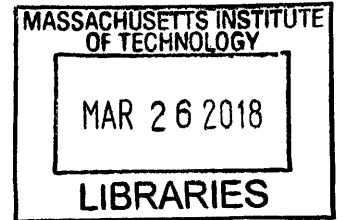


Software-Hardware Systems for the Internet-of-Things

by

Omid Salehi-Abari

M.S. in Electrical Engineering and Computer Science, Massachusetts Institute of Technology (2012)
B.Eng. in Communications Engineering, Carleton University (2010)



ARCHIVES

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

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2018

© Massachusetts Institute of Technology 2018. All rights reserved.

Signature redacted

Author
Department of Electrical Engineering and Computer Science
January 9, 2018

Signature redacted

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

Signature redacted

Certified by.....
Anantha P. Chandrakasan
Dean of Engineering
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Signature redacted

Accepted by
Leslie A. Kolodziejki
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Software-Hardware Systems for the Internet-of-Things

by

Omid Salehi-Abari

Submitted to the Department of Electrical Engineering and Computer Science
on January 9, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Although interest in connected devices has surged in recent years, barriers still remain in realizing the dream of the Internet of Things (IoT). The main challenge in delivering IoT systems stems from a huge diversity in their demands and constraints. Some applications work with small sensors and operate using minimal energy and bandwidth. Others use high-data-rate multimedia and virtual reality systems, which require multiple-gigabits-per-second throughput and substantial computing power. While both extremes stress the computation, communications, and energy resources available to the underlying devices, each intrinsically requires different solutions to satisfy its needs. This thesis addresses both bandwidth and energy constraints by developing custom software-hardware systems.

To tackle the bandwidth constraint, this thesis introduces three systems. First, it presents AirShare, a synchronized abstraction to the physical layer, which enables the direct implementation of diverse kinds of distributed protocols for IoT sensors. This capability results in a much higher throughput in today's IoT networks. Then, it presents Agile-Link and MoVR, new millimeter wave devices and protocols which address two main problems that prevent the adoption of millimeter wave frequencies in today's networks: signal blockage and beam alignment. Lastly, this thesis shows how these systems enable new IoT applications, such as untethered high-quality virtual reality.

To tackle the energy constraint, this thesis introduces a VLSI chip, which is capable of performing a million-point Fourier transform in real-time, while consuming 40 times less power than prior fast Fourier transforms. Then, it presents Caraoke, a small, low-cost and low-power sensor, which harvests its energy from solar and enables new smart city applications, such as traffic management and smart parking.

Thesis Supervisor: Dina Katabi

Title: Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Anantha P. Chandrakasan

Title: Dean of Engineering

Professor of Electrical Engineering and Computer Science

To my parents

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisors, Dina Katabi and Anantha Chandrakasan, for their support and guidance throughout my years at MIT. Dina has been amazingly supportive. Her passion, hard work, and dedication to the success of her students is truly inspiring. I could not have hoped for a better advisor, mentor, researcher, and role model. I am fortunate to have had such a wonderful advisor guiding me through my PhD, and I have no doubt that I could not have succeeded without her support. I would also like to thank Anantha, who has always been available to support and help me despite his busy schedule. In particular, he has helped me improve my critical thinking skills and my ability to conduct meaningful and practical research. It has been an honor to work with him.

I would also like to thank my thesis committee members Ranveer Chandra and Joshua Smith for their time, support, and invaluable feedback. They have been immensely helpful in shaping this thesis.

I am grateful for Vladimir Stojanovic's assistance. He has supported and guided me through my graduate career and has always been encouraging. I truly admire and respect him. I would also like to thank Adrian Chan and Calvin Plett at Carleton University, Canada. I started my exciting research journey under their supervision, and I learned so much from them, both scientifically and personally. Their encouragement and support were and continue to be invaluable.

I would also like to thank the Electrical Engineering and Computer Science Department staff, especially Janet. She has been helpful from the day I received my admission letter all the way to the day I submitted this thesis. I also owe many thanks to our group admin, Mary, who was always available to help me with ordering

equipment, making travel arrangements, and more importantly, ordering food for us before our deadlines.

I would not have been able to complete the research in this thesis without my collaborators. I had an amazing time working with Ezz on Sparse Fourier Transform chip. I will never forget all those sleepless nights in the lab. Working with Deepak on Caraoke was the most fun and most challenging part of my research. Evaluating the system outdoors during the snowiest winter was not easy, but Deepak helped make it enjoyable. I also enjoyed working with Austin and I hope we can work together again in the future.

My labmates at NETMIT, Ananthagroup, and ISG created a friendly and pleasant environment and offered their help whenever needed. I could not have wished for a more supportive and fun group of labmates. I would like to thank Anubhav, Arun, Austin, Chen-Yu, Dinesh, Deepak, Georgios, Phil, John, Ezz, Fadel, Haitham, Jue, Lixin, Mingmin, Rahul, Swarun, Yildiz, and Zach. Special thanks to Fred and Fabian, who were my first mentors at MIT. I enjoyed working with them and learned a lot from it.

The best part of my MIT life was making amazing friends who have always been there for me. Arman has been a source of energy and my partner-in-crime. He has been an amazing friend. Reza and Iman have made my PhD life much more fun, especially during deadlines. It was always fun to hang out with them and have kabab "Soltani." I am also grateful to all my other friends at MIT and in the Boston area: Hamid, Reza, Arman, Donna, Amir, Mahsa, Ali, Mina, AmirBahador, Naghmeh, Damoon, Elham, Afrooz, Saman, Elica, Ali, Azarakhsh, Soheil, Saeed, Vahid, Meena, and Mohammad.

Finally, yet most importantly, I would like to thank my family. My brother AmirAli, sister-in-law Julie, and adorable niece Sarah have always been encouraging and inspiring. And my parents have always been supportive and loving. I owe everything to their unconditional support.

Previously Published Material

Chapter 2 revises a previous publication [44]: O. Abari, H. Rahul, D. Katabi and M. Pant. AirShare: Distributed Coherent Transmission Made Seamless. INFOCOM, 2015.

Chapter 3 revises a previous publication [42]: O. Abari, H. Hassanieh, M. Rodriguez and D. Katabi. Millimeter Wave Communications: from Point-to-Point Links to Agile Network Connections. HotNets, 2016.

Chapter 4 revises a previous publication [38]: O. Abari, D. Bharadia, A. Duffield and D. Katabi. Cutting the Cord in Virtual Reality. HotNets, 2016.

Chapter 4 revises a previous publication [40]: O. Abari, D. Bharadia, A. Duffield and D. Katabi. Enabling High-Quality Untethered Virtual Reality. NSDI, 2017.

Chapter 5 revises a previous publication [41]: O. Abari, E. Hamed, H. Hassanieh, A. Agarwal, D. Katabi, A. Chandrakasan, and V. Stojanovic. A 0.75 Million-Point Fourier Transform Chip for Frequency-Sparse Signals, ISSCC, 2014.

Chapter 6 revises a previous publication [45]: O. Abari, D. Vasisht, D. Katabi and A. Chandrakasan. Caraoke: An E-Toll Transponder Network for Smart Cities. SIGCOMM, 2015.

Contents

Acknowledgements	7
Previously Published Material	9
List of Figures	15
List of Tables	19
1 Introduction	21
1.1 Combating Bandwidth Constraints	22
1.2 Combating Energy Constraints	26
1.3 Organization	28
I Combating Bandwidth Constraints	29
2 Distributed Coherent Transmission	31
2.1 Related Work	35
2.2 AirShare	37
2.2.1 Why Do Wireless Nodes Have CFO?	38
2.2.2 How Does AirShare Work?	38
2.2.3 AirShare's System Architecture	41
2.2.4 Dealing with Wireless Channel Impediments	43
2.2.5 Range of AirShare Emitter	44
2.2.6 Scalability	46

2.3	AirShare Implementation	47
2.4	Empirical Evaluation of AirShare	48
2.4.1	Eliminating CFO between nodes	48
2.4.2	Enabling Coherent Transmission	51
2.5	Applications of AirShare	53
2.5.1	Distributed MIMO with AirShare	53
2.5.2	Distributed Rate Adaptation for Wireless Sensors	55
2.6	Discussion	58
3	Agile mmWave Networks	59
3.1	Related Work	64
3.2	Illustrative Example	66
3.3	Agile-Link	69
3.3.1	Problem Statement	69
3.3.2	Agile-Link's Algorithm	69
3.3.3	Antenna Arrays on Both Transmitter and Receiver	74
3.3.4	Algorithm Implementation	75
3.4	Agile-Link Implementation Platform	76
3.4.1	Heterodyne Architecture	76
3.4.2	Phased Array Calibration	78
3.4.3	Radio Performance:	80
3.5	Experimental Evaluation	82
3.5.1	Compared Schemes	82
3.5.2	Evaluation Metrics	83
3.5.3	Beam Alignment Accuracy vs. the Ground Truth	84
3.5.4	Alignment Accuracy in Multipath Environments	86
3.5.5	Beam Alignment Latency	87
3.6	Discussion	90
4	High-Quality Untethered Virtual Reality	91
4.1	Related Work	94

4.2	MoVR Overview	96
4.3	Blockage Problem	97
4.3.1	Impact of Blockage	98
4.3.2	Programmable mmWave Mirrors	101
4.4	Dealing with Mobility	104
4.4.1	Impact of Beam Misalignment	104
4.4.2	Beam Alignment and Tracking	105
4.5	System Details	109
4.6	Evaluation	110
4.6.1	Blockage During an Actual VR Game	111
4.6.2	MoVR's Mirror Performance	112
4.6.3	MoVR's Beam Alignment and Tracking Performance	114
4.6.4	MoVR System Performance	117
4.7	Discussion	117
 II Combating Energy Constraints		120
 5 A million-point Fourier transform chip		121
5.1	Algorithm	122
5.1.1	STEP 1: Frequency Bucketization	122
5.1.2	STEP 2: Frequency Estimation	123
5.1.3	STEP 3: Collision Detection and Resolution	124
5.2	Architecture	126
5.2.1	FFT's Micro-architecture	128
5.2.2	Estimation's Micro-architecture	128
5.2.3	Collision Detection's Micro-architecture	131
5.2.4	Memory Structures	131
5.3	Related Work and Results	133
5.4	Discussion	134

6	An E-Toll Transponder Network for Smart Cities	137
6.1	Related Work	140
6.2	Background	142
6.3	Caraoke Overview	144
6.4	Counting Despite Collisions	146
6.5	Localizing E-Toll Transponders	149
6.6	Detecting a Car's Speed	154
6.7	Decoding Transponders' Ids	155
6.8	Caraoke Multiple Reader Protocol	158
6.9	Caraoke Reader's Hardware Design	159
6.10	Evaluation Setup	161
6.11	Empirical Evaluation of Caraoke	163
6.11.1	Counting Accuracy	163
6.11.2	Localization Accuracy	165
6.11.3	Speed Estimation Accuracy	168
6.11.4	Decoding Accuracy	168
6.11.5	Caraoke Reader's Power Consumption	170
6.12	Discussion	170
7	Conclusion	173
7.1	Lessons Learned	175
7.2	Future Work	177
7.2.1	3D Printing of Sensors:	178
7.2.2	Wireless Systems for Driverless Cars:	178
7.2.3	Security for IoT Sensors:	179
A	Analysis of Agile-Link System	181
	Bibliography	187

List of Figures

- 1-1 Thesis Contributions 23
- 2-1 Illustration of AirShare’s Design 39
- 2-2 AirShare’s Network Topology 41
- 2-3 AirShare’s System Architecture 41
- 2-4 AirShare is Scalable to Large Networks 46
- 2-5 CFO between pairs of nodes 48
- 2-6 CFO between pairs of nodes at 2.4 GHz carrier frequency 49
- 2-7 CFO between pairs of AirShare-equipped nodes 51
- 2-8 Received Constellation Points for Nodes Transmitting BPSK 52
- 2-9 Traditional AP Deployments versus Distributed MIMO 53
- 2-10 Distributed MIMO using AirShare 54
- 2-11 Traditional Sensor Networks versus Distributed Rate Adaptation 57
- 2-12 Channel Quality versus Throughput for Distributed Rate Adaptation 57
- 3-1 mmWave Communication 61
- 3-2 Illustrative Example of Agile-Link’s Algorithm 66
- 3-3 Example of Hierarchical Search in the Presence of Multipath 68
- 3-4 Hashing Beam Patterns 71
- 3-5 Agile-Link Platform 76
- 3-6 Agile-Link’s Architecture 77
- 3-7 Phased Array Radiation Pattern 78
- 3-8 Phase Shifters’ Performance 79

3-9	Performance of Phased Array Calibration	80
3-10	Agile-Link Coverage	81
3-11	256 QAM and 16 QAM Constellation	81
3-12	Recovered Directions	84
3-13	Beam Accuracy with a Single Path	86
3-14	Beam Accuracy with Multipath	87
3-15	Beam Alignment Latency	88
3-16	802.11ad Beacon Interval Structure	89
4-1	Virtual Reality Experience	92
4-2	MoVR's Setup	97
4-3	Blockage Scenarios	98
4-4	Blockage Impact on Data Rate	99
4-5	MoVR Programmable mmWave Mirror	101
4-6	MoVR's Mirror Block Diagram	101
4-7	Leakage Between Mirror's Transmit and Receive Antennas	103
4-8	Beam Alignment in mmWave Radios	104
4-9	SNR versus Amount of Headset Rotation	105
4-10	Localizing the Mirror	108
4-11	MoVR Mirror's Controller Board	111
4-12	Blockage Duration	112
4-13	MoVR's Mirror Performance	113
4-14	AP to Mirror Beam Alignment Accuracy	115
4-15	MoVR Beam Alignment Accuracy	115
4-16	SNR at the Headset's Receiver	119
5-1	Bucketization using Aliasing Filter.	123
5-2	Resolving collisions with Co-prime Filters	125
5-3	Block Diagram of the $2^{10} \times 3^6$ -point Sparse Fourier Transform	126
5-4	FFT's Input	127
5-5	The Micro-architecture of the 2^{10} -point FFTs	129

5-6	The Micro-architecture of Collision Detection and Estimation units . . .	130
5-7	The Micro-architecture of Phase Detection.	131
5-8	Memory Structures.	132
5-9	Die Photo of the Sparse Fourier Transform Chip	132
5-10	Chip Measurement Results	133
6-1	Transponder Transmission	143
6-2	Illustration of Caraoke	145
6-3	Transponders Response in Frequency Domain	148
6-4	Transponder Localization	150
6-5	CaraokeAntennas Re-configuration	152
6-6	Localizing the Car from AoA	153
6-7	Decoding a Transponder's Response	156
6-8	Caraoke Reader	159
6-9	Block Diagram of the Caraoke Reader	160
6-10	CaraokeExperimental Setup	163
6-11	Counting Accuracy	164
6-12	Traffic Monitoring at an Intersection	165
6-13	Transponder's Localization Accuracy	166
6-14	Multipath Profile	167
6-15	Speed Detection Accuracy	168
6-16	Identification Time versus Number of Transponders	169

List of Tables

3.1	Beam alignment latency for different array size	90
5.1	Sparse Fourier Transform chip versus traditional FFT Chips	134

Chapter 1

Introduction

With the advent of the Internet of Things (IoT), billions of new connected devices will come online, enabling new applications in very diverse areas: from smart cities and industrial automation to smart glasses and virtual reality (VR) [37, 75, 183, 198]. While exciting, these applications come with their own new set of network constraints, many of which are beyond the capabilities of today's wireless networks. These constraints can be summarized along two axes:

- **Bandwidth Constraints:** The first set of constraints in designing IoT devices involves bandwidth. Although the number of wireless devices is growing rapidly, the availability of resources, such as radio spectrum, remain largely unchanged. This particular mismatch has put a huge burden on available spectrum [75, 87, 182]. Along the same lines, certain new applications have high data-rate requirements, which inherently need a large amount of bandwidth from a shared spectrum. For example, wireless smart TVs and VR will require extremely high-data-rate internet connections [37, 79].
- **Energy Constraints:** The second set of constraints in designing IoT devices centers on energy [71, 129, 171]. These constraints stem from the small form factor and inaccessible placement of these devices, which restrict the capacity and replacement frequency of the batteries [171]. Furthermore, in order to reduce bandwidth usage, these devices tend to perform a significant amount of

computation locally, resulting in higher energy consumption [102].

Past work which has looked into these constraints has designed algorithms, protocols and underlying hardware in isolation and often without considering the application. Unfortunately, due to the strict requirements of IoT applications, this approach to designing wireless networks does not work for IoT devices. This thesis addresses these constraints by designing IoT systems while considering all layers of the “computer network stack”. The stack encompasses the physical layer, which includes circuits and signals; the network layer which refers to network protocols; and the application layer which delivers the final service to the user. A software-hardware design approach in which all of these layers are jointly designed is essential in order to optimize computing, communications and sensing. In particular, this thesis presents multiple software-hardware systems for IoT that represent either new technologies that enable wireless infrastructure with much higher capabilities than that which exists today, or simple low-cost technologies that can be easily added to existing infrastructure. Further, this thesis shows how these technologies can enable new applications such as untethered VR and smart city services. The following sections briefly introduce these systems and present their core contributions.

1.1 Combating Bandwidth Constraints

The first part of this thesis deals with bandwidth constraints faced by today’s IoT networks. These constraints stems from a shortage of available radio frequency spectrum. Although the number of wireless and IoT devices is growing rapidly, the available unlicensed spectrum remains unchanged [75, 87, 182]. The traditional approach to overcome this challenge is to optimize wireless protocols to use the spectrum more efficiently. However, this approach is no longer sufficient, as it can not keep up with the increase in the number of devices. In response to this challenge, this work adopts an interdisciplinary approach that designs new protocols and algorithms in conjunction with new networking hardware. For example, we introduce a technology which synchronizes wireless devices at the physical layer. This technology enables direct

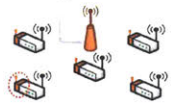





	Technology	Application
Throughput	<p>Network Synchronization</p>  <p>Agile mmWave Networks</p> 	<p>Untethered Virtual Reality</p> 
Energy	<p>A Million-Point Fourier Transform Chip</p>  <p>An E-Toll Transponder Network</p> 	<p>Smart City Services</p> 

Figure 1-1: Thesis Contributions

and seamless implementation of diverse kinds of distributed PHY protocols, which allows multiple nodes to transmit simultaneously over the same frequency band. This capability is necessary, as we are entering an era with a very large scale deployments where each node cannot have exclusive access to spectrum. We further present a new technology that allows mobile networks to benefit from millimeter wave (mmWave) communications, which has traditionally been used only in point-to-point links. This technology has the potential to help shift the burden from overused wireless spectrum bands to underused high-frequency bands. Finally, we show how this technology enables new services, such as high-quality untethered virtual reality, which require very high-data-rate wireless links. Our solutions have been implemented and evaluated in wireless testbeds and have demonstrated significant gains in performance compared to state-of-the-art technologies.

Contributions. This work makes the following key contributions:

- *Distributed Coherent Transmission:* Distributed coherent protocols are theoretically well-understood to provide large gains in network throughput and reliability. These protocols include distributed MIMO, distributed modulation,

distributed compressive sensing over the air, and distributed lattice coding. These schemes assume that independent wireless nodes can perform distributed coherent transmission; that is, they can transmit their signals with predetermined synchronized relative phases to ensure the signals combine with each other over the air in a predictable manner. However, existing radios do not provide this synchronization capability since independent wireless nodes have different crystal oscillators which generate clocks with slightly different frequencies; consequently, the relative phases of their signals are also changing [153].

To address this shortcoming, we develop AirShare[44], which enables a direct and seamless implementation of distributed coherent protocols. AirShare does this by introducing a new coordination primitive that allows multiple nodes to act as if they are driven by a single clock, which is wirelessly received over the air. This enables distributed coherent transmission, as devices have exactly the same frequency (i.e., constant relative phase), causing their transmissions to combine coherently over the air. Accordingly, we implement AirShare in a network of software radios, and demonstrate that it achieves tight frequency synchronization. We also show that it allows direct implementations of cooperative protocols, such as distributed MIMO and distributed rate adaptation.

- *Agile mmWave Networks:* Due to its ability to enable orders of magnitude higher throughput, mmWave technology has been declared a central component in next-generation cellular (5G) and Wi-Fi (802.11ad) networks. However, a key challenge still must be addressed: mmWave signals decay very quickly with distance, requiring a mmWave radio to use highly directional antenna with a very narrow beam to focus its power. Due to the antenna's narrow beam, communication is possible only when the transmitter's and receiver's beams are well-aligned. Current solutions scan the space by trying various beam directions to find the best beam alignment, a process that can take several seconds. While a long delay before establishing a link may be acceptable in fixed point-to-point links, it is unacceptable in cellular or Wi-Fi networks, as a base station

has to quickly switch between users and accommodate mobile clients in these instances.

To overcome this challenge, we introduce Agile-Link [42], a new protocol for fast link establishment in mmWave networks. Agile-Link leverages the fact that there are only a few paths that the mmWave signal can take between the transmitter and the receiver. Thus, instead of creating a narrow beam and trying all beam directions, Agile-Link builds on recent approaches in sparse recovery, and introduces a new hashing and voting mechanism to quickly identify the spatial direction for the best beam alignment. We built a full-fledged mmWave radio with phased arrays and demonstrate that Agile-Link reduces the beam alignment delay by orders of magnitude, enabling the deployment of mmWave links in mobile networks.

- *Untethered Virtual Reality*: Today’s high-end virtual reality (VR) systems require a cable connection to stream high-definition videos from a PC or game console to the headset. This cable significantly limits the user’s mobility and, hence, their overall VR experience. The high-data-rate requirement of this cable (multiple Gbps) precludes its replacement by today’s wireless systems, such as Wi-Fi. To enable untethered VR, we develop MoVR [38, 40], a system that creates a mmWave link between the PC and the headset. As stated previously, mmWave technology enables orders of magnitude higher throughput than today’s wireless systems. However, mmWave signals decay very quickly with distance, meaning that a mmWave radio must focus its power into a very narrow beam. Due to this limitation, high-data-rate communication is only possible when the transmitter’s and receiver’s beams are directed toward each other and well-aligned.

There is still a key challenge which currently prevent this technology from being used in VR systems: blockage. mmWave radios require a clear line-of-sight between the transmitter and the receiver to align their beam and achieve high-data rates. However, in a VR setup, the line-of-sight can be easily blocked if

a user raises her hand between the headset and the transmitter, or even just moves her head, interrupting the VR data stream. To overcome this problem, we present a novel self-configurable mmWave mirror which can be placed on the wall of the room to reroute the signal around any blockage. When the line-of-sight between the transmitter and the headset is blocked, the transmitter beams its signal towards the mirror. Then, the mirror reconfigures itself to amplify and reflect the signal toward the headset. MoVR is a system that allows mmWave links to sustain high data rates even in the presence of blockage, enabling untethered high-quality VR.

1.2 Combating Energy Constraints

The second part of this thesis deals with energy constraints in today's IoT networks. These constraints stem from the small form factor of IoT devices. The traditional approach to improve the energy consumption of wireless systems is to separately optimize the design of each layer of the wireless network stack: from circuit components and hardware to signal processing and networking algorithms. In contrast, we show that by joint optimization across layers, one can restructure the computation and reduce energy consumption for the respective IoT device.

For example, we design a low-power wireless sensor which exploits circuit's non-idealities to decode the signals in a low-power manner. Additionally, our sensor can harvest its energy from solar power. Further, we design and build a chip to compute Fourier transforms for sparse signals, relaxing power-hungry sampling and processing requirements. Finally, we show how such technology can enable new services such as smart city applications which requires a very low power wireless device. 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:

- *A million-point Fourier transform chip:* Many advanced wireless protocols such as wideband spectrum sensing require computing the Fourier transform of large

amounts of data in real-time, a process that cannot be achieved using software alone. On the other hand, hardware implementations of such Fourier transforms are challenging due to their large silicon area, memory requirements and high power consumption.

To tackle these challenges, we have leveraged the fact that the signals in these applications are typically sparse, and built a Fourier transform VLSI chip by exploiting recent advances in sparse recovery [41]. The chip performs a 0.75 million-point Fourier transform in 6 microseconds which is 88 times faster than a C++ implementation of the algorithm running on a 3.4GHz Quad-Core i7 CPU. It consumes 40 times less energy than prior Fourier transform VLSI implementations while also performing the computation on a much larger data set.

- *An E-Toll Transponder Network for Smart Cities:* There is considerable interest in enabling smart city applications, where an infrastructure of low-power sensors can estimate traffic flow, track parking usage, and detect instances of speeding. However, there is a major challenge: deploying sensors at the scale of a city population requires a long time and substantial investment. Our goal was to build a system that delivers smart services using an existing, widely-deployed infrastructure: E-toll transponders (e.g., E-ZPass). This goal is particularly challenging because the transponders are very low power and designed under the assumption that only one of them transmits at any point in time, and hence they have no MAC protocol to prevent collisions. This is precisely why E-toll collection systems use restricted deployments and highly directional antennas; to ensure that only one car responds to a reader's query. Without this physical isolation, all transponders in range would respond simultaneously, creating collisions. This limitation hampers the use of E-toll transponders in smart city services since each vehicle can not be uniquely identified.

To overcome this problem, we present Caraoke [45], a software-hardware solution for smart cities. On the software side, we introduce an algorithm that

enables a network of low-power active RFIDs. Specifically, our algorithm solves the collision problem in low-power RFID devices which have no MAC protocols. We then show how our algorithm can be used to count, localize, and decode many low-power E-toll transponders, despite the collision of their signals. A key feature of our algorithm is its ability to exploit a circuit non-ideality found in all radios, which is called carrier frequency offset. This non-ideality results in different active RFIDs or transponders communicating at slightly different center frequencies. We leverage this property to extract the individual response of each transponder from the collision signal. On the hardware side, We build an embedded reader, enabling three core requirements to be satisfied: low power, low cost and easy deployability. The reader connects to the Internet via a wireless modem and harvests its energy from solar power, making it easy to deploy on street lamps. We then evaluate and demonstrate its ability to enable smart services, such as monitoring the vehicle flow at an intersection, detecting speeding instances, and determining street parking occupancy.

