48, 51, 82, 85, 93 and 116.)

[167] K. Tan et al. Fine-grained channel access in wireless lan. *CCR*, 2011. (Cited on pages 51 and 84.)

[168] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik. Indoor localization without infrastructure using the acoustic background spectrum. MobiSys, 2011. (Cited on pages 34 and 159.)

[169] ThinkRF. Product Concept Brief for Distributed and Remote LTE Analysis. http://thinkrf.com/. (Cited on pages 125 and 156.)

[170] A. Tirumala et al. Iperf: The tcp/udp bandwidth measurement tool. *http://dast.nlanr.net/Projects*, 2005. (Cited on page 75.)

[171] B. Triggs et al. Bundle adjustment a modern synthesis. In *Vision Algorithms Theory/Practice*, 2000. (Cited on pages 162, 180, 181 and 189.)

[172] D. Tse and P. Vishwanath. *Fundamentals of Wireless Communications*. Cambridge University Press, 2005. (Cited on pages 46, 85, 89, 96, 164 and 168.)

[173] J. N. Twigg, J. Fink, P. L. Yu, and B. M. Sadler. Efficient base station connectivity area discovery. *Int. Journal of Robotics Research*, 32(12), 2013. (Cited on page 227.)

[174] Netgear a6200 usb wi-fi adapter with sliding antennas. www.netgear.com/home/products/wireless-adapters/ultimate-wireless-adapters/a6200.aspx. (Cited on pages 47, 86, 88 and 117.)

[175] M. A. M. Vieira, M. E. Taylor, P. Tandon, M. Jain, R. Govindan, G. S. Sukhatme, and M. Tambe. Mitigating multi-path fading in a mobile mesh network. *Ad Hoc Netw.*, 1, 2013. (Cited on page 227.)

[176] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. *CCR*, 2009. (Cited on pages 48, 51 and 82.)

[177] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. MobiSys, 2012. (Cited on page 189.)

[178] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus. RF-compass: Robot object manipulation using rfids. MobiCom, 2013. (Cited on pages 124, 125, 163, 189 and 228.)

[179] J. Wang and D. Katabi. Dude, where's my card?: RFID positioning that works with multipath and non-line of sight. SIGCOMM, 2013. (Cited on pages 124, 125, 130, 138, 140, 143, 154, 157, 159, 163, 189, 206, 227, 228 and 251.)

[180] T. Wang and Y. Yang. Analysis on perfect location spoofing attacks using beamforming. In *INFOCOM, 2013 Proceedings IEEE*, pages 2778–2786, April 2013. (Cited on pages 249 and 251.)

[181] X. Wang, V. Yadav, and S. Balakrishnan. Cooperative uav formation flying with obstacle/collision avoidance. *Control Systems Technology, IEEE Transactions on*, 15(4):672–679, 2007. (Cited on page 251.)

[182] Y. Wang, G. Attebury, and B. Ramamurthy. A survey of security issues in wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 8(2):2–23, Second 2006. (Cited on pages 195, 250 and 251.)

[183] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 1992. (Cited on page 125.)

[184] WiGle. Database. https://wigle.net. (Cited on page 159.)

[185] A. Williams, D. Ganesan, and A. Hanson. Aging in place: fall detection and localization in a distributed smart camera network. In *Proceedings of the 15th International Conference on Multimedia*, 2007. (Cited on page 125.)

[186] C. Wu. VisualSFM : A Visual Structure from Motion System. http://ccwu.me/vsfm/. (Cited on pages 162 and 182.)

[187] C. Wu, Z. Yang, Y. Liu, and W. Xi. Wireless indoor localization without site survey. In *INFOCOM*, 2012. (Cited on page 189.)

[188] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. NSDI, 2013. (Cited on pages 33, 34, 37, 124, 130, 138, 154, 157, 159, 163, 168, 173, 178, 189, 222 and 227.)

[189] J. Xiong and K. Jamieson. SecureArray: Improving WiFi Security with Fine-grained Physical-layer Information. MobiCom, 2013. (Cited on pages 143, 195 and 251.)

[190] Y. Yan and Y. Mostofi. Co-optimization of communication and motion planning of a robotic operation in fading environments. In *IEEE ASILOMAR*, 2011. (Cited on pages 49 and 117.)

[191] Y. Yan and Y. Mostofi. Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments. *Wireless Communications, IEEE Transactions on*, 1, 2013. (Cited on pages 193, 198, 199, 204, 226 and 227.)

[192] J. Yang, Y. Chen, W. Trappe, and J. Cheng. Detection and localization of multiple spoofing attackers in wireless networks. *Parallel and Distributed Systems, IEEE Transactions on*, 24(1):44–58, Jan 2013. (Cited on pages 195 and 251.)

[193] L. Yang, W. Hou, L. Cao, B. Y. Zhao, and H. Zheng. Supporting demanding wireless applications with frequency-agile radios. NSDI, 2010. (Cited on pages 48, 51 and 82.)

[194] Z. Yang, E. Ekici, and D. Xuan. A localization-based anti-sensor network system. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2396–2400, May 2007. (Cited on pages 195 and 251.)

[195] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: Wireless indoor localization with little human intervention. Mobicom, 2012. (Cited on pages 125 and 189.)

[196] K. Yap et al. The stanford openroads deployment. In *WINTECH*, 2009. (Cited on pages 49 and 83.)

[197] M. Youssef and A. Agrawala. The Horus WLAN location determination system. MobiSys, 2005. (Cited on pages 124 and 189.)

[198] P. Zerfos et al. Dirac: a software-based wireless router system. MobiCom, 2003. (Cited on pages 48 and 83.)

[199] L. Zheng and D. Tse. Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels. *IEEE Transactions on Information Theory*, 2003. (Cited on pages 55, 56, 77, 204, 236 and 237.)