1.3 Organization

The rest of this thesis is organized as follows. Chapter 2 describes AirShare in more detail and how it enables distributed coherent transmission from independent wireless nodes by sharing a single reference clock across them. Chapter 3 details how Agile-Link brings mmWave technology in today's wireless networks by introducing a new algorithm to find the best beam alignment required by mmWave radios. Chapter 4 discusses MoVR and explains how it enables high-quality untethered VR. Chapter 5 presents a VLSI implementation of a Sparse Fourier transform, which performs a million-point Fourier transform in real-time, while consuming 40 times less energy than a traditional fast Fourier transform (FFT). Lastly, Chapter 6 presents Caraoke, a low-power low-cost sensor, which enables a networked system capable of delivering smart services using E-toll transponders.

Part I

Combating Bandwidth Constraints

Chapter 2

Distributed Coherent Transmission

Distributed cooperative PHY protocols are theoretically well understood to provide large gains in throughput and reliability in a large variety of scenarios. These include distributed MIMO [147, 165, 190], distributed modulation [113], distributed compressive sensing over the air [55, 120], distributed lattice coding [139], noisy network coding [128], and transmitter cooperation for cognitive networks [131]. These schemes assume that independent wireless nodes can perform distributed coherent transmission—that is, they can transmit their signals without phase drifts with respect to each other.

However, practical radios do not provide distributed coherent transmission. Independent wireless nodes have different crystal oscillators generating clocks with different frequencies. As a result, different nodes always have an offset in their carrier frequencies (CFO); the CFO causes signals transmitted by every pair of nodes to rotate with respect to each other, and their phases to drift over time. Thus, even if two signals start with their phases aligned in a desired manner, the CFO very quickly causes the phases to rotate with respect to each other and the signals to combine in an undesired manner. Typical CFOs between two wireless radios even those that belong to the same technology (e.g., two Wi-Fi radios) and the same manufacturer vary between 100s of hertz to tens of kilohertz [56, 74]. Such CFOs are large enough to lose coherence even within a single packet.¹

¹For example, a CFO of a few hundred hertz causes a phase misalignment of π or more in less

In this chapter, we investigate how practical radios may deliver an abstraction of distributed coherent transmission. Designers of cooperative PHY protocols (distributed MIMO, distributed modulation, etc.) would then leverage this abstraction and free themselves from having to work out the details of coherent transmission. The most straightforward approach for delivering such an abstraction would connect the nodes to a shared clock using wires [89]. Such a system eliminates CFO and ensures coherent transmission. However, it defeats the notion of a wireless network and is not practical for mobile nodes. Alternatively, one may connect each node to a GPS clock. Such clocks use the GPS signal and temperature-controlled crystals to maintain a very low CFO with respect to each other. Unfortunately, however, GPS clocks are power-hungry, cost hundreds to thousands of dollars, and do not work in indoor settings [118, 179]. As a result, they are neither suitable for sensor nodes nor indoor Wi-Fi deployments. In the absence of a suitable generic abstraction for distributed coherence, most wireless cooperation protocols have remained theoretical [55, 128, 131, 139]. The few protocols which were demonstrated empirically address the coherence issue within a particular context. For example, systems like [56, 153] implement distributed MIMO, but focus specifically on OFDM systems in their phase tracking and compensation algorithms. In contrast, solutions like [185] focus on the RFID technology, where nodes are passive reflectors that do not have CFO. Neither of these solutions however provide a generic coherence abstraction that can be leveraged by various cooperative PHY protocols, and applied broadly across technologies (Wi-Fi, ZigBee, Bluetooth).

Ideally, one would like a solution that: (a) avoids wires and supports mobility. (b) Further, it should be independent of the protocol and the radio technology so that it might be used by a variety of technologies (e.g., Wi-Fi, ZigBee, Bluetooth) to build existing or future distributed communication protocols (e.g., distributed MIMO, distributed compressive sensing, or other cooperative PHY protocols.) (c) Finally, it should be cheap and low-power so that it may be incorporated with cheap wireless nodes such as sensors.

than one millisecond, leading to highly incoherent signals.

This chapter presents AirShare, a primitive that makes distributed coherent transmission seamless while satisfying the above three requirements. At a high-level, AirShare transmits a shared clock on the air and feeds it to the wireless nodes as a reference clock, hence eliminating the root cause for incoherent transmissions. Networks interested in using AirShare for coherent transmission deploy an AirShare clock emitter within radio range, and add a simple circuit to each node to capture the shared clock.

Transforming this design into a practical system, however, requires overcoming multiple challenges.

- Perhaps the most fundamental challenge is: How do we transmit the shared reference clock? At first blush, it might seem that we could just transmit the reference clock signal over the air. The problem, however, is that wireless nodes typically use for their reference clock a sine wave of 10 MHz to 40MHz [46]. FCC regulations, however, forbid transmitting such a low-frequency signal for unlicensed use [92]. Besides, receiving a signal efficiently at this low frequency requires antennas that are several meters long [57], which is impractical for typical wireless nodes. In §2.2.2, we explain how AirShare addresses this problem by introducing a new approach that extracts a low-frequency clock from multiple high-frequency sine waves.
- Second, the circuit used by the individual nodes to receive the clock has to be simple and low-power. It also should be possible to integrate this circuit with existing radio designs as an add-on module, independent from the details of the radio technology (whether it is ZigBee, Wi-Fi, etc.). In §2.2.3, we describe the detailed design of AirShare, demonstrating that the per node circuit includes only simple analog components such as a mixer and a filter and does not need any digital receive chain, allowing it to stay cheap and low-power. Further, it operates as an add-on module that is oblivious to the details of the radio.
- Finally, in §2.2.4, we describe how AirShare deals with various impediments that affect the wireless medium, including the possibility of the clock fading at

some of the nodes.

We built a prototype of AirShare and integrated it with USRP software radio to show that it can be used to augment off-the-shelf radios. We evaluated AirShare and its applications in a wireless testbed with line-of-sight and non line-of-sight scenarios. Our results reveal that AirShare provides tight phase synchronization across multiple nodes. In particular, in our testbed, the median and 95th percentile CFO between two AirShare nodes operating at 2.45 GHz are 0.38Hz (0.16 parts per billion) and 1.24Hz (0.5 parts per billion) respectively. AirShare can thus achieve orders of magnitude tighter synchronization than the traditional situation of free running oscillators on independent nodes.

To demonstrate AirShare’s versatility we used it as a coherence abstraction to build two cooperative protocols:

- **Distributed MIMO** Distributed MIMO is a powerful concept that allows multiple wireless transmitters to behave like one huge MIMO transmitter with the number of antennas equal to the sum of antennas on all the cooperating nodes.

To evaluate how AirShare enables distributed MIMO, we took 10 USRPs that implement Wi-Fi OFDM physical layer. We divided the nodes to 5 transmitters and 5 receivers, and equipped the transmitters with the AirShare clock recipient circuit. We then made the transmitters transmit concurrently to the 5 receivers, while applying MIMO multiplexing. Our results show that AirShare delivers a distributed MIMO system whose throughput scales linearly with the number of transmitters. Specifically, with 5 transmitters, AirShare’s distributed MIMO delivers a median throughput gain of 4.4× over traditional 802.11 style unicast.

- **Distributed Rate Adaptation** Traditional sensor systems such as ZigBee [58] are typically limited to a single low-density modulation, (e.g., BPSK). As a result, they cannot exploit good channel conditions to transmit at dense modulation (e.g., 16-QAM) and achieve higher spectral efficiency. A recent work has proposed a distributed scheme that enables nodes to transmit concurrently and

combine their signals over the air, creating a rate-less modulation scheme that adapts the bits per transmitted symbol to the channel quality [185]. Since this design requires coherent transmission, it has been limited to only RFID nodes which do not have local oscillators or CFO with respect to each other.

We show that AirShare extends such distributed modulation and rate adaptation to generic sensors. We have used our AirShare-equipped USRPs to emulate ZigBee sensors and perform distributed modulation and rate adaptation as described in [185]. Our results show that an AirShare-based ZigBee system can adapt modulation and bitrate to channel quality and achieve throughput gains over traditional ZigBee. Specifically, with 6 ZigBee nodes transmitting jointly to a central node, AirShare provides throughput gains of $1.6 - 3\times$ across a range of SNRs (5-25 dB).

We believe that AirShare provides an important step toward enabling distributed cooperative PHY protocols at the physical layer, and adds a useful primitive to the toolkit available for building such protocols.

2.1 Related Work

Related work falls into three categories.

(a) Shared clocks: The most straightforward approach for sharing the clock is to connect the clock input on the nodes to one external clock via wires. In fact, this is how the research community builds MIMO software radios by connecting multiple USRPs to the same clock output. While this approach eliminates any frequency offset or phase noise between the nodes, it effectively transforms the nodes to one device with multiple components connected via wires.

Another traditional solution to synchronize clocks in a wireless network is to use global positioning system (GPS)-disciplined oscillator (GPSDO) to generate the reference signal [73, 118]. A GPSDO is equipped with a high-performance GPS receiver and antenna, a microprocessor and an oven-controlled crystal oscillator (OCXO).

Since GPS signals are accurate to the nanoseconds, they provide a good reference to compare the frequency of the crystal oscillator and estimate its offset from its nominal value. Since the frequency of the crystal oscillator is a function of the temperature, this offset can be corrected by precise adjustment of the oven temperature. Due to their complexity, these clock generators cost thousands of dollars which makes them unfeasible to be used in each wireless node. In addition, line of sight to the GPS satellite is required, which makes these clocks generator applicable only to outdoor applications. In contrast, AirShare does not require expensive hardware and can also work in indoor applications.

There are also terrestrial time signals such as WWV and WWVB [141, 142], transmitted by the National Institute of Standards and Technology, which can synchronize receivers on the order of seconds. Similarly, traditional clock synchronization schemes like NTP can synchronize nodes to within 10s of milliseconds. However, unlike AirShare, these schemes are too coarse grained for PHY systems, which require synchronization on the order of nanoseconds.

Many communication systems, such as PCI Express and FireWire [52, 60], transmit a shared clock with the data to allow the receiver to synchronize its clock with the transmitter. However, unlike in AirShare, these schemes connect the transmitter to the receiver over a wired link and do not apply to wireless systems, or to multiple transmitters.

(b) Phase tracking and compensation: The typical approach for dealing with frequency offset in wireless nodes is to estimate the resulting phase rotation and compensate for it. In particular, traditional OFDM transmitters send pilots in some subcarriers in the data stream, which OFDM receivers then use to track the phase drift of the transmitted signal and compensate for it. For distributed MIMO, practical schemes that can achieve only diversity [150] or can perform both diversity and multiplexing (e.g. MegaMIMO and AirSync [56, 153]) incorporate sophisticated schemes to track transmitter phase drift and correct signals prior to transmission. These schemes are necessary to ensure that the signals from different transmitters combine coherently at each receiver. In contrast, with AirShare, wireless nodes synchronizes their clocks

over the air eliminating the need for phase tracking and compensation either at the receiver or other transmitters and can achieve both diversity and multiplexing gain. Further, AirShare's system is independent of the protocol, and modulation schemes, and applies to various wireless technology (WiFi, Bluetooth, Zigbee, etc.).

(c) Pulse Coupled Oscillators (PCO): Our work is also related to past works focused on synchronizing wireless nodes using pulse coupled oscillators (PCO) [136, 188, 189]. In the PCO approach, oscillators of different nodes are coupled, and therefore become synchronized. To achieve the required coupling in wireless nodes, each node sends periodic pulses to its neighbors, which then use these pulses to synchronize their oscillators. The closest past work to ours is [188, 189], which uses PCO to achieve tight time synchronization in carrierless IR-UWB. However, this work shows that the jitter can go up to a few nS in nodes spaced a few meters apart from each other. Although this amount of jitter is small enough for the required time synchronization in UWB systems, it can be problematic for systems which provide coherent transmission and require tight frequency synchronization. In contrast, we show how AirShare can provide tight frequency synchronization enabling distributed coherent transmission. It is worth noting that PCO and AirShare are complementary schemes, as the former provides time synchronization, while the later provides frequency synchronization. In fact, these two systems can be combined to provide a complete distributed MIMO system which requires both time and frequency synchronizations. We plan to explore such designs in future work.

2.2 AirShare

AirShare enables independent wireless nodes to transmit coherently by sharing a reference clock transmitted over the wireless medium. In this section, we explain how AirShare works. We start with a description of how radios use a reference clock for transmission and reception, and why the existing system leads to incoherent transmissions. We then describe the structure of AirShare's shared reference signal. We follow with the circuit details and how AirShare can be incorporated in a wireless

node as an add-on module.

2.2.1 Why Do Wireless Nodes Have CFO?

Wireless signals are transmitted at a particular carrier frequency. The signal is up-converted from baseband to the carrier RF frequency at the transmitter and down-converted back to the baseband at the receiver. Both up-conversion and down-conversion are performed by a process called mixing, where the signal is multiplied by the carrier frequency. Thus, both transmitter and receiver need to generate the RF carrier signal with a precise and stable frequency.

Each node generates the carrier frequency as follows: The node has a local crystal that produces a low frequency sine wave, which is used as a reference clock. This reference clock is fed to a special circuit called a Phase Locked Loop (PLL), which uses it to generate the desired carrier frequency. For example, if the node is a Wi-Fi radio, the crystal might generate a 10 MHz sine wave, which the PLL then upconverts to a center frequency in the 2.4 GHz bands.

The key problem is that reference clocks on different nodes have slight differences in their frequencies, because different crystals naturally have different properties. Since the PLLs on different nodes lock to reference clocks with different frequencies, their output signals have different frequencies, and this leads to frequency offsets (CFO) between nodes.

It is important to realize that the CFO is not a constant value. Even minute variations of 0.1 degree in the temperature can cause CFO variations of a few hundred hertz [62, 112]. Similarly, noise in the supply voltage cause fast variations in the crystal's frequency [112].

2.2.2 How Does AirShare Work?

AirShare eliminates the root cause for incoherent transmission by ensuring that all nodes feed their PLL with the same reference clock that they receive over the wireless medium. Due to FCC regulations, however, one cannot simply transmit a 10MHz



Figure 2-1: **Illustration of AirShare’s Design.** Wireless nodes multiply the received signal by itself and extract the desired clock signal by applying a band pass filter centered at f_{ref} .

clock signal on the wireless medium. Also, one cannot simply up-convert the clock at the transmitter and down-convert it at the receiver, since this introduces a chicken-and-egg problem. Upconversion and downconversion to a band will require independent carrier generation at the transmitter and receiver. Since the reference signals for these independent carriers are generated by different crystals, they will have frequency offset relative to each other, leading to a frequency offset in the downconverted reference signal at different nodes.²

To address this problem, AirShare transmits a signal with a specific format such that it can be used to extract a reference clock without any other additional information. Specifically, AirShare transmits two single frequency tones (i.e., sine or cosine waves) separated by the desired clock frequency. These tones might be transmitted in the newly opened white spaces [158, 196], e.g., for a clock of 10 MHz, AirShare can send tones at 175 MHz and 185 MHz. Let us denote the transmitted tones by f_1 and f_2 , and the desired clock frequency by $f_{ref} = f_2 - f_1$, then the transmitted signal can be written as:

$$S_{tx}(t) = A_1 \cos(2\pi \cdot f_1 \cdot t) + A_2 \cos(2\pi \cdot f_2 \cdot t). \quad (2.1)$$

This signal passes over the wireless channel before reception, and the received wireless signal can be written as:

$$S_{rx}(t) = B_1 \cdot \cos(2\pi \cdot f_1 \cdot t + \phi_1) + B_2 \cdot \cos(2\pi \cdot f_2 \cdot t + \phi_2), \quad (2.2)$$

where B_1 , B_2 , ϕ_1 and ϕ_2 capture the channel impact.

²Note also that one cannot simply transmit a high frequency signal and generate the desired clock frequency from it using a PLL configured as a divider. This is because the high frequency PLL input will need to be passed through a very narrow (high-Q) band pass filter to reject any surrounding noise, and such filters are not practical at high frequencies.

To obtain the shared clock, each wireless node multiplies the received signal by itself and applies a band pass filter to extract the desired clock frequency. To see why this works, recall that the multiplication of two tones at different frequencies produces tones whose frequencies are the sum and difference of the original frequencies. Hence, after multiplying the received signal with itself the node obtains:

$$S_m(t) = [B_1 \cdot \cos(2\pi \cdot f_1 \cdot t + \phi_1) + B_2 \cdot \cos(2\pi \cdot f_2 \cdot t + \phi_2)]^2$$

Simplifying this equation results in:

$$\begin{aligned} S_m(t) = & \mathbf{B_1 B_2 \cos(2\pi \cdot (f_2 - f_1) \cdot t + (\phi_2 - \phi_1))} \\ & + B_1 B_2 \cos(2\pi \cdot (f_2 + f_1) \cdot t + (\phi_2 + \phi_1)) \\ & + \frac{B_1^2}{2} + \frac{B_1^2}{2} \cdot \cos(2\pi \cdot 2f_1 \cdot t + 2\phi_1) \\ & + \frac{B_2^2}{2} + \frac{B_2^2}{2} \cdot \cos(2\pi \cdot 2f_2 \cdot t + 2\phi_2) \end{aligned} \quad (2.3)$$

This signal includes a DC component, some high frequency components at $2f_1$, $2f_2$ and $f_1 + f_2$, and a component at $f_2 - f_1$ (highlighted in bold in the formula above) which is equal to the desired reference frequency f_{ref} . Hence, a simple band-pass filter centered at the reference clock frequency f_{ref} (e.g., 10 MHz) is used to extract the single-tone reference signal, as illustrated in Fig. 2-1.

The signal after the filter will be:

$$\begin{aligned} S_{ref}(t) &= B_1 B_2 \cos(2\pi \cdot (f_2 - f_1) \cdot t + (\phi_2 - \phi_1)) \\ &= B_1 B_2 \cos(2\pi \cdot f_{ref} \cdot t + \delta\phi). \end{aligned} \quad (2.4)$$

This signal is then used as an input to the node's PLL. Since all nodes feeds their PLL with a reference clock of the exact same frequency, they will have no CFO with respect to each other. Further, even if the frequency of the clock signal varies due to variation in the temperature or supply voltage at the clock transmitter, all nodes will see the same frequency variation in their reference clocks and hence stay coherent.

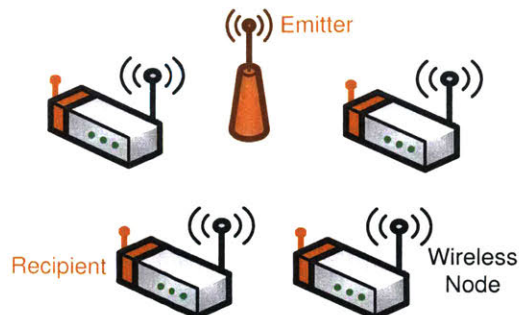


Figure 2-2: **AirShare's Network Topology.** An AirShare emitter transmits the reference signal. Wireless nodes that are equipped with AirShare recipient components receive the AirShare signal and extract the reference clock.

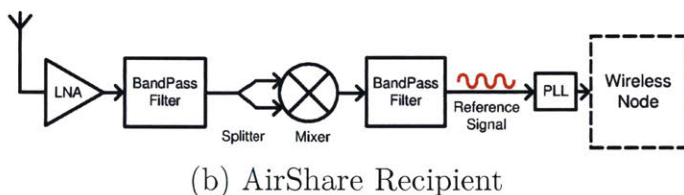
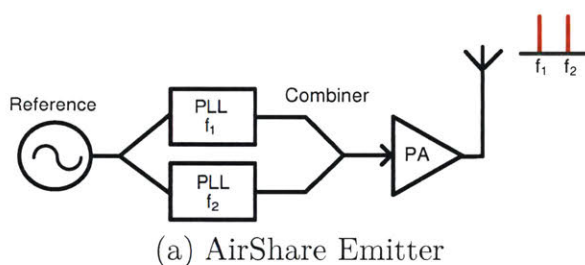


Figure 2-3: **AirShare's System Architecture.** The AirShare emitter includes two PLLs to generate two single tones separated by the desired frequency. The AirShare recipient mixes the received AirShare signal with itself and extracts the reference clock using a band pass filter.

2.2.3 AirShare's System Architecture

Architecturally, AirShare has two components: a clock emitter and a clock recipient. To enable a set of nodes to transmit coherently, one deploys an AirShare clock emitter in the network and equip each node with an AirShare clock recipient as illustrated in Fig. 2-2.

Emitter: Fig. 2-3(a) presents the AirShare emitter circuit. To transmit the AirShare signal, we start with a local oscillator (*i.e.*, a crystal) that generates a reference signal, and feed its output to two PLLs to generate two tones, f_1 and f_2 , that are separated

by the desired clock frequency f_{ref} . The two tones are then amplified using a power amplifier and transmitted on the wireless medium. Note that our signal does not occupy the entire band between f_1 and f_2 , it simply consists of two single-tones which are separated by the desired clock frequency.

Recipient: Fig. 2-3(b) presents the AirShare recipient circuit. The AirShare recipient receives the emitted signal. As is usual in RF radios, the received signal is passed to a low-noise amplifier (LNA) and the band of interest (from f_1 to f_2) is filtered out using a band pass filter. After filtering, the signal is mixed with itself and the desired reference clock is extracted using a band pass filter centered at the reference frequency (*e.g.*, 10 MHz), and input to a PLL whose output is fed to the wireless node.

Finally, we note a few points:

- First, AirShare is protocol and technology independent, and the AirShare recipient circuit can be incorporated in various radios (Wi-Fi, ZigBee, etc.) as an external clock that feeds the PLL.
- Second, the AirShare recipient circuit is simple and cheap and hence can be incorporated in low-end wireless nodes. In particular, the circuit does not need any analog to digital converters (ADC) or baseband processing. It is composed of cheap and off-the-shelf components such as amplifiers, band pass filters, a splitter, and a mixer. Further, the splitter, mixer and filters are all passive components and do not need power supply. Most of the power consumption is due to the LNA. An LNA that operates in the white space frequency range consumes only 7-10 mW [133], which is less than 0.1% of the power consumption of a Wi-Fi AP [76], and about 10% of the power consumption of a Zigbee node [174].
- Third, while AirShare transmits two tones separated by 10 MHz, these tones are single frequencies and hence occupy very little bandwidth. Others can transmit in the spectrum between the tones as long as they leave a buffer of about 1 MHz

around each tone. The band pass filter around f_{ref} in the recipient circuit can extract the desired reference clock without interference.

2.2.4 Dealing with Wireless Channel Impediments

Fading: Similar to other wireless systems, AirShare tones can suffer from multipath fading. In particular, either of the tones used by AirShare can be faded at wireless nodes using AirShare recipients. In such a case, the wireless node will not be able to participate in coherent transmission. Let p be the probability that any single tone is faded at one of the nodes participating in coherent transmission. Since the two tones are separated by a wide gap of 10 MHz, we can treat the fading at the two tones as independent. The probability of node failure, *i.e.*, the probability that either tone is faded, is therefore:

$$P(fail) = 2p(1 - p) + p^2 = 2p - p^2$$

For example, in a case where fading happens with 1% probability, the node will fail with a probability of 2%. This may not be tolerable for some applications. To reduce the failure rate, our solution is to transmit three tones instead of two tones. For example, the AirShare emitter can transmit tones at 175 MHz, 185 MHz and 195 MHz. In this case, the recipient can pick any combination of two tones. In particular, it can pick the two strongest tones it receives.³ Note that since these three tones are generated from the same reference crystal at the emitter, clocks generated using any combination of two tones will not have CFO with respect to each other. In this case, a node fails when two tones or all three tones are faded at that node. This happens with probability

$$P(fail) = 3p^2(1 - p) + p^3 = 3p^2 - 2p^3 \quad (2.5)$$

As can be seen from the equation, the failure rate is reduced quadratically with the addition of a single tone. For the same example as before, where the fading probability is 1%, the failure rate is reduced to 0.03%. Note that this failure rate is much lower

³The PLLs can use either 10MHz or 20MHz as their reference. They simply need to be configured to use a different multiplier ratio, a typical feature available in PLLs.

than the typical packet loss in Wi-Fi networks.

Multipath: AirShare continues to work in non-line-of-sight multipath scenarios. In this case, an AirShare recipient receives multiple delayed versions of the signal, each of which encounters a different channel. Note that adding multiple copies of signals at f_1 and f_2 , each with a different attenuation and phase, does not change the output frequency of AirShare. Specifically, similar to Eq. 2.4, the AirShare recipient still produces an output reference clock with frequency f_{ref} . We demonstrate empirically in §2.4.1 that AirShare’s performs well in non-line-of-sight multipath-rich scenarios.

Interference: AirShare can be affected by interference, just like other unlicensed band systems. This causes the PLL in the affected node to lose its lock to the reference. In this case, the lock detect circuitry (a standard feature in PLLs) signals the node to fall back to the on-board clock. This allows the node to operate correctly until the transient interference clears. A node suffering from interference however cannot take part in coherent transmissions and has to transmit separately as in today’s systems. Other nodes in the network can continue to participate in coherent transmission using AirShare.

2.2.5 Range of AirShare Emitter

The signal transmitted by the AirShare emitter propagates through the medium, is amplified by the receive chain at the AirShare recipient, and is then input to the PLL.

We can write the received power at the input of the PLL as:

$$\begin{aligned} \text{PLL Input Power} &= \text{Emitter Transmitted Power} \\ &\quad - \text{Path Loss} + \text{Recipient Rx Chain Gain} \end{aligned} \tag{2.6}$$

Given a particular AirShare implementation, we can determine the *emitter transmitted power* and the *recipient Rx chain gain* and substitute them in the above equation. We can then work back: measure the minimum required *PLL input power* so that it can lock on the signal, and estimate the maximum tolerable *path loss* using Eq. 2.6. Then, based on standard channel models, we can translate this maximum

tolerable path loss to the maximum propagation distance for AirShare.

Based on the above, the values in Eq. 2.6 can be computed as:

PLL Input Power: Based on our empirical measurements, the signal at the input of the PLL must be at least -17 dBm to enable the PLL to lock.

Recipient Rx Chain Gain: The total gain of the recipient receiver chain (antenna, amplifiers, mixer and filters) is 15 dB.

Emitter Transmitted Power: The signal strength from the emitter is determined by FCC regulations. For the whitespaces, the FCC limits the total transmitted power including the transmitted antenna gain to 18.6 dBm in any 100kHz band.⁴ Since AirShare uses two different tones separated by 10 MHz, it can use double this transmitted power for a total transmitted power of 21.6 dBm.

Path Loss: Substituting these numbers in Eq. 2.6, we can compute the maximum tolerable path loss as 53.6 dBm.

In order to translate this into a distance, we use the standard link budget/path loss (PL) model [159]:

$$\text{PL at distance } d = \text{PL at reference dist. } d_0 + 10n \log_{10} \frac{d}{d_0} \quad (2.7)$$

where n is the path loss exponent, and PL is measured in dBm.

We use the standard Friis equation to determine the path loss at reference distance $d_0 = 1m$. For a frequency of 175 MHz (wavelength of 1.72m), the path loss at reference distance d_0 can be computed as $20 \log_{10}(\frac{1.72}{4\pi}) = 17.3$ dB. Substituting these in Eq. 2.7, the maximum allowable distance is 65 m (≈ 210 ft) in a line-of-sight system ($n = 2$), and 21 m (≈ 70 ft) in a non-line of sight system spread across a large floor ($n = 2.76$) [159]. This means that a single AirShare emitter can span all nodes in a 210 ft radius circle around it in a line-of-sight scenario, and 70 ft radius around it in a non-line-of-sight scenario.

⁴The FCC limits conducted power to 12.6 dBm, and allows antenna gain up to 6 dBi [93]

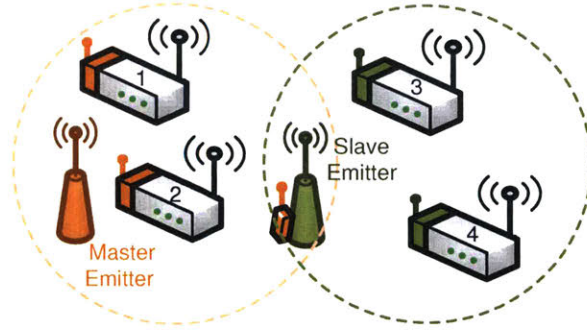


Figure 2-4: **AirShare is Scalable to Large Networks.** In a scenario where nodes are far from the master AirShare emitter, AirShare uses a slave emitter to regenerate the clock signal. Nodes can synchronize to either the master or slave emitter, and still achieve tight synchronization with each other.

2.2.6 Scalability

The previous section evaluated the radius of a deployment across which a single AirShare emitter can synchronize all nodes. We now explain how we can synchronize nodes deployed over a region larger than the radius of coverage of a single emitter. Naively, one might think that we can use multiple emitters in different locations to guarantee that each wireless node receives the AirShare signal. However, this solution does not work because each AirShare emitter has a different reference oscillator (i.e., crystal) that it uses to generate its AirShare signal. As a result, signals from different AirShare emitters will have CFO with respect to each other.

Instead AirShare uses a hierarchy of clock emitters. To understand how our proposed solution works, consider the scenario in Fig. 2-4. Here, we want to eliminate CFO between any pair of four wireless nodes, where nodes 3 and 4 are far from nodes 1 and 2. The master emitter generates the AirShare signal from a local oscillator. Nodes 1 and 2 are in the range of this emitter and use the signal transmitted from the master emitter to generate their reference clocks. However, nodes 3 and 4 are not in the range of the master, and cannot directly receive the AirShare signal. Hence, we use another emitter called a **slave emitter** to transmit the AirShare signal to them.

The architecture of a slave emitter is similar to that of the master. However, there is one key difference. A slave emitter does not use a local oscillator as its reference. Instead, it uses an AirShare recipient which listens to the master's signal to generate

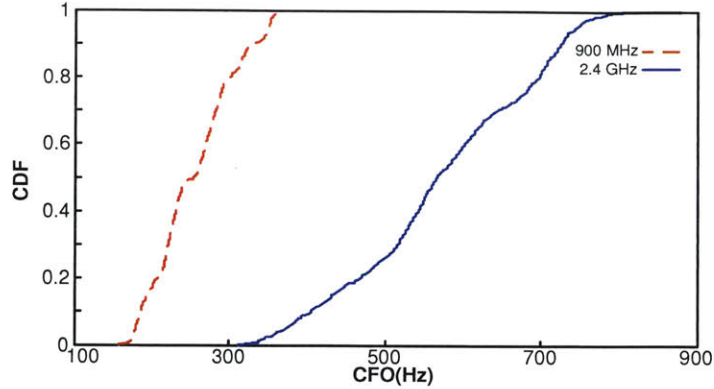
its reference. The slave then produces its two tones using this recipient generated reference. In order to avoid interference between master and slave, the frequencies of the AirShare signal generated by the slave need to be different from the frequencies generated by the master. For instance, the slave emitter can use tones at 620 MHz and 630 MHz, while the master emitter uses tones at 175 MHz and 185 MHz.

Note that tones generated by the slave emitter have no CFO with respect to the master emitter because the reference clock for the slave is generated by using the master emitter signal. Consequently, the reference clock extracted from the master emitter signal by wireless node 1 will be synchronized with the reference clock extracted from the slave emitter signal by wireless node 3. Cooperating wireless nodes can pick either the master or the slave signal to extract the shared clock, depending on which of the two are in their radio range. Further, one may cascade multiple master-slave emitters where each slave acts as a master for the next emitter in the cascade. In §2.4.1, we empirically show that this mater-slave design works in practice.

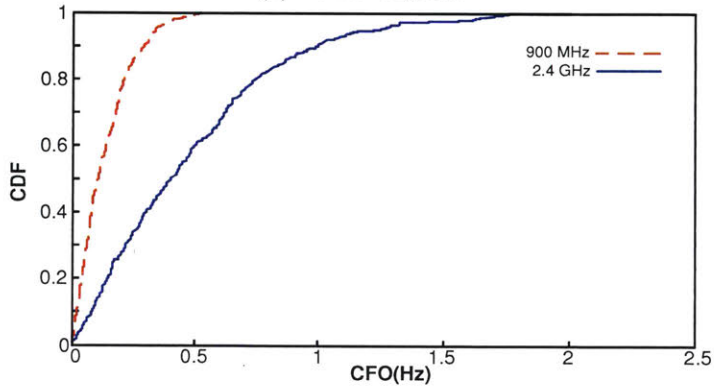
2.3 AirShare Implementation

We built a prototype of the AirShare emitters and recipients as described in §2.2.3 using off-the-shelf components. For the emitter, we used the following components: Fury Jackson GPSDO as a local oscillator, Analog Devices ADF4350 as PLLs, Mini-Circuits TVA-R5-13 as a power amplifier, Laird Technologies EXB-164-BN as an antenna. For the recipient, we used: Laird Technologies EXB-164-BN as an antenna, Mini-Circuits ZLW-1 as a mixer, Mini-Circuits SBP-10.7+ as bandpass filter around 10MHz, Mini Circuits SHP-175+ as a highpass filter and Mini-Circuits SLP-200+ as a lowpass filter to create a bandpass filter around 175-185 MHz.

We integrate the recipient subsystem with USRP by connecting the output of the recipient to the 10 MHz external clock input of the USRP N210 device.



(a) Local Crystal



(b) AirShare (Note the difference in the scale of the x-axis)

Figure 2-5: CFO between pairs of nodes at carrier frequencies of 2.4 GHz and 900 MHz: (a) Independent clocks and (b) AirShare.

2.4 Empirical Evaluation of AirShare

We evaluate AirShare in an indoor testbed with line-of-sight and non-line-of-sight scenarios. The testbed spans $10\text{m} \times 10\text{m}$. All experiments in this section are run with USRP nodes that use OFDM, a 1500 byte packet length, and 10MHz bandwidth. The experiments use a total of 6 USRPs. For each evaluation, we run 500 experiments for a variety of the nodes locations.

2.4.1 Eliminating CFO between nodes

A key promise of AirShare is that it can address the CFO problem. We verify if AirShare delivers on this promise.

Line-of-Sight Experiments. We place an AirShare emitter in one location in

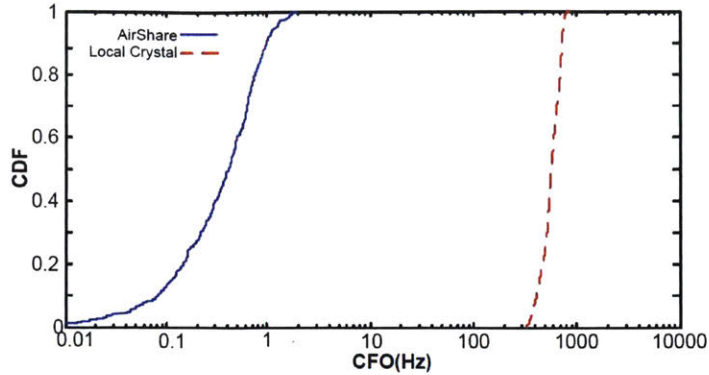


Figure 2-6: **CFO between pairs of nodes at 2.4 GHz carrier frequency for independent clocks and AirShare.** Figure shows that AirShare reduces the CFO by multiple orders of magnitude.

the testbed. We place two USRP nodes equipped with AirShare recipients at two random locations in the testbed, with one acting as a transmitter and the other as a receiver. The transmitter transmits packets consisting of OFDM symbols. The receiver receives these packets, and computes its CFO with respect to the transmitter using the traditional correlation-based OFDM CFO estimation algorithm [166]. We repeat the experiment for two carrier frequencies: 2.4 GHz and 900 MHz, and for a variety of USRP nodes and transmitter receiver locations. We repeat each run both with the USRPs operating using their internal crystals and with the USRPs using the AirShare signal as a reference.

Fig. 2-5 plots the CDF of the observed CFO for both 2.4 GHz and 900 MHz carriers. The graph in Fig. 2-5(a) correspond to using the internal crystals, whereas the graph in Fig. 2-5(b) corresponds to using AirShare. The CFO at 900MHz is smaller than at 2.4 GHz because the PLL multiplies the 10 MHz reference clock by an appropriate factor to deliver the carrier frequency (90 and 240 for 900 MHz and 2.4 GHz, respectively).

The figure shows that when the USRPs operate with their internal crystals, their CFO varies in the range 310Hz–880Hz for a 2.4GHz carrier and 160Hz–370Hz for a 900MHz carrier. Fig. 2-6 plots CFO between pairs of nodes at 2.4 GHz carrier frequency for both AirShare and independent clocks. The figure shows that AirShare

reduces the CFO by two to three orders of magnitude. Specifically, with AirShare, the median and 95th percentile CFO at 2.4 GHz are 0.4 Hz and 1.24 Hz, respectively.

To put these numbers in context, consider the accumulated phase error with and without AirShare for a single 1500B packet at the lowest OFDM rate (BPSK, 1/2 rate) used by Wi-Fi. This packet takes $\approx 2ms$. Thus, with AirShare the median and 95th percentile phase error across this packet are 0.005 radians, and 0.016 radians, which are negligible and have no effect on coherence. In contrast, in the absence of AirShare, the phase errors across the packet would be between 3.9 to 11.1 radians (*i.e.*, over a 180 degree change in phase across a packet), and hence the signals are very far from being combined coherently within the packet [153].

(b) Non-Line of Sight Experiments: We repeat the above experiment but this time we place the AirShare emitter in a different room such that it has no line of sight to the other nodes in the testbed. Fig. 2-7 shows the CDF of the measured CFO in the presence of non-line-of-sight channels for a carrier frequency of 2.4 GHz. The median and 95th percentile CFO in this case are 0.4 Hz and 1.3 Hz respectively, showing that AirShare achieves the same tight performance as in the case of line-of-sight channels.

(c) Impact of using a master-slave deployment: In §2.2.6, we described how AirShare can scale to large deployments by using a master-slave emitter design. Here, we evaluate the impact of such design on AirShare’s performance.

We place an AirShare master emitter and an AirShare slave emitter within radio range of each other. As before, we configure two USRP nodes equipped with AirShare recipients as a transmitter and receiver, respectively. The AirShare recipient on the transmitter receives its clock signal from the master emitter, and the AirShare recipient on the receiver receives its synchronization signal from the slave emitter. As before, we compute the CFO between the transmitter and the receiver.

Fig. 2-7 plots the CDF of the observed CFO (the dashed line). It shows that the increase in CFO compared to a scenario with only one AirShare emitter is less than 0.1 Hz. Thus, cascading AirShare emitters using a master-slave approach maintains the same orders of magnitude reduction in CFO in comparison to local clocks. The figure also indicates that the system can operate with a longer cascaded chain of

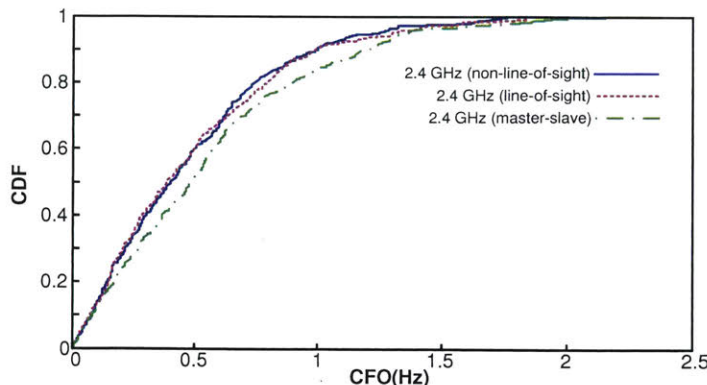


Figure 2-7: **CFO between pairs of AirShare-equipped nodes at 2.4 GHz carrier frequency.** Figure shows that AirShare ensures tight synchronization of oscillators both in line-of-sight and non-line-of-sight scenarios, as well as in scenarios with master-slave emitters.

master-slave emitters. This is because each emitter in the chain acts as a master for the next emitter. Hence, the worst-case increase in CFO is bounded by 0.1 Hz multiplied by the length of the cascaded chain, which stays small for a chain with a few hops.

2.4.2 Enabling Coherent Transmission

We examine whether AirShare can enable independent nodes to transmit coherently. We place an AirShare emitter and four USRP nodes at random locations in our testbed. One of the USRPs acts as a receiver and the other three as transmitters. The transmitters concurrently transmit random data to the receiver using BPSK. We repeat the experiment with two different schemes: (a) transmitters and receiver using independent oscillators driven by their local crystals, (b) transmitters and receiver using AirShare.

Fig. 2-8 plots the received constellation diagram in two scenarios: Fig. 2-8(a) corresponds to each node using its local crystal, and Fig. 2-8(b) corresponds to the nodes using AirShare.

If the transmitters were coherent with each other, then their signals would combine in a predictable manner across time. That is, when transmitters 1, 2 and 3 all transmit the symbol “+1”, the receiver would always receive the same constellation point

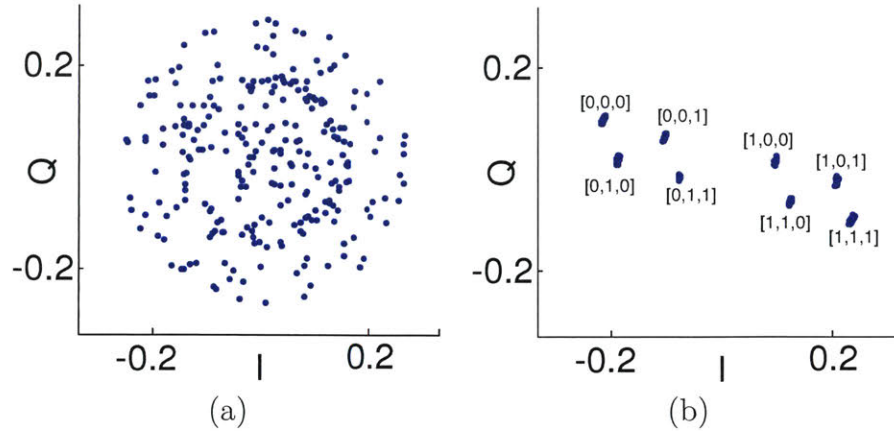


Figure 2-8: **Received constellation points for three nodes transmitting BPSK in different scenarios: (a) Independent clocks, and (b) AirShare. Each point in (b) is labeled with the associated combination of transmitted bits.** AirShare enables coherent transmission. Without AirShare, the signals from multiple transmitters do not have a constant phase relationship with each other. As a result, the received constellation points for a given combination of transmitted signals vary over time. In contrast, with AirShare, the signals from the different transmitters are coherent. Hence, the received constellation points for a given combination of transmitted signals stay constant over time.

(and similarly for other combinations of transmitted symbols). In contrast, if the transmitters are not coherent with each other, the same transmitted symbols would rotate relative to each other, and combine in different ways across time producing different received constellation points.

We see this latter effect in Fig. 2-8(a). The transmitters and receiver oscillators have significant CFO relative to each other when using their local crystals. As a result, the constellation points produced by joint transmission from the different nodes are smeared uniformly across space. In contrast, when AirShare is used, the received constellation has 8 distinct points (Fig. 2-8(b)), corresponding to each of the three transmitters transmitting a “+1” or “-1”. This is because each combination of transmitted symbols from the three transmitters combines in a predictable manner at the receiver. This experiment demonstrates visual evidence that AirShare provides coherent transmission across wireless nodes.

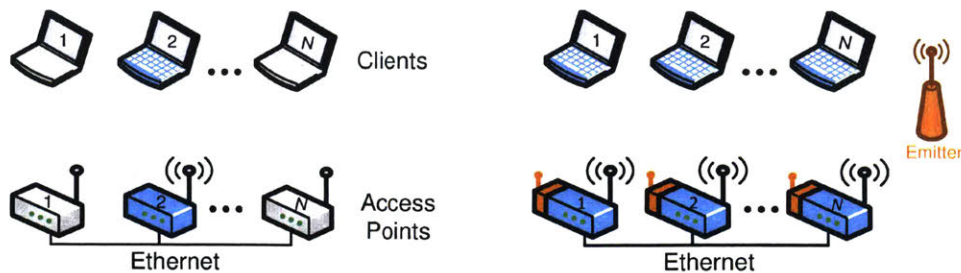


Figure 2-9: **Traditional AP Deployments (left) versus Distributed MIMO (right)**. A blue node indicates an active transmitter or receiver. With traditional Wi-Fi, only one AP transmits at any time in a given channel. In contrast, with AirShare, multiple APs transmit to multiple clients at the same time in the same channel, thereby scaling network throughput with the number of APs.

2.5 Applications of AirShare

Multiple high-gain cooperative PHY protocols assume coherent transmission and hence can benefit from AirShare. We demonstrate AirShare’s versatility by explaining how it can be used to build two cooperation protocols: distributed MIMO and distributed rate adaptation.

Our aim here is to demonstrate that a core functionality in these protocols can be derived seamlessly using AirShare.

2.5.1 Distributed MIMO with AirShare

MIMO beamforming allows a single MIMO node with n antennas to transmit n concurrent packets to n independent clients without interference. Let x_i be the data intended for client i , \vec{x} the vector $[x_1 \dots x_n]$, and M the channel matrix from the n -antenna MIMO transmitter to n clients. The MIMO node transmits $M^{-1}\vec{x}$. The clients receive the vector $\vec{y}_i = MM^{-1}\vec{x}$. Thus, each client receives $y_i = x_i$, and obtains its intended data with no interference from data intended for the other clients. By transmitting n independent data units in a unit of time using a unit of spectrum, the system achieves a multiplexing gain of n , which translates to a throughput gain that increases linearly with the number of antennas. For a more formal description of MIMO beamforming we refer the reader to [181].

Distributed MIMO beamforming enables independent transmitters to act as if they

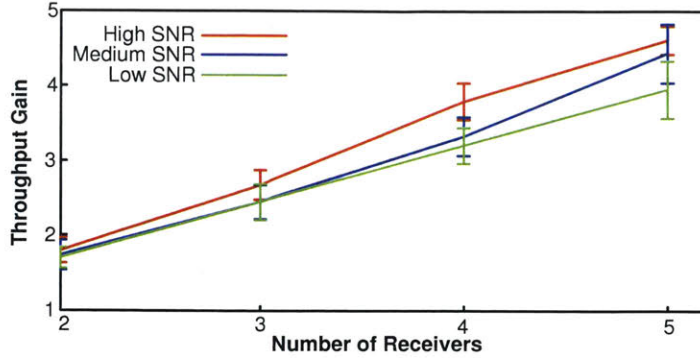


Figure 2-10: **Distributed MIMO using AirShare.** AirShare’s throughput gain increases linearly with the number of transmitter-receiver pairs in the network.

were antennas on a single virtual MIMO node. Hence, n single antenna transmitters can use distributed MIMO to deliver n packets to n independent clients, using the same above equations. However for n independent transmitters to act as if they were antennas on a single node, they need to transmit coherently without CFO between them.

While the theory of distributed MIMO has been around for many years, practical implementations have emerged recently [56, 153]. These systems transmit training signals to estimate the rotation due to CFO. They then correct for the impact of the CFO on the channel estimates from different transmitters by applying a time-dependent inverse rotation to the transmitted symbols. These algorithms are OFDM specific, and deeply intertwined with the details of the baseband system.

In contrast, with AirShare, the nodes have a shared reference, which eliminates the need for phase tracking and compensation altogether. It frees the designer from having to think through the interaction of OFDM and coherent transmission, and provides a technology-independent design. In the next section, we reproduce the distributed MIMO system in [153] but after eliminating any phase tracking and compensation procedures, and providing the nodes with AirShare clock recipients. The results show that the system continues to operate correctly delivering the gains of distributed MIMO but without the need for phase tracking or compensation.

Evaluation of Distributed MIMO with AirShare We place an AirShare emitter in our testbed. We also place USRPs with AirShare recipients to act as APs and

clients in our testbed. Similar to the scenario in [153], we assume that transmitters use the back-end Ethernet to obtain the packets intended for all clients. Also like [153], we use SourceSync [151] to estimate when the nodes should transmit jointly their beamformed data.

We evaluate distributed MIMO with AirShare in three different SNR regimes: low (5-10 dB), medium (10-16 dB), and high (> 16 dB). Since USRPs cannot perform carrier sense due to high software latency, we evaluate traditional 802.11 by scheduling each transmitter so that it gets its fair share of the medium. We repeat the experiment for different node placements and different number of transmitter-receiver pairs.

Fig. 2-10 plots the throughput gain obtained by distributed MIMO using AirShare as a function of the number of transmitting APs, for different SNR ranges. We see that AirShare enables the wireless network throughput to scale with the number of transmitter-receiver pairs, for a gain of $3.95 - 4.61\times$ across the range of SNRs. This is because, with traditional 802.11, only one transmitter-receiver pair is active at any time irrespective of the number of transmitters. In contrast, distributed MIMO enables all transmitters to transmit jointly to their desired receivers without interfering with each other, and achieving throughput proportional to the number of active transmitters. This shows that, with AirShare, distributed MIMO systems can achieve results similar to prior work [153] but without the need for phase tracking and compensation algorithms.

2.5.2 Distributed Rate Adaptation for Wireless Sensors

Sensors typically support only a single modulation scheme, such as on-off keying, BPSK, or QPSK [86]. The modulation supported is low rate so as to ensure that the sensors can communicate even when channel conditions are adverse. Further, sensors avoid supporting high rate (dense) modulations such as 16-QAM, 64-QAM *etc.*, because these modulations require linear transmitter power amplifiers that consume significant power. As a result, wireless sensors do not utilize the wireless channel efficiently. In particular, wireless sensors cannot take advantage of a good channel to send at dense modulation that packs multiple bits into each transmitted symbol.

One can imagine exploiting channel conditions through distributed rate adaptation across the network to overcome the absence of the ability of any single node to adapt its rate. Prior work [185] has proposed such distributed rate adaptation in the context of RFID networks. Specifically, multiple RFID nodes can transmit simultaneously and the receiver receives a collided transmission. Consider, for instance, the case in Fig. 2-8(b), where 3 nodes, each using BPSK (*i.e.* +1 or -1), transmit simultaneously. Let the channel between the nodes and the sink be h_1 , h_2 and h_3 respectively. In such a case, the receiver will receive one of 8 points, $h_1+h_2+h_3$, $h_1+h_2-h_3$, $h_1-h_2+h_3$, etc. As can be seen from the figure, the receiver can decode the individual transmissions from all three transmitters using a single collided transmission, if the channel conditions are sufficiently good. If not, it can simply continue to receive additional transmissions and combine these multiple receptions till it can decode the transmitted signals. Such a system will effectively achieve a distributed rateless code across the nodes in the RFID network. We refer the reader to [185] for full details of the transmission protocol, and the receiver algorithm to decode the distributed sparse rateless code.

The protocol from [185] described above is designed specifically for RFID networks. RFIDs, however, do not have independent oscillators; they transmit by reflecting a carrier signal from a single device, and hence do not suffer from CFOs relative to each other. In contrast, general wireless devices, *e.g.* sensors, have independent oscillators, which they use for transmission. As a result, transmissions from different nodes rotate relative to each other, and collide in different ways across time. Consequently, as can be seen in Fig. 2-8(a), the joint constellation formed by the collisions experiences different rotation and scaling over time, preventing the receiver from decoding the transmitted bits from multiple collision receptions.

AirShare eliminates the problem of differing frequency offsets across sensors. In particular, when each sensor is equipped with an AirShare recipient, their oscillators are driven by the shared wireless clock and hence they do not have any offset relative to each other. The sensors can therefore perform joint transmission and enable the sink to decode, similar to the case of RFID networks described above.

Evaluation of Distributed Rate Adaptation with AirShare: As in other exper-

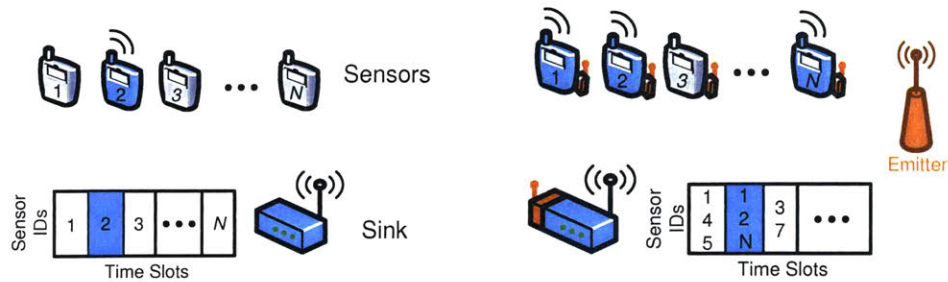


Figure 2-11: **Traditional Sensor Networks (left) versus Distributed Rate Adaptation (right)**. A blue node indicates an active sensor. With traditional sensor networks, only one sensor transmits at a time. In contrast, AirShare-equipped sensors perform distributed rate adaptation by transmitting simultaneously. This enables decoding data from multiple sensors in a single joint transmission, and consumes significantly fewer than N slots to transmit data from all N sensors.

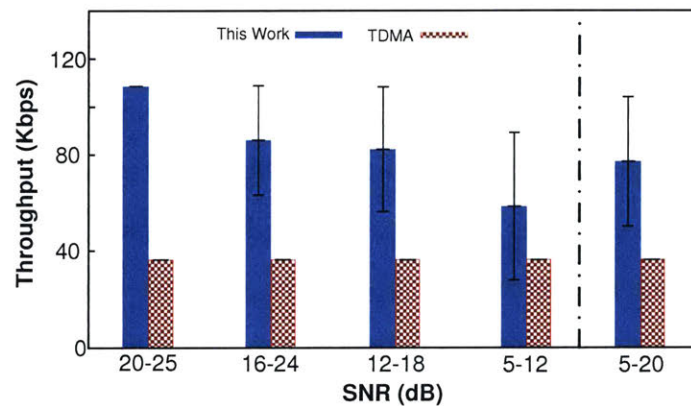


Figure 2-12: **Channel Quality versus Throughput for Distributed Rate Adaptation**. AirShare enables distributed rate adaptation for wireless sensors.

iments, we deploy an AirShare emitter in the testbed. We deploy 6 USRPs equipped with AirShare implementing ZigBee and acting as sensors, and one USRP with AirShare acting as a sink. We run 100 experiments for a variety of node locations. We compare distributed rate adaptation with AirShare with TDMA where only one sensor transmits at a time, and the different sensors transmit one after the other. We evaluate both schemes in various SNR ranges.

Fig. 2-12 plots the throughput of distributed rate adaptation using AirShare, and of traditional TDMA, for different SNR ranges. Distributed rate adaptation achieves $1.64 - 3\times$ the throughput of TDMA. Since TDMA cannot exploit good channel conditions to increase its transmission rate, its throughput is constant independent of SNR. In contrast, distributed rate adaptation can exploit good channel conditions by allowing the receiver to decode multiple simultaneous transmitters from a single collision. Since the receiver decodes more simultaneous transmitters as the SNR increases, the throughput gain of distributed rate adaptation increases with the average SNR of the network, similar to how the throughput of traditional rate adaptation increases with increasing link SNR.

2.6 Discussion

This chapter presents AirShare, a system that can synchronize multiple nodes using a reference signal that they receive over the wireless medium. AirShare has a simple, low-cost, low-power architecture that easily integrates with today's wireless transceivers. AirShare provides an abstraction that multiple different wireless nodes are driven by the same reference clock, and hence enables straightforward implementation of a variety of distributed PHY algorithms, including distributed MIMO and distributed rate adaptation for sensor networks. AirShare's synchronization primitive can serve as a building block that brings a large body of distributed information theoretic schemes closer to practice.

Chapter 3

Agile mmWave Networks

The ever-increasing demands for mobile and wireless data have placed a huge strain on wireless networks [75, 87, 182]. Millimeter wave (mmWave) frequency bands address this problem by offering multi-GHz of unlicensed bandwidth, $200\times$ more than the bandwidth allocated to today's WiFi and cellular networks [148, 155]. They range from the 24GHz ISM band all the way to hundreds of GHz [8, 16]. They are expected to play a central role in dealing with increased multimedia traffic, the introduction of new high data-rate applications such as virtual reality, and the anticipated surge in IoT wireless devices [37, 39, 79]. This role has been cemented with new standards that incorporate mmWave technologies into 5G cellular networks [107, 122, 155], and 802.11 wireless LAN [116].

Millimeter wave radios however do not play well with mobile devices or dynamic environment, a key challenge that has been emphasized in the standards [116, 144]. Specifically, mmWave signals attenuate quickly with distance; hence they need to use highly directional antennas to focus their power on the receiver. Luckily, due to their small wavelength (millimeter scale), it is possible to pack hundreds or thousands of antennas in a small area, creating an array with many antennas, and hence a very narrow beam, as shown in Figs. 3-1(a) and (b). Yet, since the beam is very narrow, communication is possible only when the transmitter's and receiver's beams are well aligned. Current solutions for aligning the beams scan the entire space, trying various beam alignments until they find the best one. This process can take up to

several seconds [168, 203]. Such a long delay makes the deployment of mmWave links infeasible in wireless networks, where the access point has to keep realigning its beam to switch between users and accommodate mobile clients.

To understand the problem, consider a phased array with many antennas. Its beam-width can be a few degrees or even smaller. The naive approach to finding the best alignment would have the transmitter and receiver scan the 3D space with their beams to find the direction of maximum power, as shown in Fig. 3-1(b). The receiver has to repeat the scan for each choice of beam direction on the transmitter side. Thus, the complexity of this exhaustive search is $O(N^2)$, where N is the number of possible beam directions. To speed up the search, the 802.11ad standard decouples the steering at the transmitter and receiver. In particular, the transmitter starts with a quasi-omnidirectional beam, while the receiver scans the space for the best beam direction. The process is then reversed to have the transmitter scan the space while keeping the receiver quasi-omnidirectional [116, 201] (see §3.5.1 for the details). This approach reduces the search complexity to $O(N)$. Still, for a beam of a few degrees, the delay can be hundreds of milliseconds to seconds [168, 203], which would easily stall realtime applications.

But, can one identify the best alignment without scanning the space of all possible signal directions? In principle, "yes". There is much past work that shows that mmWave signals travel along a small number of paths (e.g., the direct path from transmitter to receiver and a few reflections) [51, 155]. This means that the space of possible signal directions is sparse. One would hope to use the sparse recovery theory to estimate the direction of the best alignment using a logarithmic number of measurements [69, 80, 109], hence avoiding excessive delays.

Problem Formalization: To be more concrete, let us formalize the problem. The objective of beam alignment is to measure the signal power along each spatial direction. Let \mathbf{x} be an N -element vector that denotes the signal along various spatial directions. Since in practice the signal arrives only along few directions K , we can say that \mathbf{x} is K -sparse. Let, h_i be the signal at the i^{th} antenna, as shown in Fig. 3-1(c). Based on the standard antenna array equation [191], we can write $\mathbf{h} = \mathbf{F}'\mathbf{x}$, where \mathbf{F}'

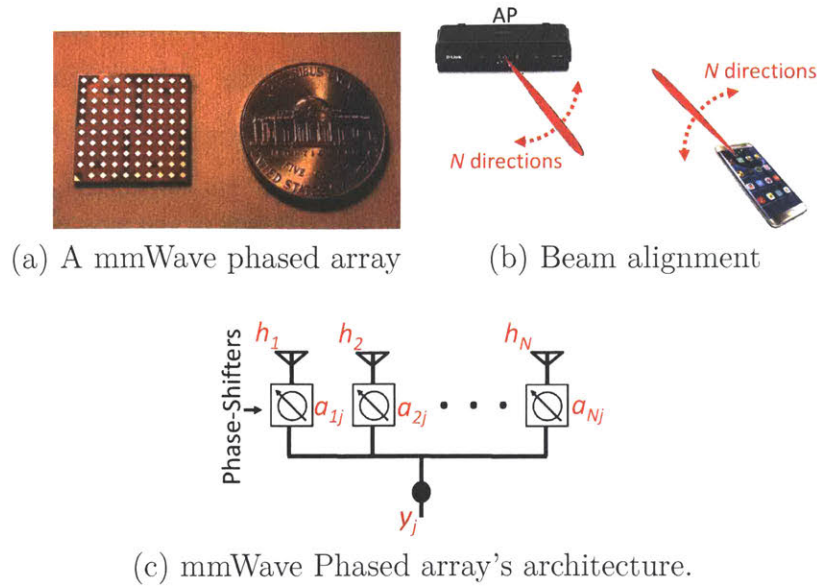


Figure 3-1: **mmWave Communication** (a) An example of mmWave phased arrays where hundreds of antennas are packed in a small area [61]. (b) mmWave radios need to find the best alignment between the transmitter's and receiver's beams to establish a communication link. (c) mmWave phased arrays use a set of phase shifters to steer the beam electronically.

is the inverse Fourier transform matrix. We can steer the antenna beam by applying a phase shift to each antenna $a_i = e^{-j2\pi\phi_i}$ (see Fig. 3-1(c)). For each setting of the phase shifters, we can measure the received power as $y_j = |\mathbf{a}_j \mathbf{F}' \mathbf{x}|$, where the notation $|\cdot|$ refers to the magnitude of the signal (i.e., the power),¹ and \mathbf{a}_j is a vector whose elements are the applied phase shifts. In 802.11ad, each measurement corresponds to sending a special frame. Each time a frame is sent, the signal incurs a random phase due to the Carrier Frequency Offset (CFO) between the transmitter and receiver [116, 153] (see §3.3.1 for details). Thus, one cannot compare the phase of two measurements; only the magnitude of the measurements is relevant. Since we want to know the power along each spatial direction, the problem can be formulated as:

$$\text{estimate each } |x_i|, \text{ given measurements } y_j = |\mathbf{a}_j \mathbf{F}' \mathbf{x}|.$$

Of course, one way to solve this problem is to use N measurements, each time setting

¹The power is the magnitude square; hence knowing the magnitude is the same as knowing the power.

\mathbf{a} to one row of the Fourier matrix. This corresponds to measuring one direction every time, as proposed in 802.11ad. Alternatively, one could leverage that the vector \mathbf{x} is sparse, and hope to solve the problem in a logarithmic number of measurements. Unfortunately, however, using off-the-shelf algorithms like compressive sensing or the sparse FFT does not work since neither of them deal with the scenario where the measurements return the magnitude of the complex signal, i.e., the presence of the $|\cdot|$ term [69, 80, 100].

Our Design & Analysis: This chapter introduces Agile-Link, a new protocol for fast beam alignment in mmWave networks. Agile-Link provably finds the best alignment in a logarithmic number of measurements. At a high-level, it works as follows: Instead of creating a narrow beam and sampling the power along one spatial direction each time, Agile-Link manipulates the phase shifters to create multi-armed beams, which can sample multiple spatial directions simultaneously (see Fig 3-2(a)). When sampling multiple directions simultaneously, one cannot tell which direction has produced the resulting power. Agile-Link however uses a carefully-designed combination of such multi-armed beams, which quickly provide enough information to identify the signal power along all directions, as we detail in §3.3.2. We formally analyze Agile-Link and prove that it can deliver the best alignment in $O(K \log N)$ measurements, where K is the number of paths traveled by the signal. Since K is typically 2 or 3 paths [51, 155], Agile-Link can significantly reduce the beam alignment delay.

Agile-Link has additional important features:

- Agile-Link is compatible with the 802.11ad protocol, i.e., a Agile-Link device can work with a non-Agile-Link device to find the best alignment while using the 802.11ad protocol. In this case, the Agile-Link device finds the best alignment on its side in a logarithmic number of measurements whereas the traditional 802.11ad device takes a linear number of measurements.
- When both transmitter and receiver are equipped with Agile-Link, they can coordinate the search and find the best transmit and receive beam directions in $O(K^2 \log N)$ (as opposed to $O(K^2 \log^2 N)$). They can do so without using

quasi-omnidirectional beams, which are known to reduce the quality of the alignment [111, 143].

- Finally, by the results of [54, 149], our measurement complexity of $O(K \log N)$ is asymptotically *optimal* for small K .

Implementation & Empirical Results: We have evaluated Agile-Link empirically using mmWave radios, each equipped with a phased array that has 8 antennas. We have implemented the design as a daughterboard for the USRP software radio, which enables easy manipulation of mmWave signals using standard GNU-radio software. We also use simulations to explore its scaling behavior to large arrays with hundreds of antennas, which are expected in the future [66]. We compare Agile-Link with two baselines: an exhaustive scan of the space to find the best beams, and the quasi-omnidirectional search proposed in the 802.11ad standard. Our evaluation reveals the following findings.

- In comparison with the exhaustive search, Agile-Link reduces the search time by one to three orders of magnitude for array sizes that range from 8 antennas to 256 antennas. In comparison to the quasi-omnidirectional search, Agile-Link reduces the number of measurements by $1.5\times$ to $16.4\times$ for the same range of array sizes. In particular, for large arrays with 256 antennas, Agile-Link reduces the alignment delay from over a second to 2.5ms. We note that our results are based on a phased array which has linear number of phase shifters and only one transmit-receive chain. One may reduce this delay further by using a more complex hardware architecture or designing inference algorithms that synthesize the inherent WiFi or GPS information with RF-localization in order to quickly steer the mmWave radios' beams.
- The quasi-omnidirectional search yields poor performance in scenarios with multi-path effects. This is because using a quasi-omnidirectional antenna allows the signals along different paths to combine destructively, which yields low power and prevents accurate detection of the best signal direction. Further, due

to imperfections in the quasi-omnidirectional patterns [111, 143], some paths can get attenuated and hence this approach can choose the wrong direction to align its beam. In contrast, Agile-Link performs well both in single path and multipath scenarios.

3.1 Related Work

(a) mmWave Solutions & Research: Research on fast beam alignment for mmWave can be divided into two classes: empirical and simulation-based. Past empirical work has demonstrated the large delays incurred during beam alignment [168, 203]. It also proposed failover protocols that switch to the next best beam when the current beam becomes blocked [105, 170]. This approach however assumes that the signal propagation paths are known a priori and hence one can quickly switch to a failover direction.

Much of the previous work on fast beam alignment is simulation-based. Most of this work proposes enhancements to the standard that impose a form of hierarchy to speed up the search [53, 123, 126, 180, 187, 197, 201]. However, in practice, hierarchical search requires feedback from the receiver to guide the transmitter at every stage of the hierarchy, which incurs significant protocol delay.

Our work is closest to past work that leverage compressive sensing to speed up the search for best beam alignment. The work in [48, 83] requires a more complex architecture with a quadratic number of phase shifters, and multiple transmit receive chains (typically 10 to 15 [48]). Despite being less constrained, the best known results for such complex architecture can guarantee a logarithmic number of measurements *only* for scenarios with no multi-path (K is strictly 1), and only for the average case error [48]. In comparison, Agile-Link can provably find the best alignment in a logarithmic number of measurements even when the signal experiences multipath, and its guarantees apply to the worst case behavior.

Some past theoretical work applies the standard compressive sensing and assumes it can correctly obtain the phase of the measurements [97, 154]. This approach does

not work with practical 802.11ad or cellular radios because it ignores CFO (Carrier Frequency Offset) which corrupts the phase of the measurements [153]. Note that correcting for CFO across measurement frames is neither possible in the current 802.11ad standard nor easy. For example, a small offset of 10 parts per million at such high frequencies can cause drastic phase misalignment in less than hundred nanoseconds.

Finally, some companies such as TP-Link and SiBeam [35, 66, 157, 164, 176, 199] offer mmWave systems but they take a long time to steer the beam or require complex hardware, making them unsuitable for mobile clients [168, 203]. Also, some research on mmWave focuses on point-to-point links for Data Centers [77, 106, 202] or cellular and WiFi applications [105, 145, 168, 170, 203]. These implementations typically use a horn antenna to direct the beam, which requires mechanical steering and is unsuitable for mobile links.

(b) Sparse Recovery Theory: The theoretical problem we consider falls under “sparse phase retrieval” [117, 119]. Generally, the goal is to recover an approximation of a K -sparse vector, \mathbf{x} , from M measurements of the form $|\mathbf{B}\mathbf{x}|$. The presence of the absolute value is what makes this problem different from the usual compressive sensing [69, 80] and sparse FFT [98, 99, 100, 108, 109]. In our context, we have an extra restriction that the matrix \mathbf{B} is of the form $\mathbf{A}\mathbf{F}'$, where \mathbf{F}' is the inverse Fourier transform and all entries in \mathbf{A} have unit magnitude, i.e., $|a_{ij}| = 1$ for all i, j . To the best of our knowledge, this form has not been considered before.

Some of our proof techniques are inspired by past work on sparse FFT, particularly the work in [98, 99] which used boxcars filters for sparse Fourier transform algorithms. However, the technical development of our proofs is different due to the leakage between multiple beam arms, which requires extra layers of randomization. Furthermore, our results and problem are different due to the restrictions on our measurements that do not exist in sparse FFT.

(c) Massive MIMO Beamforming: Massive MIMO [162, 163] has many antennas, but the signal from each antenna can be received and manipulated independently. In

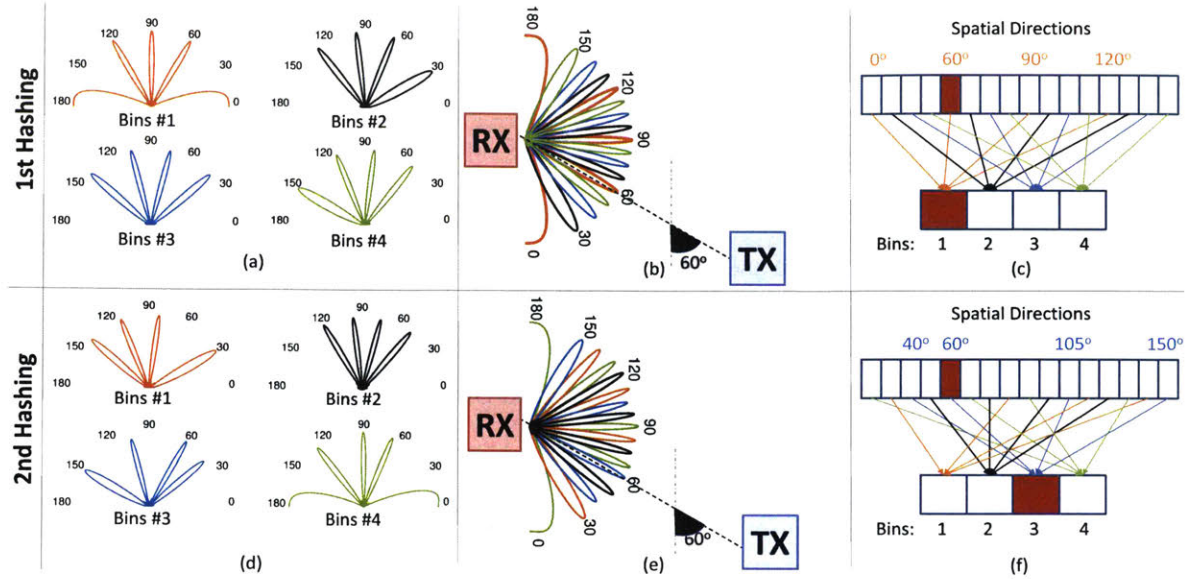


Figure 3-2: Illustrative Example of Agile-Link's Algorithm

contrast, a mmWave phased array receives only the combined signal from all its antennas. Thus, beamforming techniques in massive MIMO (and standard MIMO) do not apply to mmWave phased array.

3.2 Illustrative Example

(a) **Single Path Example:** We start by explaining the intuition of our algorithm. Consider a case where the transmitter is at a 60° angle with respect to the receiver. For clarity, assume a 2D setting. The same argument can be extended to 3D. We would like the receiver to detect that the best alignment is along the 60° angle without scanning the space.

Agile-Link avoids the need to scan all spatial directions by using multi-armed beams, which simultaneously sample the signal along multiple directions. Say for example, that there are 16 possible directions in space, i.e., $N = 16$. Agile-Link can sample all of these directions using 4 multi-armed beams, each covering $N/4 = 4$ directions in space. Fig. 3-2(a) shows four such multi-armed beams, and Fig. 3-2(b) shows how together they cover the whole space of directions. Such set of multi-armed beams operates like a hash function, where $N = 16$ directions are hashed into 4 bins,

and each bin covers $N/4 = 4$ distinct directions. The value of the bin represents the combination of the signals that hash into it. For example, if the signal is coming along the 60° direction and 60° hashes to bin number 1, then only bin one will have energy whereas the other bins will have no energy, as shown in Fig. 3-2(c). Thus, one can ignore directions that hash to bins 2,3, and 4, and focus only on directions that hash to bin 1. This significantly reduces the search space to the directions that hash to the first bin (i.e., the first multi-armed beam).

At this stage, we know that the signal could have come from the directions covered by the first bin i.e., 0° , 60° , 90° and 120° . But we do not know which among them is the correct direction. Thus, we change the hash function and try again. To do so, we use a second set of multi-armed beams which together hash the whole space of directions into a set of bins. The hash however is randomized with respect to the previous hash so that directions that got hashed together are unlikely to hash together again. Figs. 3-2(d,e,f) show an example of hashing the spatial directions into bins after randomizing the multi-armed beams and hence randomizing which directions map into which bins. The first bin now collects energy along 30° , 80° , 110° and 140° . Since the signal is arriving along 60° , it will be captured by the third bin which is represented in blue in Fig. 3-2(f). Hence, in this second hashing, only the energy of the third bin will be large. This suggests that the signal arrived along one of the directions that mapped to the third bin which in this case are 40° , 60° , 105° and 150° . Since the 60° direction is the only common candidate from both the first hashing and the second hashing, Agile-Link picks it as the direction of the signal. Thus, Agile-Link is able to find the correct direction from which the transmitter's signal arrives without having to scan all possible directions.

(b) Multipath Example: But, what if the signal arrives along multiple paths? For example, in Fig. 3-3, the signal arrives along three directions: p_1 , p_2 and p_3 . Paths p_1 and p_2 are higher power than p_3 because they are significantly shorter; and the best beam alignment is along the direction of p_1 which delivers the highest power.

Let us compare two beam alignment algorithms: Hierarchical Search and Agile-Link. As explained in §6.1, multiple past proposals use a hierarchy of beams to speed

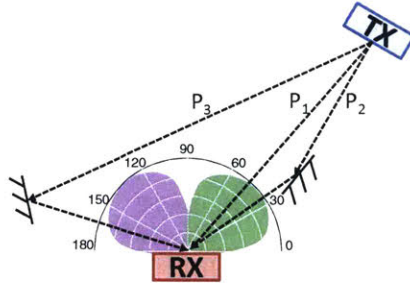


Figure 3-3: **Example of Hierarchical Search in the Presence of Multipath.**

up the search for the best alignment [126, 180, 197]. Proposals differ in the details, but they use wide beams at the top of the hierarchy – e.g., they might start with two wide beams, check which beam returns more power, then zoom in on the part of the space that returns more power and use narrower beams to explore it. Like Agile-Link Hierarchical Search needs only a logarithmic number of measurements. The problem is that using wide beams means nearby signal directions would collide within the same beam. Since RF signals are waves, the colliding signals can combine destructively to cancel each other’s power. In our example, p_1 and p_2 have close directions and thus will collide in the wide beam. Say that p_1 and p_2 have opposite phases, they will cancel each other’s power, making it look as if the signal from the left-half of the space (i.e., directions 90° to 180°) had more power. As a result, Hierarchical Search will zoom in on those directions and explore them with narrower beams until it finds path p_3 , which is the worse alignment out of the three signal directions. Note that this failure mode can be quite common and does not require the phases of p_1 and p_2 to be exact opposite; it is sufficient that they point away from each other so that the resulting power degradation makes it look as if p_3 were higher power.

In contrast, Agile-Link does not use wide beams, where nearby directions are bound to collide. Agile-Link uses multi-armed beams where the arms in each beam can be randomized. Even if two paths collide in one multi-armed beam, our randomization ensures that the same two paths will not collide in other multi-armed beams, as formalized in §3.3.2. At a high level, the system operates as multiple randomized hash functions, which can be provably resolved to recover all signal directions even in the presence of multipath.

3.3 Agile-Link

This section describes Agile-Link in detail. For clarity, we describe the algorithm assuming only the receiver has an antenna array whereas the transmitter has an omni-directional antenna. In §3.3.3, we extend it to the case where both transmitter and receiver have antenna arrays.

3.3.1 Problem Statement

Recall that the problem is defined as follows: Let \mathbf{x} be a K -sparse N -element complex vector that denotes the signal along various spatial directions. The objective is to estimate the power (i.e., magnitude) of the signal along each direction, $|x_i|$, using a small number of measurements of the form $y_j = |\mathbf{a}_j \mathbf{F}' \mathbf{x}|$, where \mathbf{F}' is the inverse Fourier matrix, and \mathbf{a}_j is a vector of phase shifts, $|a_{ij}| = 1$, that are under our control.

Before describing our solution to this problem, it is important to understand why the phases of the measurements are not usable. Every measurement involves sending a frame from transmitter to receiver. Since the oscillators on the transmitter and receiver always experience some CFO (Carrier Frequency Offset) [153], the signal of each frame incurs an additional unknown phase shift. Further, this phase shift changes across frames. Correcting the CFO across measurement frames is not supported in the 802.11ad standard [116]. Furthermore, such a correction will be very hard due to the high frequencies of mmWave. For example, a small offset of 10 parts per million at such frequencies can cause a large phase misalignment in less than hundred nanoseconds.

3.3.2 Agile-Link's Algorithm

Agile-Link works in two stages. First, it randomly hashes the space into bins (using multi-armed beams) such that each bin collects power from a range of directions. Second, it uses a voting mechanism to recover the directions that have the power. Below, we describe these two stages in detail.

A. Hashing Spatial Directions into Bins

Agile-Link hashes the signal along various directions to bins using multi-armed beams. Let us refer to the arms in each multi-arm beam as the *sub-beams*. Let R be the number of sub-beams in each beam, B the number of bins in each hash function, and L the number of hash functions.

Each setting of the phase shifter vector, \mathbf{a} , creates a different beam pattern and the resulting measurement $y = |\mathbf{a}\mathbf{F}'\mathbf{x}|$ will correspond to the power in the directions covered by the beam pattern.

So how do we create multi-armed beams? It should be clear that, given the structure of the measurements, we can create a beam that points in one direction, s , by setting \mathbf{a} to the s -th row in the Fourier matrix. Thus, to create a multi-armed beam, Agile-Link divides the vector \mathbf{a} into R segments each of length N/R *i.e.*, $\mathbf{a}_{[1:N/R]}$, $\mathbf{a}_{[N/R+1:2N/R]}$, \dots , $\mathbf{a}_{[(R-1)N/R:N]}$. Each segment then sets its sub-beam towards a different direction. This is done by setting the segment $\mathbf{a}_{[i:i+N/R]}$ to the corresponding segment in the desired row of the Fourier matrix. Formally, if an index i belongs to the r -th segment pointing towards the direction s^r , then $a_i = (\mathbf{F}_{s^r})_i \cdot e^{-j2\pi t_r/N}$, where t_r is a random integer between $0, \dots, N-1$, and $(\mathbf{F}_{s^r})_i$ refers to the i -th entry in row s^r in the Fourier matrix. The term $e^{-j2\pi t_r/N}$ results in a phase shift of the sub-beam without changing its direction, and it simply helps in the proof.

Due to the properties of the Fourier transform, the sub-beam created by each segment will be wider than a single beam created by the full array by a factor of R , so each sub-beam covers R adjacent directions. Since there are R such sub-beams, the multi-armed beam created by this setting of the vector \mathbf{a} will cover R^2 directions. Now, if we wish to hash the space of directions into B bins, then each will cover R^2 directions and hence $B = N/R^2$.

But how do we pick the directions of the sub-beams in each multi-armed beam? The best scenario is when the sub-beams in each multi-armed beam are well spaced so that the leakage from their side-lobes is minimized. Fig 3-4(a) shows an example of well-spaced sub-beams, where we have a multi-armed beam with two sub-beams

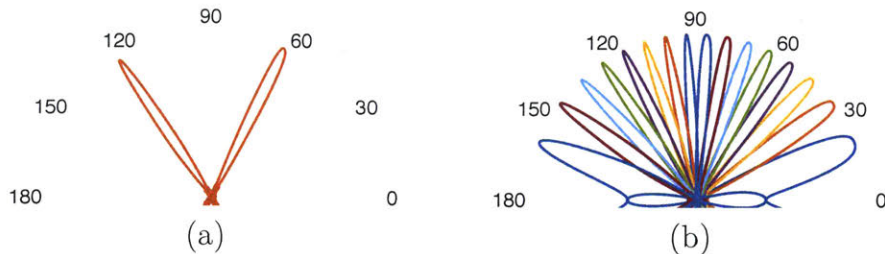


Figure 3-4: **Hashing Beam Patterns** (a) Agile-Link hashes well spread directions into a bin; (b) All the bins of Agile-Link's hash function where the same color corresponds to directions in the same bin, and hash function covers the whole space.

directed 60° apart. In this case, the beam pattern will hash directions that are 60° apart into the same bin. By shifting the direction, we can then create all different bins in the hash function, as shown in Fig 3-4(b) where each color corresponds to a bin in the hash function. Formally, the above process is achieved by setting the direction of the r -th segment in bin b to be equal to $s_b^r = Rb + rP$, where $P = N/R$ is the spacing between two sub-beams corresponding to the same bin.

The question that remains is how do we randomize the hashing process to make sure that two large signals are not hashed to the same bin every time. Ideally, we can solve this problem by randomly permuting the entries in of the vector \mathbf{x} . Physically, however we cannot permute \mathbf{x} . Instead, we leverage a nice property of the Fourier transform that says that we can pseudo-randomly permute the *input* of the Fourier transform (i.e., the entries of the vector \mathbf{x}) by pseudo-randomly permuting its output (i.e., the entries of the vector $\mathbf{F}'\mathbf{x}$) [98, 99, 108]. In the simplest setting, suppose that we want to permute the input samples according to:

$$\mathbf{x}'(f) = \mathbf{x}(\sigma f), \quad (3.1)$$

where σ is chosen at random for an appropriate distribution, and all operations on indices are done modulo N . This is equivalent to permuting the output $\hat{\mathbf{x}} = \mathbf{F}'\mathbf{x}$ according to:

$$\widehat{\mathbf{x}'}(t) = \hat{\mathbf{x}}(\sigma^{-1}t). \quad (3.2)$$

where σ^{-1} is the inverse of σ modulo N . This enables us to “randomize” the positions of large entries in \mathbf{x} by rearranging the entries of $\mathbf{F}'\mathbf{x}$. This result is very useful since

we can easily permute $\mathbf{F}'\mathbf{x}$ in our measurements by simply permuting the elements in the vector \mathbf{a} , i.e., by permuting the phase shifts applied to the phase shifters on each antenna.

More generally, let us use the vector $\mathbf{y}_{1 \times B} = |\mathbf{A}_{B \times N} \mathbf{F}'\mathbf{x}|$ to refer to the B measurements performed as part of a hash function, where \mathbf{A} is a matrix of phase shifts. To permute this hash function, we measure $\mathbf{y}_{1 \times B} = |\mathbf{A}\mathbf{P}'\mathbf{F}'\mathbf{x}|$, where we use the term $|\cdot|$ to refer to the magnitude of the individual elements in the vector, and the matrix \mathbf{P}' is a generalized permutation matrix.² That is, our phase shifter matrix is equal to $\mathbf{A}\mathbf{P}'$ (note that each entry of the latter matrix has a unit magnitude, i.e., it represents proper phase shifts). This is equivalent to measuring $|\mathbf{A}\mathbf{F}'\mathbf{P}\mathbf{x}|$, for the corresponding generalized permutation matrix \mathbf{P} .

B. Recovering the Directions of the Actual Paths

After hashing the spatial directions into bins, Agile-Link discovers the actual directions of the signal using a voting scheme where each bin gives votes to all directions that hash into that bin. After few random hashes, the directions that have energy will collect the largest number of votes which allows Agile-Link to recover them. Unfortunately, simply directly applying this voting approach does not work well because the side-lobes of the beams create leakage between the bins and hence a strong path in one bin can leak energy into other bins which corrupts the voting process. To overcome this problem, Agile-Link uses a form of soft voting that takes into account the leakage between the bins.

Specifically, Agile-Link models the beam patterns (example of which are shown in Fig. 3-4(a) and Figs. 3-2(a) and (d),) as a *coverage function* $I(b, \rho, i)$ that indicates the coverage of the direction i by the beam corresponding to the bin b , assuming the

² \mathbf{P}' is a generalized permutation matrix if each row or column of \mathbf{P}' contains exactly one non-zero entry, and that entry has unit magnitude. We use the matrix \mathbf{P}' defined as in [108]: the i -th column of \mathbf{P}' contains the value $\omega^{a\sigma i}$ in the row $\sigma(i-b)$, where $\omega = e^{2\pi j/N}$ and a, b, σ are randomly chosen parameters. This has the effect of rearranging the vector \mathbf{x} by moving the entry x_i to $x_{\rho(i)}$ for $\rho(i) = \sigma^{-1}i + a$, and multiplying it by $\omega^{bj+\sigma ba}$. Note that the role of the latter multiplier is similar to that of the phase multiplier described in §3.3.2(A), namely it helps with the proof. See the appendix for a more detailed description of the process.

indices are permuted by ρ . It is defined as:

$$I(b, \rho, i) = |\mathbf{a}^b \mathbf{F}'_{\rho(i)}|^2 \quad (3.3)$$

where \mathbf{a}^b is the vector defining the settings of the phase shifters corresponding to bin b .

If we hash into B bins, we will have B such patterns and collect B measurements $\mathbf{y}_{1 \times B}$ corresponding to the bins. After taking the magnitude squared of each measurement, we can estimate the energy of the signal coming from direction i as:

$$T(i, \rho) = \sum_{b=0}^{B-1} y_b^2 \times I(b, \rho, i), \quad (3.4)$$

If the estimate $T(i, \rho)$ exceeds a predefined threshold T then we conclude that there is a signal coming from direction i . Otherwise we conclude that there is no such signal.

C. Performance Analysis

A detailed performance analysis is provided in the Appendix. Here we note the main theorem and its implications.

Theorem 3.3.1. *Suppose that the vector \mathbf{x} has at most K non-zero entries, at that the energy x_i^2 of each non-zero entry x_i is at least $1/K$. Furthermore, suppose that N is a prime. There exists a setting of parameters T , R and $B = O(K)$, so that for each candidate direction i , we have that:*

- *If $x_i \neq 0$ then $T(i, \rho) \geq T$ with probability at least $2/3$*
- *If $x_i = 0$ then $T(i, \rho) < T$ with probability at least $2/3$*

The probability of correctness can be amplified by performing several (L) random hashes and aggregating the results. There are multiple ways of performing the aggregation. A simple approach is to use “hard voting”, i.e., conclude that a signal is coming from a direction i if the signal from that direction has been detected by the

majority of hashes. By Chernoff bound, this approach reduces the probability of incorrect detection from $1/3$ to $e^{-C''L}$ for some constant C'' . By letting $L = O(\log N)$, we can compute correct estimates for all *all* indices i , with the probability of failure at most $1/N$. The algorithm uses $BL = O(K \log N)$ measurements and its running time complexity is $O(NK \log N)$.

The algorithm can be also used to provide estimations of the values of $|x_i|^2$'s. The guarantees are provided by the following theorem. Note that no assumptions about the sparsity of \mathbf{x} are required, although the guarantees are meaningful only for x_i 's whose magnitude is large enough. This makes the estimate resilient to the presence of small amounts of noise at all coordinates.

Theorem 3.3.2. *Suppose that N is a prime. There exists a setting of parameters R , $B = O(K)$, and a constant $C > 1$ so that for each candidate direction i , we have:*

$$\Pr[|x_i|^2/C - \|\mathbf{x}\|_2^2/K \leq T(i, \rho) \leq C|x_i|^2 + \|\mathbf{x}\|_2^2/K] \geq 2/3$$

3.3.3 Antenna Arrays on Both Transmitter and Receiver

We now extend the above model and algorithm to the case where both transmitter and receiver have antenna arrays. In this case, each measurement can be written as:

$$y_{1 \times 1} = |\mathbf{a}_{1 \times N}^{rx} \mathbf{F}' \mathbf{x}_{N \times 1}^{rx} \mathbf{x}_{1 \times N}^{tx} \mathbf{F}' \mathbf{a}_{N \times 1}^{tx}| \quad (3.5)$$

where \mathbf{a}^{rx} and \mathbf{a}^{tx} are vectors corresponding to a setting of the phase shifters, \mathbf{F}' is an $N \times N$ inverse Fourier transform matrix and \mathbf{x}^{rx} and \mathbf{x}^{tx} are sparse vectors representing the angle of arrival at the receiver and the angle of departure at the transmitter respectively. Our goal is to recover \mathbf{x}^{rx} and \mathbf{x}^{tx} from several measurements y .

We can reduce this problem to the problem solved in the earlier section. Specifically, we make $B \times B$ measurements of the form:

$$\mathbf{Y}_{B \times B} = |\mathbf{A}_{B \times N}^{rx} \mathbf{F}' \mathbf{x}_{N \times 1}^{rx} \mathbf{x}_{1 \times N}^{tx} \mathbf{F}' \mathbf{A}_{N \times B}^{tx}| \quad (3.6)$$

where \mathbf{A}^{rx} is the phase shift matrix as in Theorem 3.3.1, and \mathbf{A}^{tx} is its transpose. Let \mathbf{A}_i^{rx} be the i -th column of \mathbf{A}^{rx} and \mathbf{A}_j^{tx} be the j -th row of \mathbf{A}^{tx} . Furthermore, let $y_i = \sum_j Y_{i,j}$. We observe that

$$\begin{aligned}
y_i &= \sum_j Y_{i,j} \\
&= \sum_j |\mathbf{A}_i^{rx} \mathbf{F}' \mathbf{x}^{rx} \mathbf{x}^{tx} \mathbf{F}' \mathbf{A}_j^{tx}| \\
&= \sum_j |\mathbf{A}_i^{rx} \mathbf{F}' \mathbf{x}^{rx}| |\mathbf{x}^{tx} \mathbf{F}' \mathbf{A}_j^{tx}| \\
&= |\mathbf{A}_i^{rx} \mathbf{F}' \mathbf{x}^{rx}| \left(\sum_j |\mathbf{x}^{tx} \mathbf{F}' \mathbf{A}_j^{tx}| \right) \\
&= |\mathbf{A}_i^{rx} \mathbf{F}' \mathbf{x}^{rx}| C
\end{aligned}$$

where C is a constant independent of i . Therefore, we can recreate the measurements of the form needed by Theorem 3.3.2 from the B^2 measurements provided by the matrix \mathbf{Y} . In this way we can test, for each i , whether x_i^{rx} is non-zero, and each test is correct with probability at least $2/3$. By repeating the process $L = O(\log N)$ times, we can detect each non-zero entry with high probability. The total number of measurements is $B^2 L = O(K^2 \log N)$.

Finally, while we described the algorithm for 1D antenna arrays, the algorithm holds for 2D arrays as well. We simply need to apply the hash function along both dimensions of the array. For an $N \times N$ antenna array, the complexity will be $O(K^2 \log N^2)$ and hence will continue to scale logarithmically with the number of antennas in the array.

3.3.4 Algorithm Implementation

In practice randomizing the phases in the measurements and permutation matrices is not necessary, and therefore in our implementation we set $b = 0$ and $t_r = 0$. We also drop the assumption that N is prime. Finally, we use soft voting instead of the hard voting. Specifically, we estimate of the strength of the signal along direction i as $S(i) = \prod_{l=1}^L T_l(i, \rho)$, where $T_l(i, \rho)$ is defined as in Equation 3.4, for the l -th permutation. The soft voting approach uses more information about the

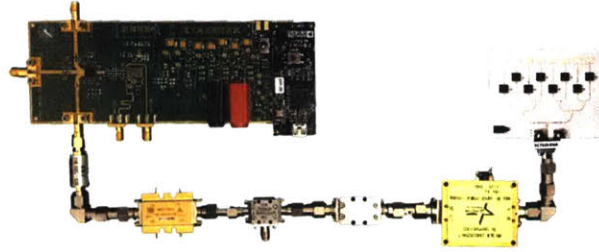


Figure 3-5: **Agile-Link Platform:** The figure shows the phased array and mmWave radio we built to operate as a daughterboard for the USRP software radio.

measurements than hard voting, and hence its practical performance is better. We extract the significant coefficients of the signal x_i by selecting the indices i with the largest values of $S(i)$.

3.4 Agile-Link Implementation Platform

We have implemented Agile-Link by designing and building a full-fledged mmWave radio capable of fast beam steering, as shown in Fig.3-5. The radio operates in the new 24GHz ISM band and serves as a daughterboard for the USRP software radios. Its physical layer supports a full OFDM stack up to 256 QAM. Our implementation addresses critical system and design issues that are described below.

3.4.1 Heterodyne Architecture

Millimeter Wave hardware is significantly more expensive than GHz hardware. Thus, we leverage a heterodyne architecture where the mmWave signal is first taken into an intermediate frequency of a few GHz, before the I and Q (real & imaginary) components are separated. Such a design reduces the number of components that need to operate at very high frequencies (e.g., mixers, filters, etc) and replaces them with components that operate at a few GHz, which are much cheaper.

The architecture of Agile-Link's receiver is shown in Fig 3-6(b). The first block of our design is a mmWave phased array which allow us to steer the beam electronically. The phased array consists of antenna elements where each element is connected to a phase shifter component. The outputs of the phase shifters are combined and fed to a

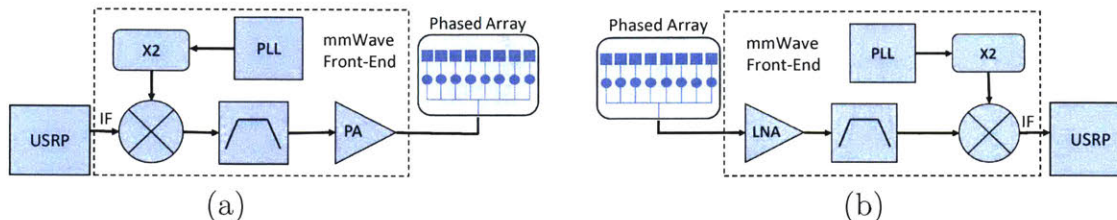


Figure 3-6: **Agile-Link's Architecture.** The figure shows block diagrams for both Agile-Link's transmitter(a) and receiver(b).

single mmWave front-end. Then, the mmWave front-end downconverts the mmWave signal to an intermediate frequency (IF) and feeds it to the daughterboard on the USRP which samples it and passes the digitized samples to the UHD driver.

The mmWave front-end consists of a low-noise amplifier (LNA) to amplify the received signal, a band-pass filter to remove the out-of-band noise and interference signals, and a mixer to downconvert the mmWave signal to the IF signal. The main step of our front-end design is the generation of a high-frequency local oscillator (LO), which is used to generate the mmWave carrier. This signal can be obtained from a phase locked loop (PLL) working at mmWave frequencies. Unfortunately, to the best of our knowledge, such PLLs are not available commercially. To overcome this difficulty, we use a component called a frequency doubler. Specifically, instead of using a mmWave PLL, we use a PLL working at much lower frequency and feed its output to a frequency doubler to generate an LO signal at a mmWave frequency. Agile-Link's transmitter shown in Fig. 3-6(a) follows a similar architecture.

We have built the design in Fig. 3-6 using off-the-shelf components. For the mmWave low-noise amplifier (LNA) and power amplifier (PA), we use Hittite HMC-C020 and Quinstar QLW-2440, respectively. For the mmWave mixer, we use Marki M1R-0726MS. To generate local oscillator (LO) signals, we use Analog Devices ADF5355 PLL and Hittite HMC-C035 frequency doubler. The phased array is designed using HFSS software and fabricated on printed circuit board (PCB) using Rogers substrate. The phased array includes 8 antenna elements separated by $\frac{\lambda}{2}$, where each element is connected to a Hittite HMC-933 analog phase shifter. We use Analog Device AD7228 digital-to-analog converters (DAC) and Arduino Due micro-controller board to digitally control the phase shifters.

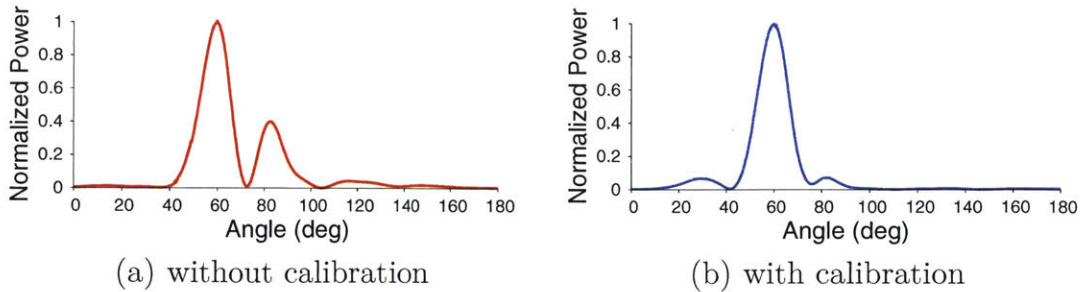


Figure 3-7: **Phased Array Radiation Pattern.** The figure plots the phased array radiation pattern for (a) before and (b) after calibration when the beam was steered at 60 degrees. The figure shows that without calibration, the phased array has an unwanted large sidelobe at 80 degrees. However, the calibration improves the radiation pattern of the phased array by minimizing the sidelobe.

To support flexible development, we implemented the mmWave radio as a daughterboard for the USRP X310 software radio which serves as an IF and baseband signal processing unit. This enables easy manipulation of mmWave signals using standard GNU-radio software and allows us to build an OFDM stack that supports up to 256 QAM.

3.4.2 Phased Array Calibration

Millimeter Wave phased arrays require a one-time calibration. To see the importance of such calibration, Fig.3-7 plots patterns of Agile-Link’s phased array before and after calibration when the beam was steered at 60 degrees. The figure shows that without calibration, the phased array has an unwanted large sidelobe at 80 degrees. Such a sidelobe reduces the power beamed to the receiver and can further cause interference to other mmWave nodes.

The need for calibration stems from the non-linearity of phase shifters. Specifically, phase shifters are analog components used to change the phase of an RF signal. The phase shift introduced by a phase shifter is a function of its control voltage. This function is typically provided by the manufacturer as a plot which shows the phase shift for each control voltage. However, once these phase shifters are mounted on the phased array board, they perform differently due to finite size of the antenna array, variation in antenna’s feeding network, etc. Hence, it is required to calibrate individ-

ual phase shifters after mounting them on the board. To do so, we fix the input of all phase shifters except one, which we vary to scan the whole range of input. We empirically observe the phase shift resulting from each input and create a table that maps a phase shifter’s input to the resulting phase.

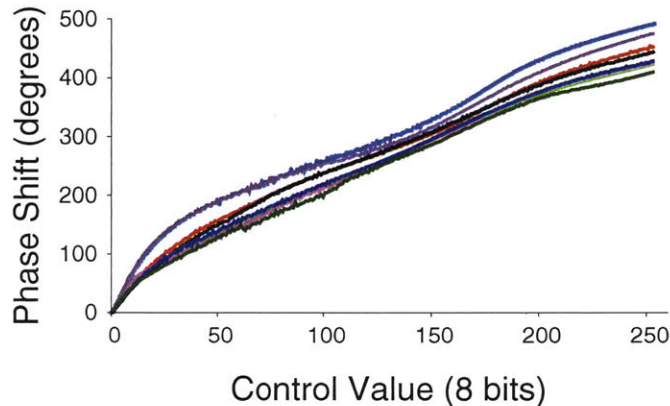


Figure 3-8: **Phase Shifters’ Performance.** The figure plots the amount of phase shift introduced by each phase shifter (mounted on Agile-Link’s phased array board) versus the value of its 8-bit control line. The figure shows that for the same control value, phase shifters may have up to 100 degrees difference in the amount of phase shift which they introduce.

It is important to realize that each individual phase shifter has to be calibrated, and results in a different calibration table. Fig. 3-8 plots the calibration functions for eight phase shifter in our array. The figure shows that for the same control value, phase shifters may have up to a 100 degree difference in the amount of phase shift they introduce. Hence, instead of the specification provided by manufacturer, we use the empirical calibration tables to adjust the phase shifters’ control voltage.

To better understand the impact of such calibration on improving the phased array radiation pattern, we measure and compare the pattern of the phased array before and after the calibration process. We run 120 experiments for a variety of beam directions on 4 different phased arrays (two transmitter and two receiver arrays). We run these experiments in an anechoic chamber designed for antenna measurement. To measure the radiation pattern of the phased array on the transmitter, we place a receiver that uses a horn antenna with a narrow beam facing toward transmitter. The transmitter phased array is then mounted on a pole equipped with a precise step motor. We setup the phase shifters of the array to steer at a specific angle, then

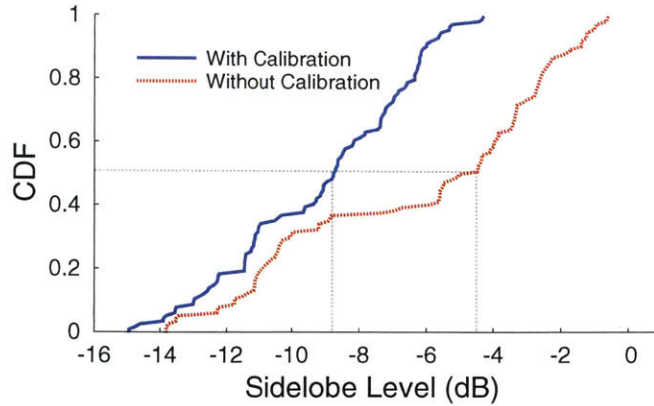


Figure 3-9: **Performance of Phased Array Calibration.** CDF of sidelobe level relative to the main beam with and without calibration. The figure shows that Agile-Link’s calibration significantly reduces the radiation outside of the main beam.

we rotate the phased array antenna from 0-180 degrees while the horn antenna is receiving the transmitted signal. We measure the received power at each angle which gives us the radiation pattern. We perform the same experiments to calibrate the receiver’s phased arrays while the horn antenna is used for transmitting.

For each measured pattern, we calculate the sidelobe level (SLL), which is the sidelobe power relative to the peak power of the main beam. Higher SLL numbers imply that the antenna array leaks more power outside the main beam, which impedes its beamforming and creates interference.

Fig. 3-9 plots a CDF of SLL for our phased arrays before and after calibration. The figure shows that without calibration, the median SLL is -4.48dB, while calibration reduces it to -8.76dB. This shows that calibration improves the directionality of the phased array’s beam which improves communication range and data rate, and reduces interference.

3.4.3 Radio Performance:

To test Agile-Link’s ability to deliver high data rates and long range using phased arrays, we measure the effective SNR of the received signal for different distances between Agile-Link’s receiver and transmitter where the transmit power complies with FCC part15. Fig. 3-10 shows the effective SNR at the receiver side versus the

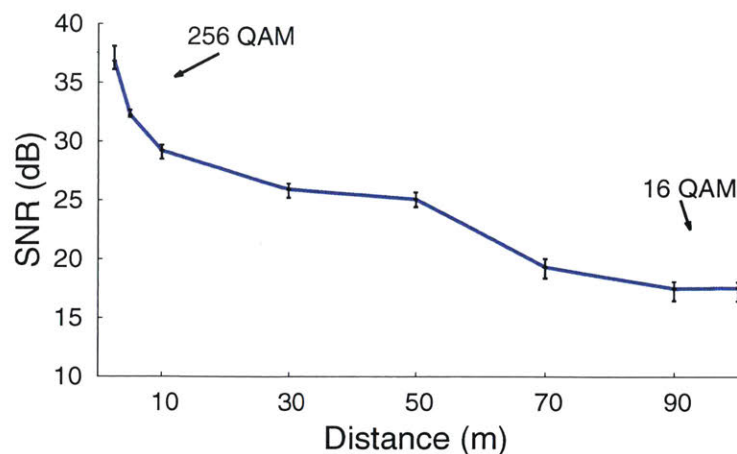


Figure 3-10: **Agile-Link Coverage** Effective SNR at the receiver versus distance between receiver and transmitter.

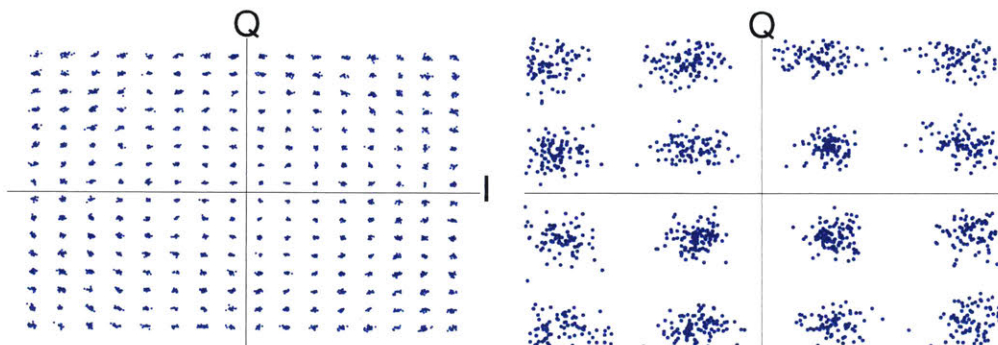


Figure 3-11: **256 QAM and 16 QAM Constellation.** Figure shows that Agile-Link's implementation provides a full OFDM PHY capable of delivering up to 256 QAM and 16QAM for short and long distances, respectively.

distance between transmitter and receiver ranging from 2.5 m to 100 meters. The figure shows that Agile-Link's implementation provides SNR of more than 30 dB for distances smaller than 10 m and 17 dB even at 100 meters which is sufficient for relatively dense modulations such as 16 QAM.³

Fig. 3-11 zooms in on the OFDM modulations at two different distances. The figure shows the constellation for 256 QAM and 16 QAM signals received at 2.5 and 100 meters, respectively. This provides visual evidence that the receiver can accurately decode the received signal, even for very dense constellations like 256 QAM and hence can deliver very high data rates.

³Note that, while one would expect higher SNR at closer distances, the increase in SNR is limited by the dynamic range of the USRP.

3.5 Experimental Evaluation

We evaluate Agile-Link’s ability to identify the best beam alignment quickly and accurately. We ran experiments in a lab area with standard furniture (desks, chairs, computers, etc.). We also ran experiments in an anechoic chamber, where we can accurately measure the ground truth. The anechoic chamber walls are covered with RF absorbers to eliminate multipath and isolate the space from exterior interference. This isolation is necessary to measure the ground truth path traveled by the signal without having RF reflections.

3.5.1 Compared Schemes

We compare the following three schemes:

- **Exhaustive Search:** In this approach, for each setting of the transmitter’s beam direction, the receiver scans all different directions. The combination of transmitter and receiver beams that delivered the maximum power is picked as the direction of the signal.
- **802.11ad Standard:** The standard has three phases[116]. The first stage is called *Sector Level Sweep (SLS)*. In this stage, the AP transmits in all possible directions, and the client sets its receiver beam pattern to a quasi-omnidirectional beam. The process is then repeated with the AP setting its receiver antenna to quasi-omnidirectional and the client sweeping through all transmit directions. At the end of this stage, the AP and client each pick the γ directions that deliver the largest power. The second stage is called *Multiple sector ID Detection (MID)*. This stage repeats the process above but with the transmit beam set to quasi-omni-directional and the scan being performed with the receive beam. This stage compensates for imperfections in the quasi omnidirectional beams. The third stage is called *Beam Combining (BC)*. In this stage, each of the γ best directions at the AP are tried with each of the γ directions at the client. Hence, γ^2 combinations are tested and the combination of transmit

and receive beam directions that deliver the maximum power is then selected and used for beamforming during the data transmission. In our experiments, we set $\gamma = 4$.

- **Agile-Link:** We run the algorithm described in §3.3.2, where the total number of measurements is set to $K^2 \log N$, where N is the number of possible signal directions. We set K to 4 since past measurement studies show that in mmWave frequencies the channel has only 2 to 3 paths [51, 155, 168, 170].

3.5.2 Evaluation Metrics

We evaluate the performance of Agile-Link’s beam searching algorithm along two axes. The first is the accuracy in detecting the best alignment of the receiver’s and transmitter’s beams. In this case, our metric is the SNR loss in comparison to the optimal alignment, *i.e.*, how much SNR could we have gained had we known the ground truth. We calculate this metric by measuring the SNR achieved by our beam alignment and subtract it from the SNR achieved by the optimal alignment.

$$SNR_{loss} = SNR_{optimal} - SNR_{Agile-Link} \quad (3.7)$$

The lower the SNR loss, the higher is our accuracy in detecting the direction of the signal. In order to measure the optimal SNR, we ran experiments in an anechoic chamber where there is no multipath and we can accurately measure the ground truth direction of the signal and align the beams along those directions. However, we also ran experiments in multi-path rich environments. In this case, since we do not know the ground truth, we compute the SNR loss metric relative to the exhaustive search baseline described above.

$$SNR_{loss} = SNR_{Exhaustive} - SNR_{Agile-Link} \quad (3.8)$$

The second metric is the latency in identifying the correct beam alignment in comparison to exhaustive search and the 802.11ad standard.

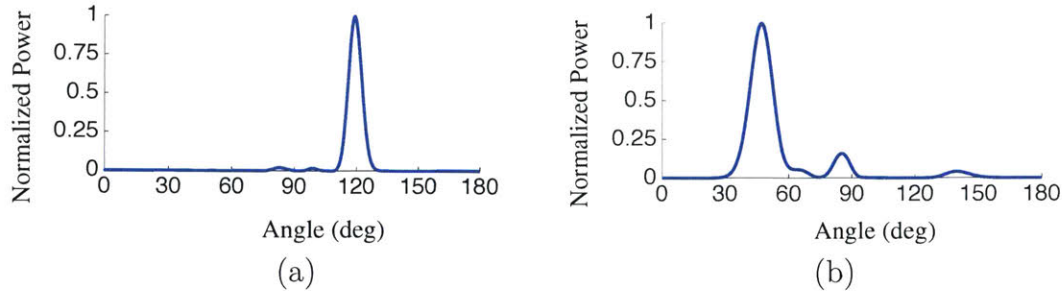


Figure 3-12: **Recovered Directions** The figure shows (a) the direction of arrival that Agile-Link recovers at the receiver and (b) the direction of departure that it recovers at the transmitter when the transmitter is at a direction of 120° relative to the receiver’s array, and the receiver is at a direction of 50° relative to the transmitter.

3.5.3 Beam Alignment Accuracy vs. the Ground Truth

As described above, we first run the experiments in an anechoic chamber, where we can accurately measure the ground truth. For each experiment, we place Agile-Link’s transmitter and receiver at two different locations. We then change the orientation of the transmitter’s and receiver’s antenna arrays with respect to each other. Since there is only a single line-of-sight path in the anechoic chamber, this path will appear at a different direction at the transmitter and at the receiver depending on the orientation of the antenna arrays. Hence, this allows us to test any combination of directions from which the strongest path can leave the transmitter and arrive at the receiver. For each setting, the transmitter transmits measurement frames (as required in 802.11ad) which the receiver uses to compute the directions of the best beam alignment. We then steer the beams based on the output of the alignment and measure the SNR achieved by this alignment.

We first show the best alignment detected by Agile-Link for a particular scenario, where we set the transmitter to be at an angle of 120° with respect to the receiver’s array, and the receiver at an angle of 50° with respect to the transmitter’s array. Fig. 3-12 shows the signal direction recovered by Agile-Link on the transmitter’s and receiver’s sides for this scenario. As can be seen in the figure, Agile-Link has accurately identified the direction of arrival of the signal at the receiver as 120° , and the direction of departure of the signal from the transmitter as 50° .

Next, we compare Agile-Link’s algorithm with the exhaustive search and the

802.11ad standard. To do so, we repeat the above experiment by changing the orientation of the transmitter's and receiver's antenna arrays with respect to each other. We try all possible combination of the antennas' orientations, by rotating each antenna array with respect to the other for all angles between 50° and 130° with increments of 10° . Fig. 3-13 plots a CDF of the SNR loss for Agile-Link's beam searching scheme, the exhaustive search and the 802.11ad standard, in comparison to the optimal alignment. The figure reveals two interesting points:

- The figure shows that Agile-Link performs better than the two baselines in that it has minimal SNR loss. While all schemes have a median SNR loss below 1dB, the 90th percentile SNR loss for both exhaustive search and the standard is 3.95dB which is higher than the 1.89dB SNR loss of Agile-Link. This is due to the fact that the standard and exhaustive search choose to steer using the best beam from a discrete set of N beams which they tested. However, the space of beam directions is continuous and the best beam may not exactly align with the discretization chosen by the algorithms. In this case, they will end up picking the closest beam in the discrete set, which may not be the exact optimal one. SNR loss is further exacerbated by the fact that this can happen on both sides *i.e.*, the transmitter and the receiver. In contrast, Agile-Link uses the beams as a continuous weight over the possible choice of directions (Equation 3.4) and picks the direction that maximizes the overall weight, as described in §3.3.2. Thus, Agile-Link can discover the direction of the path beyond the N directions used by exhaustive search and the standard.
- The figure also shows that the standard and exhaustive search have similar performance. This might seem surprising since one may expect exhaustive search to find a better beam alignment since it spends more time searching the space. However, it is important to recall that the standard differs from the exhaustive search only in the first stage where it uses a quasi-omnidirectional beams to limit the search space to a few top candidates. In the final the stage, however, the standard tries all possible combinations of these candidate beams. Since

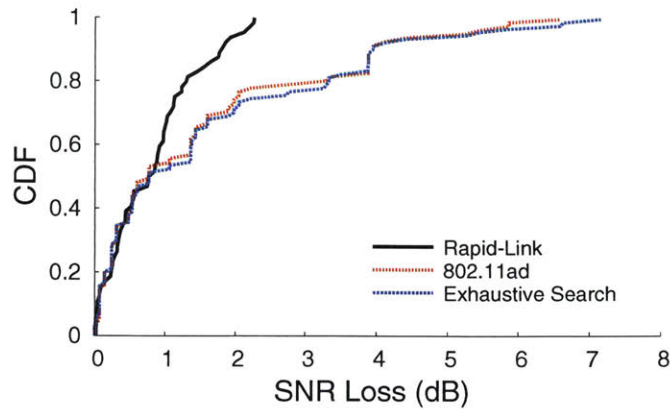


Figure 3-13: **Beam Accuracy with a Single Path** The figure shows the SNR loss due to beam misalignment for Agile-Link, the 802.11ad standard, and exhaustive search.

there is only one path in this experiment, as long as the best beam is picked as one of the candidate beams in the first stage, the standard will converge to the same beam alignment as the exhaustive search. However, we will show next that this does not continue to hold in multipath settings.

3.5.4 Alignment Accuracy in Multipath Environments

We repeat the above experiments in an office environment, where due to multipath, the signal can arrive from different directions. However, as described earlier, in this case, we do not have the ground truth for the direction of strongest path and hence we measure the SNR loss relative to the exhaustive search baseline. Note that since exhaustive search exhaustively tries all possible combinations of directions, it is not sensitive to multipath and maintains its performance.

Fig. 3-14 plots a CDF of SNR loss for Agile-Link and the 802.11ad standard with respect to the exhaustive search. The figure shows that the standard performs much worse in multipath scenarios. Specifically, instead of having a similar SNR to the exhaustive search as before, the median and 90th percentile SNR loss (with respect to exhaustive search) are 4dB and 12.5dB, respectively. This is because the standard is using its phased array as a quasi-omnidirectional antenna and hence the multiple paths can combine destructively, in which case the information is lost. Further, due to imperfections in the quasi-omnidirectional patterns, some paths can get attenuated

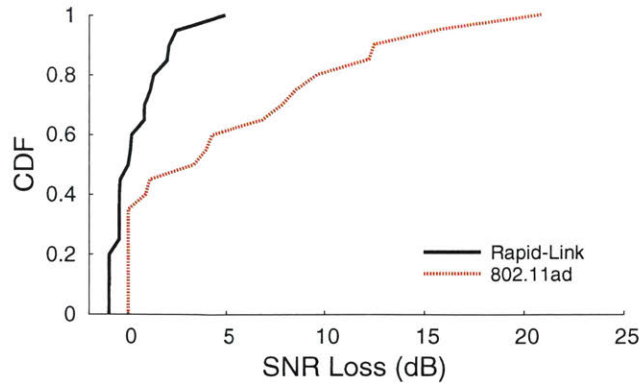


Figure 3-14: **Beam Accuracy with Multipath** The figure shows the SNR loss of the 802.11ad standard and Agile-Link with respect to the exhaustive search.

and hence the standard can easily choose the wrong direction to align its beam. In contrast, Agile-Link performs well even in the presence of multipath. Specifically, the median and 90th percentile SNR loss with respect to exhaustive search are 0.1dB and 2.4dB, respectively. Finally, the figure also shows that sometimes Agile-Link’s SNR loss with respect to exhaustive search is negative. This is because in some cases, Agile-Link performs better than exhaustive search for the same reasons described above.

3.5.5 Beam Alignment Latency

Next we would like to evaluate the gain in reducing latency that Agile-Link delivers over the two baselines. However, since our radio has a fixed array size we cannot empirically measure how this gain scales for larger arrays. Hence, we perform extensive simulations to compute this gain for larger arrays and we use the empirical results from our 8-antenna array to find the delay for this array size.

(a) *Reduction in the Number of Measurements:* Since each measurement in 802.11ad requires sending a special frame, one way to measure delay is in terms of the number of measurements frames. Fig. 3-15 plots the reduction in the number of measurements that Agile-Link achieves over exhaustive search and the standard. The figure shows that, for an 8-antenna phased array, Agile-Link can reduce the number of measurements by $7\times$ and $1.5\times$ compared to exhaustive search and standard respectively.

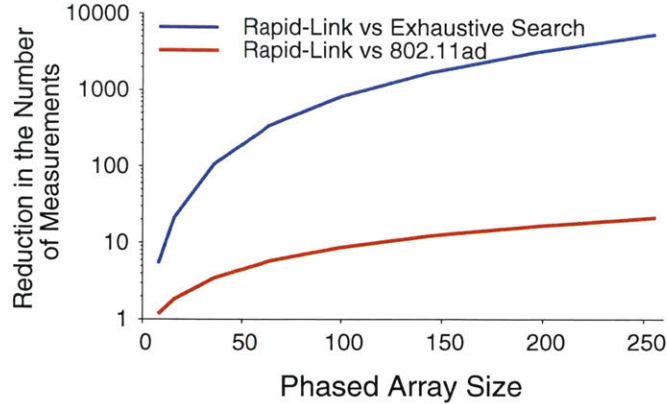


Figure 3-15: **Beam Alignment Latency** The figure plots the reduction in the number of measurements for Agile-Link compared to the 802.11ad standard and exhaustive search.

Further, the gain increases quickly as the number of antennas increase. This is due to the scaling property of each algorithm and whether it is quadratic, linear, or logarithmic. The figure shows that the gain of Agile-Link over exhaustive search and the standard increases very fast and for arrays of size 256 is $16.4\times$ better than the standard and three orders of magnitude better than exhaustive search.

(b) *Reduction in Search Time:* Next, we look at the amount of time it takes to find the best alignment in each scheme under the 802.11ad MAC protocol. The standard is still evolving; our description is based on [116]. Since the delays in exhaustive search are unacceptable in practice, we consider only Agile-Link and the beam alignment scheme in the standard.

The 802.11ad has a protocol for when the AP and clients search the space to align their beams [116, 144]. The delay produced through this process differs from simply multiplying the number of measurement frames by the duration of each measurement. This is due to three main reasons: 1) The protocol allows for beam scan (called beam training) only during certain intervals. If the client cannot collect all necessary measurements, it needs to wait until the next opportunity to perform more measurements. 2) Different clients contend for the beam alignment slots; hence, the delay will increase depending on the number of clients. 3) When the AP sweeps its beam, all clients can collect measurements; hence this part can be amortized.

To better understand the above constraints, let us describe at a high-level how

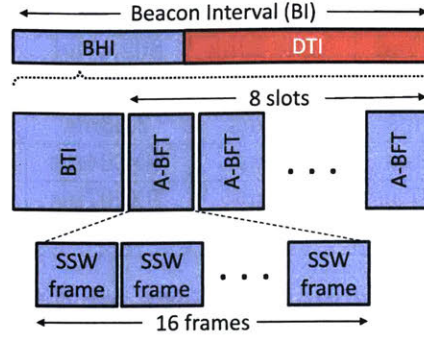


Figure 3-16: 802.11ad Beacon Interval Structure

802.11ad performs beam-forming training. The AP periodically transmits beacon intervals (BI), which typically last for 100 ms [144]. Each BI has a beacon header interval (BHI), followed by a data transmission interval (DTI), as shown in Fig. 3-16. The search for the best alignment is done during the BHI. Each BHI consists of one beacon transmission interval (BTI) which is used by the AP to train its antenna beam, and eight association beam-forming training (A-BFT) slots, which are randomly selected by clients to train their beams. Finally, each A-BFT slot consists of up to 16 SSW frames, where each frame is used to perform one measurement and has a duration of $15.8\mu s$ [36, 116]. Each BI has a maximum of 8 beam training slots. All clients contend for training in those slots. If the client cannot finish its training during one A-BFT, it can contend for further slots during the same or following BI. Yet, waiting for the next BI increases the delay by 100ms.

As explained in §3.5.1, 802.11 performs beam refinement where each of γ best directions at the AP and client are compared again. To simplify the computation, we conservatively ignore the 802.11ad beam refinement since it only increases the delay of 802.11ad, and improves the relative gains of Agile-Link. Also when simulating 4 clients, we assume that the contention succeeded without collision. This too is a conservative assumption since Agile-Link requires significantly fewer measurement slots and hence, given the same number of slots, the collision probability between clients is smaller in Agile-Link. Finally, the AP trains its beam during the BTI, and uses frames similar to those used for the client beam training. The AP does not need to repeat this training per client.

Size	One Client		Four Clients	
	802.11ad	Agile-Link	802.11ad	Agile-Link
8	0.51ms	0.44ms	1.27ms	1.20ms
16	1.01ms	0.51ms	2.53ms	1.26ms
64	4.04ms	0.89ms	304.04ms	2.40ms
128	106.07ms	0.95ms	706.07ms	2.46ms
256	310.11ms	1.01ms	1510.11ms	2.53ms

Table 3.1: **Beam alignment latency for different array size**

Table 3.1 shows the beam alignment delay for different antenna-array sizes, both for the case of one client and 4 clients. The table shows that as the number of antennas in the array increases, the delay in 802.11ad increases quickly. In contrast, Agile-Link can operate within the same standard, but it extracts more information from each measurement, hence keeping the the delay low even for large antenna arrays. In particular, for antenna arrays of 256 elements, the proposed 802.11 beam alignment algorithm takes hundreds of milliseconds for one user and over 1.5 seconds for 4 users. In contrast, Agile-Link keeps the delay below 1.01ms and 2.53ms, respectively.

3.6 Discussion

This chapter presents Agile-Link, a phased array mmWave system that can find the best beam alignment without scanning the entire space. Agile-Link delivers a mmWave beam alignment algorithm with provably logarithmic measurements for the phased-array architecture commonly used in mmWave access points and clients. Our algorithm finds the correct alignment of the beams between a transmitter and a receiver orders of magnitude faster than existing radios. In particular, it reduces the alignment delay from over a second to 2.5ms. We note that our results are based on a phased array which has linear number of phase shifters and only one transmit-receive chain. One could reduce the beam alignment latency further by using a more complex hardware architecture or designing inference algorithms that synthesize the inherent WiFi or GPS information with RF-localization in order to quickly steer the mmWave radios' beams. We plan to explore such desings in future work.

Chapter 4

High-Quality Untethered Virtual Reality

The past few years have witnessed major advances in augmented reality and virtual reality (VR) systems, which have led to accelerated market growth. Facebook has recently started shipping their VR headset (Oculus Rift) and expects to ship more than 2 million headsets by 2017 [7]. HTC sold more than 15,000 VR headsets in the first 10 minutes following their release [9]. These devices are expected to soon dominate the gaming and entertainment industry, and they have found applications in manufacturing and healthcare [29, 30]. However, a key challenge prevents this technology from achieving its full potential. High-quality VR systems need to stream multiple Gbps of data from their data source (PC or game console) to the headset. As a result, these headsets have an HDMI cable snaking down the player's neck and hardwiring her to the PC, as shown in Fig 4-1. The cable not only limits the player's mobility and interferes with the VR experience, but also creates a tripping hazard since the headset covers the player's eyes. This has left the industry searching for untethered solutions that can deliver a high-quality VR experience without these limitations. Unfortunately, typical wireless systems, such as WiFi, cannot support the required data rates. This challenge has led to awkward products: Zotac has gone as far as stuffing a full PC in the player's backpack in the hope of delivering an untethered VR.

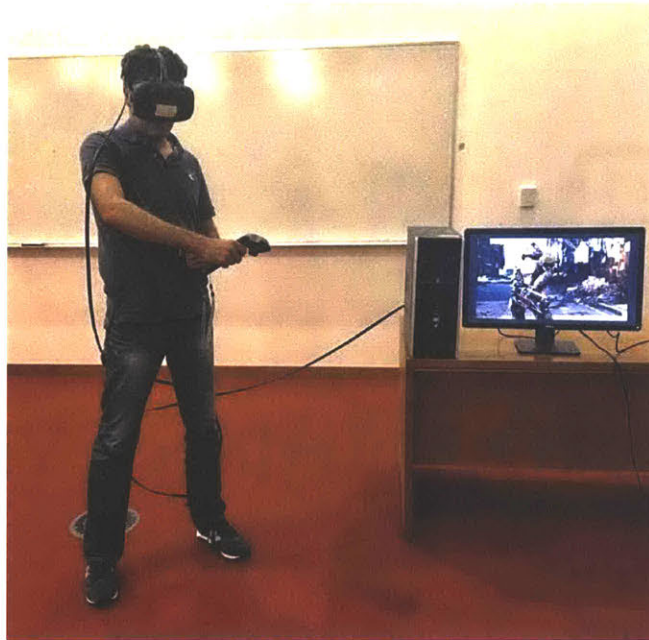


Figure 4-1: **Virtual Reality Experience:** The figure shows the headset’s cable not only limits the player’s mobility but also creates a tripping hazard.

Ideally, one would like to replace the HDMI cable with a wireless link. Thus, multiple companies have advocated the use of mmWave for VR since mmWave radios have been specifically designed to deliver multi-Gbps data rates [18, 164]. The term mmWave refers to high frequency RF signals in the range of 24 GHz and higher [8, 16]. The 802.11ad standard operates in mmWave and can transmit over 2 GHz of bandwidth and deliver up to 6.8 Gbps. No other consumer RF technology can deliver such data rates. However, mmWave links bring up new challenges that must be addressed before this technology can be used for VR applications:

- **Dealing with blockage:** mmWave links require a line-of-sight between transmitter and receiver, and they do not work well through obstacles or reflections. This problem is due to the fact that mmWave antennas are highly directional and typically generate narrow beams. Hence, even a small obstacle like the player’s hand can block the signal. Said differently, these links work well when the receiver on the headset has a clear line-of-sight to the transmitter connected to the PC, but if the player moves her hand in front of the headset (see Fig. 4-3), or other people in the environment obstruct the receiver’s view to the trans-

mitter, the signal will be temporarily lost, causing a glitch in the data stream (shown in our empirical results in §4.3). While temporary outages are common in wireless communication, VR data is non-elastic, and cannot tolerate any degradation in SNR and data rate.

- **Dealing with mobility:** Since mmWave radios use highly directional antennas, they work only when the transmitter’s beam is aligned with the receiver’s beam. Further, since the wavelength is very small, even a small movement of the headset can hamper the alignment and break the link. Past work on mmWave typically assumes static links and, hence, fixed alignment [77, 106, 202]. Identifying the correct alignment for the antennas can take up to multiple seconds [168, 203]. Such delay is unacceptable for VR systems, which need to play a new frame every 10 milliseconds, even when the headset moves [11].

This chapter introduces MoVR, a wireless system that enables a reliable and high-quality untethered VR experience via mmWave links. MoVR addresses the main challenges facing existing mmWave links. In particular, MoVR overcomes the blockage problem by introducing a self-configurable mmWave mirror that detects the incoming signal and reconfigures itself to reflect it toward the receiver on the headset. In contrast to a traditional mirror, a MoVR mirror does not require the angle of reflection to be equal to the angle of incidence. Both angles can be programmed so that our mirror can receive the signal from the mmWave transmitter attached to the data source and reflect it towards the player’s headset, regardless of its direction. In §4.3.2, we explain the design of such mmWave mirrors and how they can be implemented simply by deflecting the analog signal without any decoding.

Next, MoVR ensures that the VR system sustains high data rates to the headset in the presence of mobility. In contrast to past work on mmWave [35, 164, 203], MoVR does not scan the space to find the best way to align the mmWave directional antennas, a process known to incur significant delay [168, 203]. Specifically, MoVR finds the best beam alignment by relying on existing tracking functions available in VR systems. In designing MoVR, we observe that VR systems already track the

location of the headset to update the 3D view of the player. Thus, MoVR leverages this information to quickly localize the headset and move the transmitter antenna's beam with it. However, while the VR application tracks the movements of the headset, it does not know the location of the headset with respect to the mmWave transmitter and the MoVR mirror. Thus, we design a novel algorithm that combines the output of VR headset tracking with RF-based localization to quickly steer the mmWave antennas and keep their beams aligned as the player moves around.

We have built a prototype of MoVR and evaluated its performance empirically using an HTC VR system. Our results can be summarized as follows:

- In the absence of MoVR's mirror, even a small obstacle like the player's hand can block the mmWave signal and result in a drop in SNR of 20dB, leaving the VR headset with no connectivity. The addition of MoVR's mirror prevents the loss of SNR in the presence of blockage, sustaining high data rates.
- Given the VR headset information, MoVR aligns the antenna beams in under a few microseconds, which is negligible compared to the user's movement. Further, the resulting alignment sustains the required high SNR and VR data rates.
- Finally, in a representative VR gaming setup, MoVR provides an SNR of 24dB or more for all locations in the room and all orientations of the headset, even in the presence of blockage and player mobility. This SNR is much higher than the 20dB needed for the VR application.

4.1 Related Work

Related work can be classified into three areas.

(a) Virtual Reality: Existing VR systems can be divided into PC-based VR, like Oculus Rift and HTC Vive, and Gear VR, like systems by Samsung and Visus [22, 31]. PC-based VR systems leverage their computational horsepower to generate rich graphics that look realistic and support fast head motion. They require, however,

an HDMI cable to connect the PC to the headset. Gear VR slides a powerful smart phone into the headset. Thus, they do not need an external cable. Their mobility, however, is limited by the inability to support rich graphics that react to motion; their imagery tends to blur with motion [2]. There is a huge interest in untethered PC-based VR systems. Optoma and SiBeam have proposed using mmWave radios to connect the headset to the PC, but they have not provided any details about their proposal [18, 164]. Sulon proposed to equip the headset with an integrated computer [25]. Unfortunately, this would make the headset much larger and heavier, thus interfering with the user experience. WorldViz advertises a wireless wide-area tracking system. However, they still require the user to have a cable for the display or carry a limited data source and a processor unit [34]. Zotac advertises a mobile VR headsets where the user carries the PC in a backpack. Finally, Google has recently announced that their next VR headset will be wireless, but has not provided any details of the design or the release date [32].

(b) mmWave Communications: Much past work on mmWave communication addresses *static links*, such as those inside a data center [77, 106, 202], where there is a line-of-sight path between the transmitter and receiver. Some past work looks at mobile links for cellular networks or wireless LANs [156, 170]. These systems typically scan the space to align the antennas, a process that takes up to several seconds [168, 203]. In contrast, by leveraging the fact that VR systems already track the headset, MoVR is able to speed up antenna steering enough that it can be done faster than the VR frame rate, as we show in §4.6. Also, most past work on mmWave links assumes line-of-sight connectivity. Some past works do consider scenarios in which the line-of-sight between transmitter and receiver is blocked [105, 169, 170]. However, since they target elastic applications, their solution switches the directional antenna to the best reflected path, which typically has a much lower SNR (see Fig. 4-4). In contrast, the VR application is non-elastic and cannot tolerate reduction in its SNR and data rate. Also, there are wireless HDMI (WHDI) products from LG and Samsung which operate at mmWave frequencies, but these products assume static links and require line-of-sight between the receiver and transmitter [33]. Thus, they cannot adapt their

direction and will be disconnected if the player moves. Finally, the work in [202] has proposed a form of mmWave mirror to reflect an RF signal off the ceiling of a data center. Their approach, however, covers the ceiling with metal. Such a design is unsuitable for home applications and cannot deal with player mobility.

(c) Relay and Full-Duplex: The design of MoVR mirrors is related to that of wireless relays at lower frequencies (e.g., Wi-Fi and LTE [28]). Similarly to a MoVR mirror, these relays amplify and forward the signal of interest. However, they do not deal with the issue of directionality. In contrast, our MoVR mirror needs to capture the mmWave signal along a particular direction and reflect it in the direction of the headset. Finally, MoVR mirrors are related to previous work on full-duplex relays since they receive a signal and transmit it at the same time. However, full-duplex radios require complex analog and digital hardware with full transmit and receive chains [59]. In contrast, MoVR mirrors have only an analog front-end (i.e., antennas and an amplifier) and do not need digital transmit or receive chains.

4.2 MoVR Overview

MoVR is a wireless communication system for virtual reality applications. It enables a sustainable, high-data-rate wireless link even in the presence of blockage and headset mobility. High-quality VR systems stream multiple Gbps from a high-power PC to the headset. MoVR delivers this data over a mmWave wireless link. Fig. 4-2 shows MoVR's setup. The PC is connected to a mmWave transmitter, which we refer to as the AP, and the headset is equipped with a mmWave receiver. As the figure shows, MoVR operates in two modes, depending on the real-time scenario: when the direct path from the AP to the headset is clear, the AP beams its signal to the headset. However, if the direct path is blocked, the AP detects the blocking and reflects its signal to the headset via a MoVR mirror. The environment may have one or more MoVR mirrors; the AP picks the best one depending on the headset location.

The next few sections present the components that contribute to the design of MoVR. We start by explaining the two key challenges in using mmWave links in VR

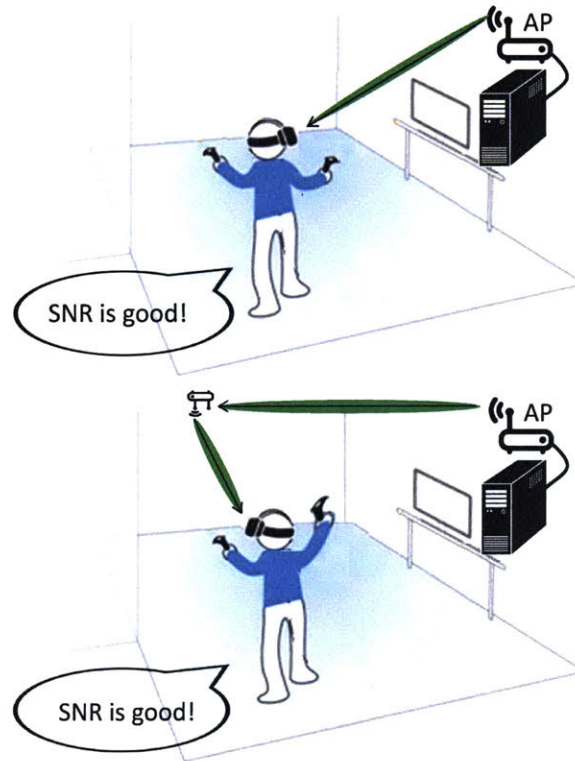


Figure 4-2: **MoVR's Setup:** The figure shows MoVR's setup. The PC is connected to a mmWave AP and the headset is equipped with a mmWave receiver. In the case of a blockage (e.g., the user raises her hand or turns her head), the AP delivers its signal by reflecting it off a MoVR mirror.

systems, and how we overcome them. We then explain how the various components work together to satisfy the application.

4.3 Blockage Problem

A key challenge in using mmWave links for VR applications is that they may be easily blocked by a small obstacle, such as the player's hand. This is a side effect of highly directional antennas, which mmWave radios must use to focus their power and compensate for path loss. Below, we investigate the impact of blockage in more detail, then explain our solution to overcome this problem.

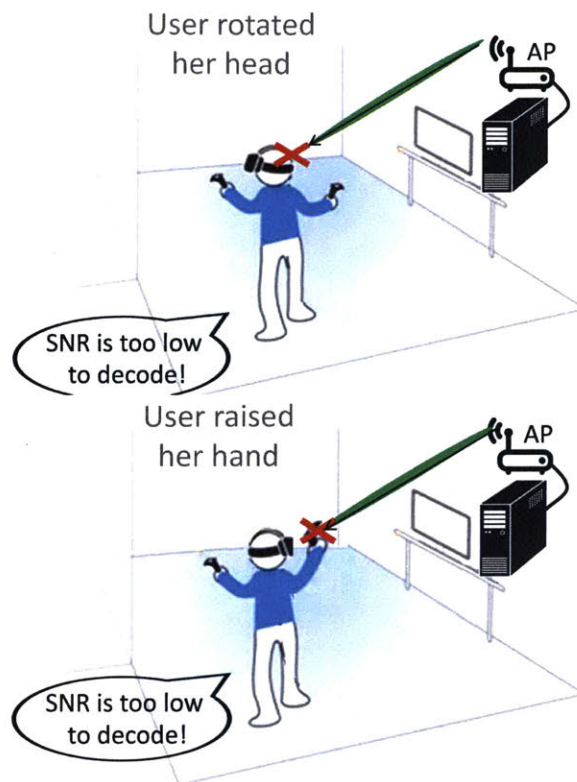


Figure 4-3: **Blockage Scenarios:** As the user moves his head or hand, the line-of-sight path between the AP and the headset’s receiver can be easily blocked. This results in a significant drop in SNR and data rate.

4.3.1 Impact of Blockage

We first investigate the impact of blocking the direct line-of-sight on the signal’s SNR and the link’s data rate. To do so, we attach a mmWave radio to an HTC PC-based VR system and another one to the headset (see §4.6 for hardware details). We conduct experiments in a $5m \times 5m$ room. We place the headset in a random location that has a line-of-sight to the transmitter, and measure the SNR at the headset receiver. We then block the line-of-sight and measure the SNR again. We consider different blocking scenarios: blocking with the player’s hand, blocking with the player’s head, and blocking by having another person walk between headset and the transmitter. We repeat these measurements for multiple different locations. Fig. 4-4 shows the results of this experiment: the top graph shows the SNR and the bottom graph shows the data rate. The SNRs are measured empirically and the corresponding data rates are computed by substituting the SNR measurements into standard rate tables based on

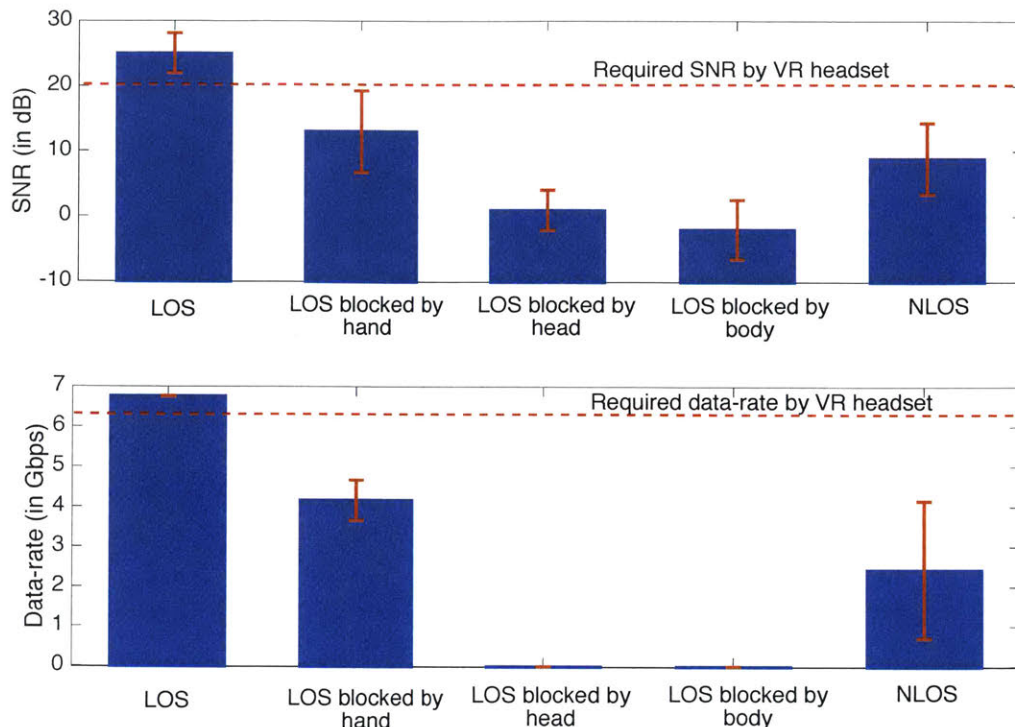


Figure 4-4: **Blockage Impact on Data Rate.** The Figure shows SNR and data rate for different scenarios: line-of-sight (LOS) without any blockage, LOS with different blockages and non-line-of-sight (NLOS). The figure shows that blocking the signal with one’s hand, head, or body results in a significant drop in SNR and causes the system to fail to support the required VR data rate. The figure also shows that simply relying on NLOS reflections in the environment does not deliver good SNR and would fail to support the required data rate.

the 802.11ad modulation and code rates [10, 13, 14]. The first bar in Fig. 4-4 shows that, in the absence of blocking, the mean SNR is 25dB and the resulting data rate is almost 7 Gbps, which exceeds the needs of the VR application. Bars 2, 3, and 4 in the figure correspond to different blocking scenarios. They show that even blocking the signal with one’s hand degrades the SNR by more than 14 dB and causes the data rate to fail to support the VR application.

One solution to overcome this challenge is to rely on non-line-of-sight paths –i.e., the signal reflections from walls or other objects in the environment. For example, both the transmitter and the headset receiver can direct their signal beams toward a wall and rely on the natural reflection from the wall. In fact, this is how current mmWave systems work. Unfortunately, non-line-of-sight paths typically have much

higher attenuation than the line-of-sight path due to the fact that walls are not perfect reflectors and therefore scatter and attenuate the signal significantly. Moreover, signals travel a longer distance in non-line-of sight scenarios than in line-of sight scenarios, which results in higher attenuation.

To confirm, we repeat the measurements for all blocking scenarios, but instead of trying to receive the signal along the blocked direct path, we sweep the mmWave beam on the transmitter and receiver in all directions. We try every combination of beam angle for both transmitter and receiver antennas, with 1 degree increments. We ignore the direction of the line-of-sight and note the maximum SNR across all non-line-of-sight paths. The last bar in Fig. 4-4 shows the results of this experiment. It shows that when the transmitter and receiver have to use a non-line-of-sight path, the SNR drops by 16dB on average. The figure also shows that this reduction in SNR causes the data rate to fail to support the VR application.

Note that one cannot solve the blockage problem by putting more antennas on the back or side of the headset, since the line-of-sight from the AP to the headset may get completely blocked by the player's hands or body (as shown in Fig. 4-3), or by the furniture and other people in the environment. One naïve solution to overcome this challenge is to deploy multiple mmWave APs in the room to guarantee that there is always a line of sight between the transmitter and the headset receiver. Such a solution requires extending many HDMI cables in the environment to connect each AP to the PC. However, this defeats the purpose of a wireless design because it requires enormous cabling complexity. Further, requiring multiple full-fledged mmWave transceivers will significantly increase the cost of VR systems and limit their adoption in the market.

In the next section, we describe how to overcome the blockage problem using mmWave mirrors. Our mirrors do not need to connect to the PC, and have a simple cheap design with no digital transmit or receive components.

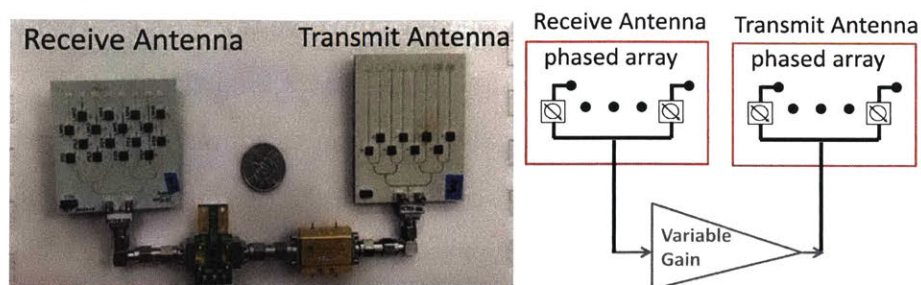


Figure 4-5: **MoVR Programmable mmWave Mirror:** The figure shows (a) the implementation and (b) the block diagram of the mirror. The design of the mirror is small and simple. It consists of two directional phased-array antennas connected via a variable-gain amplifier.

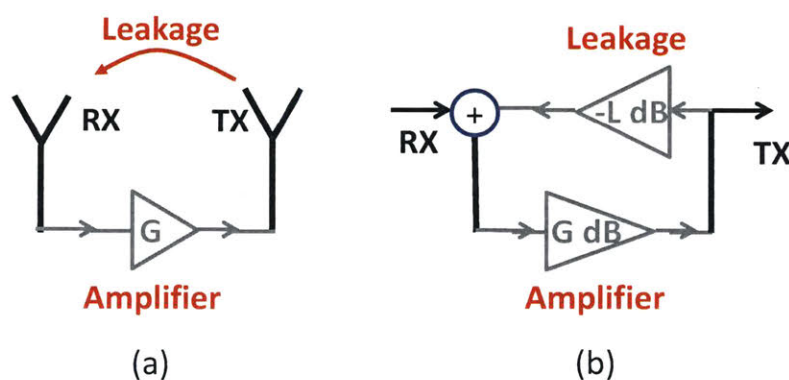


Figure 4-6: **MoVR's Mirror Block Diagram:** The figure shows (a) the block diagram and (b) equivalent signal-flowgraph of MoVR's mirror. The figure shows that the input signal is first amplified by G_{dB} , then attenuated by L_{dB} and fed back to the input as a leakage.

4.3.2 Programmable mmWave Mirrors

To overcome the blockage problem, we designed a programmable mmWave mirror that can control both the angles of incidence and reflection [39]. Fig. 4-5 shows a basic diagram of the circuit and a picture of our prototype. Each MoVR mirror consists of a transmit and receive antenna connected via a variable-gain amplifier. As is common in mmWave radios, the antennas are implemented using phased arrays in order to create highly-directional beams, which can be steered electronically in a few microseconds. Note that the design is quite simple. Specifically, it neither decodes the signal nor includes any transmit or receive digital components (DAC, ADC, mixer, etc.). This allows us to avoid complex and expensive components that would have to operate at multiple Gbps.

An important challenge in designing such a mmWave mirror stems from the leakage between the transmit and receive antennas. At a high level, a MoVR mirror works by capturing the RF signal on its receive antenna, amplifying it, and reflecting it using a transmit antenna. However, some of the signal reflected by the mirror is also received by its own receive antenna. This means that the output of the amplifier is fed back to the input of the amplifier. This creates a feedback loop that can cause the amplifier to saturate, thereby generating garbage signals. Thus, a key question in designing MoVR mirrors is: how do we set the optimal amplifier gain so that we avoid saturation, but also maximize the SNR delivered to the headset?

In order to ensure that the leaked signal is damped while the signal of interest (i.e., the received signal from the AP) is amplified, we need to ensure that the amplifier gain is less than the leakage. To see why this is the case, consider the signal-flow-graph of the mirror, shown in Fig. 4-6(b). The input signal is first amplified by G_{dB} , then attenuated by L_{dB} and fed back to the input. From Control Theory [81], we know that for this system to stay stable we need to ensure that $G_{dB} - L_{dB} < 0$ [65, 81]. This implies that the amplifier gain (G_{dB}) must be lower than the absolute value of the leakage (L_{dB}); otherwise the system becomes unstable, leading to saturation of the amplifier.

To avoid this saturation, the mirror needs to measure the leakage and then set the amplification gain lower than the leakage. The leakage, however, varies when the direction of the antenna beam changes to track the headset. Fig. 4-7 shows the leakage across different transmit beam directions for two different receive beam directions. As we can see, the leakage variation can be as high as 20dB. The variation of the leakage and the fact that the amplifier gain must always be set lower than the leakage create a need for an adaptive algorithm that reacts to the leakage in real time and adjusts the amplifier gain accordingly.

One naïve algorithm is to send a signal from the mirror's transmit antenna and measure the received power at its receive antenna in order to estimate the amount of leakage, then to use this information to set the amplifier gain accordingly. However, we cannot do this since a MoVR mirror does not have digital transmit and receive

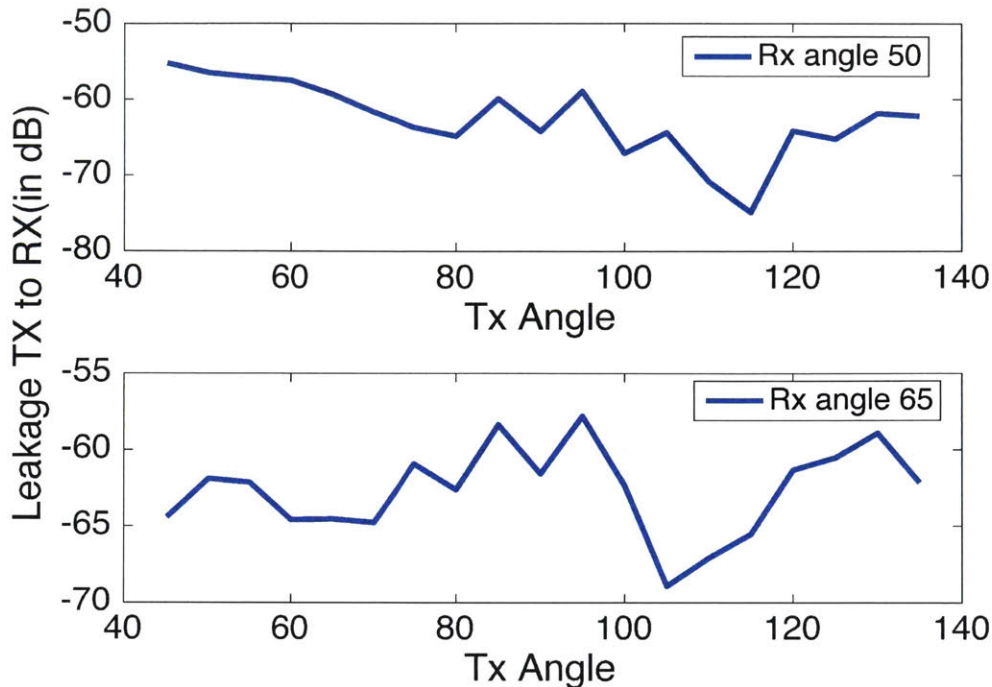


Figure 4-7: **Leakage Between Mirror’s Transmit and Receive Antennas:** The figure shows the leakage across different transmit beam directions for two different receive beam directions. The figure shows that the leakage variation can be as high as 20dB. This result confirms a need for an adaptive algorithm that reacts to the leakage in real time and adjusts the amplifier gain accordingly.

chains.

Our solution exploits a key characteristic of amplifiers: an amplifier draws significantly higher current (from a DC power supply) as it gets close to saturation mode, compared to during normal operation [68, 103].¹ We can therefore detect if the amplifier is getting close to its saturation mode by monitoring the current consumption from the power supply. Thus, our gain control algorithm works as follows: it sets the amplifier gain to the minimum. It increases the amplifier gain step by step while monitoring the amplifier’s current consumption. The algorithm continues increasing the gain until the current consumption suddenly goes high. This indicates that the amplifier is entering its saturation mode. The algorithm then backs off, keeping the amplification gain just below this point.

¹The exact quantity of the amplifier’s current consumption for its different operating modes are specified in its datasheet. We use a simple IC which measures the current consumption of the amplifier to detect its operating mode.

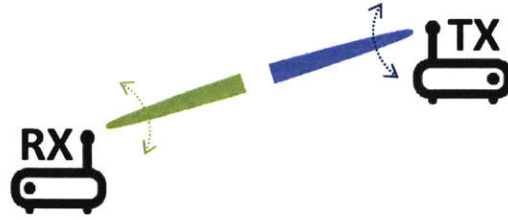


Figure 4-8: **Beam Alignment in mmWave Radios:** mmWave radios need to find the best alignment between the transmitter’s and receiver’s beams to establish a communication link.

4.4 Dealing with Mobility

Movement of the VR headset creates a critical challenge for mmWave links. Specifically, mmWave frequencies suffer from a large path loss. To compensate for this loss, mmWave radios use highly directional antennas to focus the signal power in a narrow beam. Such directional antennas can be implemented using phased arrays. In fact, since the wavelength is very small (on the order of a millimeter), tens or hundreds of such antennas can be packed into a small space, creating a pencil-beam antenna. The beam can be steered electronically in a few microseconds. However, the real challenge is to identify the correct spatial direction that aligns the transmitter’s beam with the receiver’s beam (as shown in Fig. 4-8). This is particularly difficult in VR applications since the headset is naturally in a mobile state.

Below, we investigate the impact of beam misalignment on the signal and explain our solution to overcome this problem.

4.4.1 Impact of Beam Misalignment

We first investigate the impact of beam misalignment on the SNR of the signal delivered to the headset. To do so, we attach a mmWave transmitter to the VR PC (which we call the AP), and a mmWave receiver to the headset. We position the headset’s receiver such that it has a line-of-sight path to the AP’s transmitter. We ensure that the transmitter’s and receiver’s beams are perfectly aligned by scanning for all possible alignments and picking the one that maximizes the SNR. We use this setup as the initial position of the headset in our experiment –i.e., we start the experiment

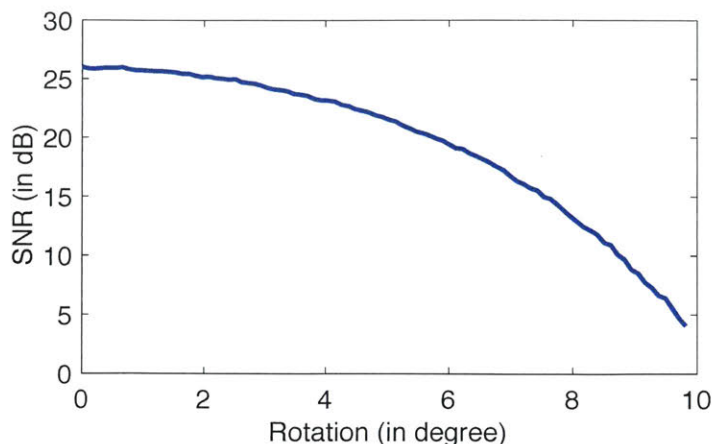


Figure 4-9: **SNR versus Amount of Headset Rotation:** Even a minor head rotation of a few degrees can cause a major degradation in the received SNR. This result confirms the need for real-time beam tracking to realign the transmitter’s and the receiver’s beams as the player moves her head.

with a perfect beam alignment. We then rotate the headset and measure the SNR as a function of the angular deviation from the perfect orientation. Note that the headset rotation causes misalignment between the transmitter’s and receiver’s beams. Fig. 4-9 shows the SNR of the received signal versus the amount of headset rotation. The figure shows that even a minor head rotation of a few degrees can cause a major degradation in the SNR. As was shown in §4.3, such reduction in SNR creates outages for the VR application. This experiment confirms the need for real-time beam tracking to realign the transmitter’s and the receiver’s beams as the player moves her head.

4.4.2 Beam Alignment and Tracking

In this section, we explain how MoVR aligns the transmitter’s and receiver’s beams, and adapts the alignment as the headset moves. Recall that MoVR operates in two different modes (as shown in Fig. 4-2). In the first mode, the AP communicates to the headset directly. This requires a beam alignment between the AP and the headset. In the second mode, the AP communicates to the headset through the mirror. This requires beam alignment between the AP and the mirror, and beam alignment between the mirror and the headset. Therefore, there are three types of

beam alignment which need to be addressed in MoVR. Below, we explain each of them in more detail.

(a) **Beam alignment between the AP and the mirror:** To deliver the signal from the AP to the mirror, the AP needs to align its transmit beam toward the mirror and the mirror needs to align its receive beam toward the AP. Since both the AP and the mirror are static, this alignment is only done once when the mirror is installed. Though this alignment has no real-time constraints, it cannot employ past work on beam alignment [83, 96, 123, 154, 197] because all of these schemes require both nodes to transmit and/or receive signals. A MoVR mirror, however, can neither transmit nor receive; it can only reflect signals.

Thus, MoVR delegates to the AP the task of measuring the best beam angle, which the AP can then communicate to the mirror using a low-bit-rate radio, such as bluetooth. During this estimation process, the AP transmits a signal and the mirror tries to reflect this signal back to the AP itself (instead of reflecting it to the headset) allowing the AP to measure the best angle. The mirror, however, does not yet know the direction of the AP, so it has to try various directions and let the AP figure out the direction that maximizes the SNR.

Thus, our algorithm works as follows. It first sets the mirror's receive and transmit beams to the same direction, say θ_1 , and sets the AP's receive and transmit beams to the same direction, say θ_2 . Then it tries every possible combination of θ_1 and θ_2 while the AP is transmitting a signal and measuring the power of the reflection (from the mirror). The θ_1 and θ_2 combination that gives the highest reflected power corresponds to the angles for the best alignment of the mirror's receive beam and the AP's transmit beam.

One problem remains. As described above, the AP needs to measure the power of the signal reflected by the mirror, while transmitting its own signal. Performing this measurement is not easy. This is due to the fact that the AP is trying to transmit and receive at the same time. As a result, the transmitted signal leaks from the AP's transmit antenna to its receive antenna. So to measure the reflected signal power, the AP first needs to separate it from the strong leakage signal it receives.

To overcome this problem, we use the fact that, if the mirror modulates the signal before it reflects it, the AP can separate the reflected signal from the leakage signal as the two signals become different. For example, if the AP transmits a sinewave at a frequency f_1 , and the mirror modulates this signal by turning its amplifier on and off at a frequency f_2 , then the center frequency of the reflected signal will be $f_1 + f_2$ while the leakage signal remains at f_1 . Hence, the AP can simply use a filter to separate the reflected signal from the leakage signal.

(b) Beam alignment and tracking between the AP and the headset:

To deliver the signal directly from the AP to the headset, the AP needs to align its transmit beam toward the headset and the headset needs to align its receive beam towards the AP. Here, we will explain how MoVR estimates the angles for best alignment of the AP's transmit beam and headset's receive beam.

We observe that VR systems have to track the location and orientation of the headset in order to update the 3D view of the player. Specifically, the HTC VR system does this using laser trackers and an IMU on the headset. Using this infrastructure, the VR system is able to calculate the headset's exact position relative to each laser tracker. By co-locating MoVR's AP with one of the VR's laser trackers, we can exploit the VR tracking system to find the exact location and orientation of the headset relative to the AP.² MoVR leverages this information to calculate the best alignment to the headset and track the alignment in real time.

(c) Beam alignment and tracking between the mirror and the headset:

In order to align the mirror's transmit beam with the headset's receive beam, we need to know the location and orientation of the headset with respect to the mirror. Unfortunately, the VR tracking system only has this information with respect to the AP. In order to switch the reference point, we need to get the location and orientation of the mirror with respect to the AP. Getting the orientation was explained in §4.4.2(a), but we still need to get the distance between the AP and the mirror.

Because the location of the mirror with respect to the AP is fixed during use of

²In practice, location of the AP may be a few cm different from the location of the laser tracker. Since this is a fixed deviation, it can be calibrated by the manufacturer. Also note that even if the AP is blocked from the headset, the VR system has enough redundancy to localize the headset.

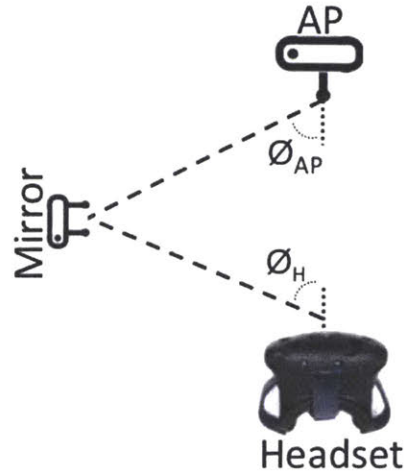


Figure 4-10: **Localizing the Mirror:** MoVR finds the location of the mirror in the VR setup by intersecting the line-of-sight angle from the AP to mirror (ϕ_{AP}) and the line-of-sight angle from the headset to the mirror (ϕ_H).

the VR system, one naïve solution is to ask the user to measure it during installation. However, this requires an accurate measurement, since even a small measurement error creates a significant inaccuracy in beam alignment. To avoid this, MoVR uses an automated calibration mechanism which calculates the location of the mirror with respect to the AP without any help from the user.

This calibration mechanism works by intersecting the line-of-sight angle from the AP to the mirror (ϕ_{AP}) and the line-of-sight angle from the headset to the mirror (ϕ_H), as shown in Fig. 4-10. In §4.4.2(a), we explained how MoVR estimates ϕ_{AP} . To estimate ϕ_H , MoVR first configures the AP to transmit to the mirror. Then it tries every combination of mirror transmit beam angle and headset receive beam angle. The receive beam angle which gives the highest SNR at the headset corresponds to ϕ_H . Finally, by intersecting the two spatial directions ϕ_H and ϕ_{AP} , we can determine the location of the mirror. Because the mirror location is fixed, this process only needs to happen once, during installation. Subsequently, MoVR can calculate the beam alignment between the mirror and the headset from VR tracking information and the mirror’s known location.

4.5 System Details

The last sections presented the solutions to the two challenges in using mmWave in VR systems –i.e. blockage and mobility. However, a number of system details must be addressed in order to put these solutions into practice. In particular: How do we provide connectivity at all locations within the VR space? And how do we choose between sending data via the direct link or sending by way of the mirror? This section will iron out these details and provide guidance on the system design trade-offs to maximize MoVR’s coverage and performance.

How do we provide connectivity at all locations within the VR space?

The system’s ability to provide wireless coverage throughout the VR space is limited by the fact that the line-of-sight to the headset can be blocked by the environment and/or by the user’s limbs. We addressed this problem by designing the MoVR mirror described in §4.3. However, it is still possible that the headset experiences blocking along the path to both the AP and to the mirror. To address this issue, MoVR supports multiple mirrors. Each mirror adds another path to the headset, and reduces the probability of the headset being blocked exponentially. Additionally, we recommend placing multiple antennas on the headset to reduce the probability of the user’s head blocking a line-of-sight path. However, any body parts that come between the headset and the AP can block all headset antennas, necessitating the use of a mirror.

How do we choose between the direct link and a link via a mirror? The AP, the mirrors, and the headset are equipped with a cheap, low-bitrate radio, e.g. Bluetooth, to exchange control information. In MoVR the mmWave receiver on the headset continuously monitors the SNR of its received signal and, whenever it drops below a certain threshold, the headset reports it back to the AP over Bluetooth. The AP then switches to a different link. The AP picks the mirror closest to the current location of the headset. If the SNR does not go above the desired threshold, the AP switches to the next closest mirror.

Because any small period of outage impacts the quality of the data rate, the

headset should act preemptively by looking at the time series of SNR and ordering a link change if there is a downward trend that is likely to result in outage. As demonstrated in §4.6, the switching latency cost is sufficiently small that it does not impact the user experience, even if the AP tries more than one mirror.

Can MoVR be enhanced to support two-way communication? Applications such as augmented reality (AR) require bi-directional communication between the headset and the PC. Since each mirror enables a one-way communication, in order to support such applications, the user can place two mirrors in the environment, where each mirror uses the same technique as described in §4.4 to align and track its beams. However, if these two mirrors are collocated, the solution explained in §4.3 to solve the leakage problem will not be sufficient. This is due to the fact that the output of each mirror will not only leak into its own input, but will also leak into the input of the other mirror. However, this problem can be solved by designing more complex mirrors which use RF circulator or full-duplex techniques to cancel out the signal from the other mirror. The design of such a mirror will be left for future work.

4.6 Evaluation

We have built a prototype of MoVR using off-the-shelf components. MoVR's mirror hardware consists of two phased array antennas (one for receive and one for transmit), connected to each other through a variable gain amplifier as shown in Fig. 4-5. The phased arrays consist of patch antenna elements, designed and fabricated on PCB. The outputs of the patch antennas are connected to Hittite HMC-933 analog phase shifters, which allows us to steer the antennas' beams. To create a variable gain amplifier, we use a Hittite HMC-C020 PA, a Quinstar QLW-2440 LNA and a Hittite HMC712LP3C voltage-variable attenuator. The mirror's current consumption is mainly dominated by its PA, which consumes 250mA during normal operation. For controlling MoVR's mirror and measuring its amplifier's current consumption, we built a controller board using an Arduino Due micro-controller, Analog Devices DACs, and a Texas Instruments INA169 DC current sensor, as shown in Fig. 4-11.

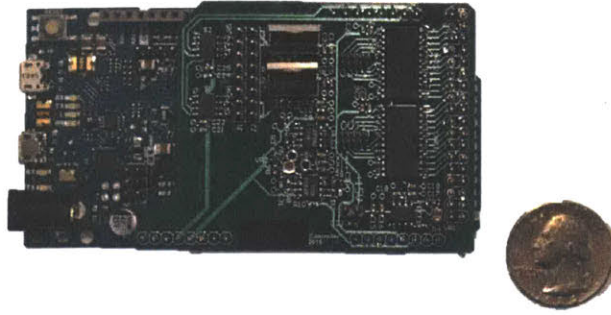


Figure 4-11: **MoVR Mirror’s Controller Board:** The figure shows our custom-designed controller board for configuring the beam alignment and the amplifier gain in real time.

In our experiments, we use the HTC VIVE virtual reality system. However, our design is also compatible with other high-quality VR platforms, such as Oculus Rift.³ We equip the VIVE headset with a mmWave receiver and the VR PC with a mmWave AP working at the 24GHz ISM band [43]. The PC has an Intel i7 processor, 16GB RAM and a GeForce GTX 970 graphics card, which is required for the HTC VIVE VR setup. The transmission power is in accordance with FCC rules [94].

We evaluate MoVR in a $5m \times 5m$ office room with standard furniture, such as desks, chairs, computers and closets.⁴ We perform experiments in both line-of-sight and non-line-of-sight scenarios.

4.6.1 Blockage During an Actual VR Game

In §4.3, we demonstrated the impact of signal blockage on the SNR and data rate of mmWave communications in a VR setup. As was shown, even blocking the signal with one’s hand significantly degrades the data rate, which is problematic. In this experiment, we investigate how often the AP’s line-of-sight to the headset is blocked in a realistic gaming scenario.

To do so, we ask a user to play a VR game (The Lab Solar System) while we extract the location information of the headset, access point, and two game controllers (held

³Although these systems use different technologies to track the user and headset, they all provide the location and position of the headset very accurately.

⁴Our test environment is the same as what HTC VIVE recommend for operation. For safety reason, they also suggest to move the furniture outside of the game area [12].

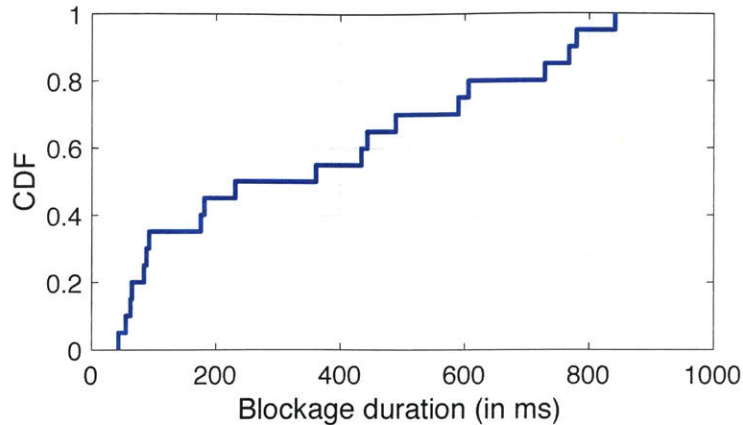


Figure 4-12: **Blockage Duration:** The figure shows the CDF of the duration of line-of-sight blockages which happened during a 5-minute VR game. The figure shows that the median blockage durations is 245ms. This is really problematic for the VR application since the duration of it's display frame is only 10ms.

by the player's hands) from the VR tracking system. Using this information, we find the equation of the line between the headset and the base station. Then, as the user plays the game, we check if the locations of the user's hands (i.e. controllers' locations) lie on that line. If either hand lies on the line, we conclude that the line-of-sight path between the headset and the base station is blocked. Our results show that the line-of-sight was blocked 20 times during a 5 minute game. Fig. 4-12 plots the CDF of the durations of these 20 cases. The figure shows that the median blockage duration is 245ms. Note that the VR frame rate is only 10ms. Hence, 245ms of blockage is highly detrimental to the user's VR experience. Our results confirm that line-of-sight blockages happen very often during VR games and persist long enough to degrade the VR experience.

4.6.2 MoVR's Mirror Performance

Next, we would like to investigate how effective MoVR is in addressing the blockage problem. We place the AP on one side of the room and the mirror on an adjacent side, as shown in Fig. 4-2.⁵ We place the headset at a random location and orientation. The AP transmits packets of OFDM symbols and the headset receives these packets

⁵Note, MoVR does not require the user to place the mirror carefully in a specific location, since its calibration mechanism is able to automatically localize the mirror.

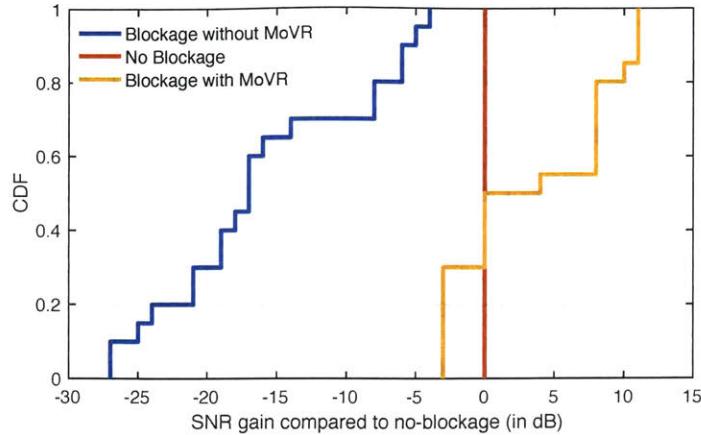


Figure 4-13: **MoVR’s Mirror Performance:** Figure shows SNR gain compared to the *No-Blockage* in all three scenarios: *No-Blockage*, *Blockage-without-MoVR* and *Blockage-with-MoVR*.

and computes the SNR. We perform the experiment for 20 runs, changing the location and orientation of the headset for each run. We repeat each run for three scenarios:

- *No-Blockage:* In this scenario, there is a clear, direct path between the AP and the headset receiver. The AP and headset have aligned their beams along this path.
- *Blockage-without-MoVR:* In this scenario, the direct path from the AP is blocked. In the absence of a MoVR mirror, the best approach is to try to reflect the signal off of a wall or some other object in the environment. Thus, to find the best SNR possible without a mirror, we make the AP and the headset try all possible beam directions and pick the one that maximizes the SNR. The headset reports this maximum SNR.
- *Blockage-with-MoVR:* Here, we have the same blockage as in the previous scenario, but the system is allowed to use the MoVR mirror to reflect the signal as described in the earlier sections.

Fig. 4-13 compares the SNRs in all three scenarios. The figure plots the CDF of the SNR Gain relative to the SNR without blockage, defined as follows:

$$SNR\ Gain\ [dB] = SNR_{Scenario}[dB] - SNR_{No\ Blockage}[dB].$$

The figure shows that, in the absence of a MoVR mirror, a blockage drops the SNR by as much as 27dB, and the average SNR reduction is 17dB. As shown in §4.3, such high reduction in SNR prevents the link from supporting the required VR data rate. Thus, simply relying on indirect reflections in the environment to address blockage is ineffective.

The figure also shows that, for most cases, the SNR delivered using MoVR's mirror is higher than the SNR delivered over the direct line-of-sight path with no blockage. This is because, in those cases, the AP's distance to the mirror is shorter than its distance to the headset's receiver. Thus, the presence of MoVR's mirror along the path, and the fact that it amplifies the signal, counters the SNR reduction due to the longer distances to the headset. The figure further shows that, in some cases, MoVR performs 3dB worse than the no blockage scenario. This loss does not affect the data rate because, in these cases, the headset is very close to the AP, which provides a very high SNR (30dB) at the headset's receiver. This SNR is much higher than the 20dB needed for the maximum data rate. This experiment shows that MoVR's mirror enables a high data rate link between a VR headset and a PC even in the presence of blockage.

4.6.3 MoVR's Beam Alignment and Tracking Performance

As explained earlier, beam alignment is essential for mmWave links. In this section, we first investigate the accuracy of aligning the beam between the AP and the mirror. We then evaluate the accuracy of the beam alignment to the headset and the latency involved in establishing the alignment.

(a) Accuracy of beam alignment between the AP and the mirror: We evaluate MoVR's ability to find the best beam alignment between the AP and the mirror. We place the mirror somewhere in our testbed and estimate the angle which provides the best beam alignment between it and the AP, using the method described in §4.4.2. We repeat the experiment for 100 runs, changing the mirror location and orientation each time. We compare this to the ground truth angle, calculated from the locations of the AP and mirror. We use a Bosch GLM50 laser distance measurement

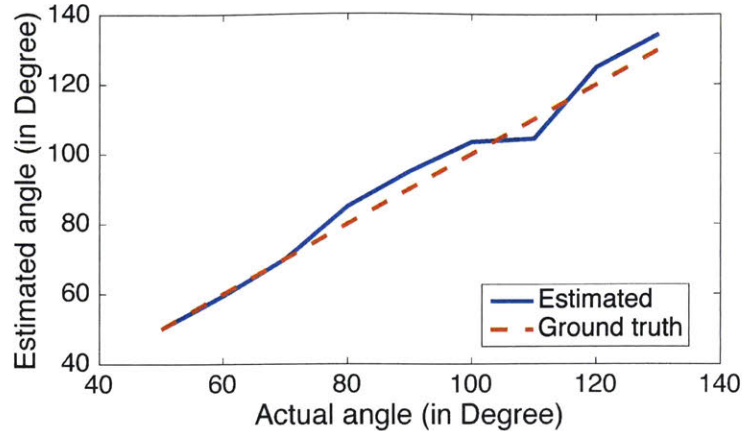


Figure 4-14: **AP to Mirror Beam Alignment Accuracy:** Angle estimated by MoVR (blue) versus the ground truth angle (red).

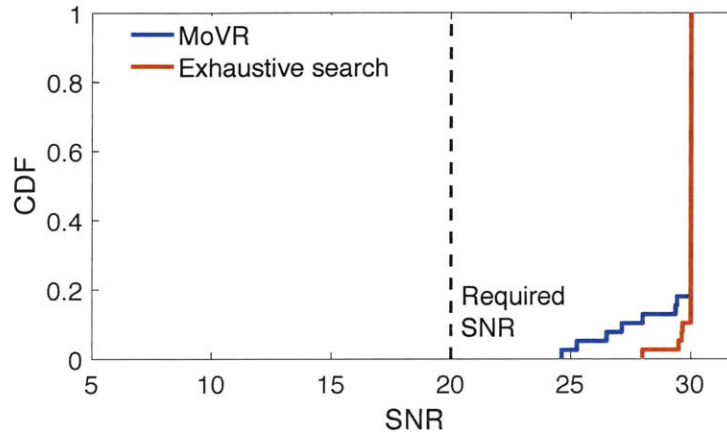


Figure 4-15: **MoVR Beam Alignment Accuracy:** The SNR of the signal at the VR receiver for two different scenarios: (1)MoVR’s beam alignment algorithm and (2)exhaustive search. The figure shows that MoVR’s beam alignment algorithm performs 4dB worse than exhaustive search in some cases. However, this loss does not affect the data rate, since the SNR is always much higher than the 20dB needed by the VR headset.

tool to measure these locations to within a few millimeters.

Fig. 4-14 plots the angle estimated by MoVR versus the ground truth angle. The figure shows that MoVR estimates the angle of best beam alignment to within 2 degrees of the actual angle. Note that since the beam-width of our phased array is ~ 10 degrees, such small error in estimating the angle results in a negligible loss in SNR.

(b) **Beam alignment accuracy for the whole system:** As explained in §4.4,

MoVR leverages the location information to track and align the transmitter's and receiver's beams. In this experiment, we evaluate the performance of MoVR in finding the best beam alignment to the headset (either from the AP or from the mirror). We place the headset at a random location and orientation in our testbed and compute the SNR it receives in two different scenarios: (1) MoVR's beam alignment algorithm and (2) Exhaustive search, which tries all possible combinations of AP, mirror and headset beam directions and picks the set which provide the highest SNR. We repeat this experiment 40 times, changing the location and orientation of the headset each time. Fig. 4-15 plots the results of this experiment. For most cases, MoVR's algorithm performs as well as Exhaustive search in finding the best beam alignment. MoVR's beam alignment performs 4dB worse than Exhaustive search in some cases, but this loss does not affect the data rate since the SNR is always much higher than the 20dB needed by the VR headset.⁶ This result is significant since Exhaustive search requires trying all possible beam alignments, and hence introduces much latency and overhead. In contrast, given an initial calibration, MoVR obtains its alignment for free by leveraging the location information already available to the VR system.

(c) Beam alignment latency for the whole system: Next, we evaluate the capability of MoVR to perform beam alignment and tracking in real time. MoVR's beam alignment process includes multiple sources of delay, which sum to the total latency of the system. First, it takes $1ms$ for the VR tracking to update the headset position[15]. Our beam alignment algorithm, implemented in C++, uses this position to calculate new beam angles in $0.9\mu s$. Finally, MoVR's hardware (including the DACs and phase shifters) takes $1.7\mu s$ to reconfigure the beam [3, 4]. Given that our computation and the hardware reaction time are on the order of a few microseconds, the total delay is dominated by the VR location tracking delay, which is $1ms$. This delay is intrinsic to the VR system and is low enough to support the VR frame rate.

⁶ The maximum SNR is limited at 30dB because of the dynamic range of the hardware.

4.6.4 MoVR System Performance

Finally, we would like to evaluate the system as a whole and its ability to deliver the desired performance as the player moves around anywhere in the room. We place the headset at a random location and orientation in our VR testbed, and block the direct path between it and the AP with a hand. We then compute the SNR that the headset receives for three different scenarios: (1) No mirror, which tries all possible combinations of AP and headset beam directions and picks the one which provides the highest SNR; (2) Fixed gain mirror, where there is a mirror with a fixed amplification gain in our setup; and (3) MoVR, where we use our tracking algorithm and a mirror with our automatic gain control algorithm. We repeat this experiment 40 times, changing the location and orientation of the headset each time.

Fig. 4-16 plots the results of this experiment. The figure shows the received SNR at the headset for different room locations and for each scenario. Our results show that, in the presence of blockage, having a mirror with a fixed gain improves the SNR over relying on indirect reflections from the environment. However, there are still some locations with SNR below the 20dB needed by the VR headset. Adapting the mirror's amplifier gain improves the performance further and allows the system to achieve high SNR (24dB or higher) in all locations. This experiment confirms the need for our automatic gain control algorithm and shows that MoVR enables a high-quality untethered virtual reality, providing the required SNR for every location in a representative VR testbed.

4.7 Discussion

This chapter presents MoVR, a system that enables a reliable and high-quality untethered VR experience via mmWave links. It provides a sustainable, high-data-rate wireless link to the VR headset even in the presence of blockage and mobility. In particular, it overcomes blockage of the mmWave link by introducing a smart and simple mmWave mirror that can reconfigure itself and adapt its angles of incidence and reflection. Further, MoVR introduces a novel algorithm that combines VR head-

set tracking information with RF-based localization to quickly steer the mmWave radios' beams and keep them aligned as the player moves around. Finally, it is worth mentioning that we have focused on eliminating the high-rate HDMI connection between the PC and headset. However, the current headset also uses a USB cable to deliver power. This cable can be eliminated by using a small rechargeable battery. The maximum current drawn by the mmWave radio and the HTC Vive headset is 1500mA. Hence, a small battery (3.8x1.7x0.9in) with 5200mA capacity can run the headset for 3-4 hours [1]. An end-to-end evaluation of full system while VR data is streamed in real time and also improving the efficiency of mmWave hardware to increase the battery life are interesting avenues for future work.

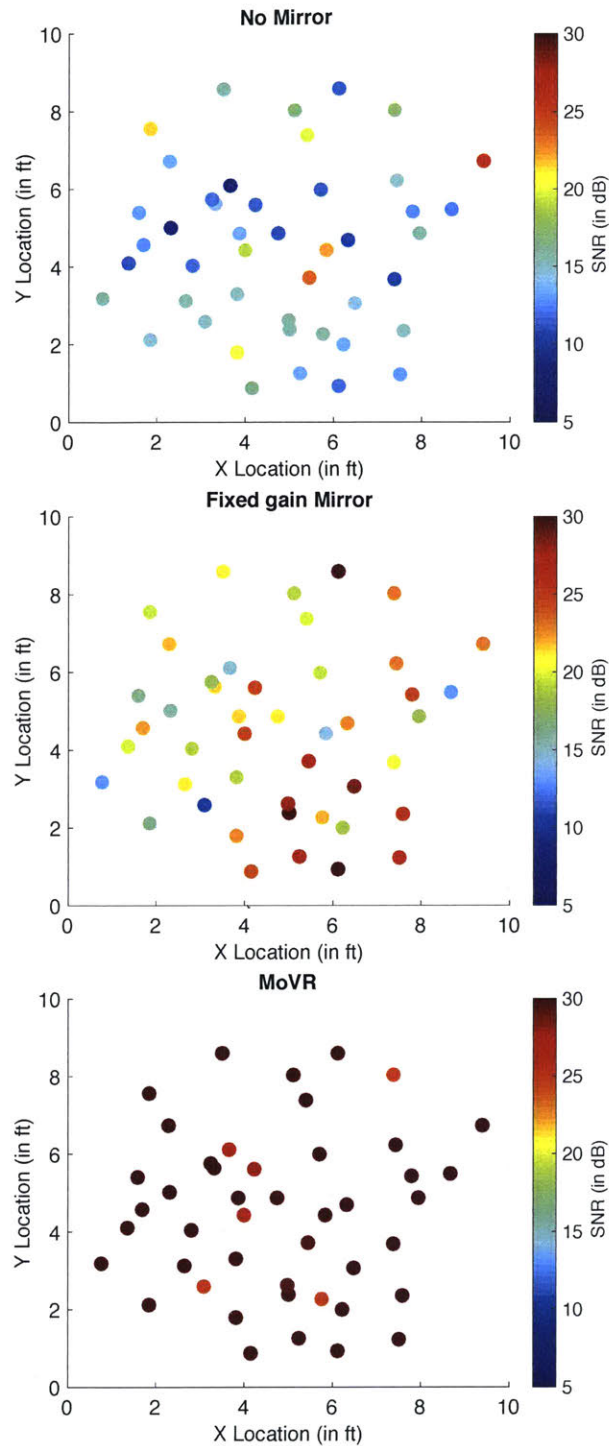


Figure 4-16: **SNR at the Headset's Receiver** The figure shows the SNR at the headset's receiver in the case of blockage for three different scenarios. (1) no mirror, (2) Fixed gain mirror, where there is a mirror with a fixed amplification gain in our setup, and (3) MoVR, where we use our beam tracking algorithm and a mirror with automatic gain control. The figure shows that during blockage the SNR is below the 20dB SNR needed by the VR headset for most locations. In contrast, MoVR enables high SNR in all locations.

Part II

Combating Energy Constraints

Chapter 5

A million-point Fourier transform chip

Many applications require real-time FFT process on long-data signal. ASIC implementations of such FFTs are challenging due to their large silicon area and high power consumption. However, the signals in these applications are sparse, i.e., the energy at the output of the FFT/IFFT is concentrated in a limited number of frequencies and the rest are negligible or zeros. Recent advances in signal processing have shown that, for such sparse signals, a new algorithm called the Sparse Fourier Transform can compute the Fourier transform more efficiently than traditional FFTs.

In this chapter, we present a VLSI implementation of the Sparse Fourier Transform algorithm. The chip implements a 746,496-point Sparse Fourier Transform, yet occupies only 0.6mm^2 of silicon area. At 0.66V, it consumes 0.4pJ/sample and has an effective throughput of 36 GS/s. The effective throughput is computed over all frequencies but frequencies with negligible magnitudes are not produced. The chip works for signals that occupy up to 0.1% of the transform frequency range (0.1% sparse). It can be used to detect a signal that is frequency hopping in a wideband, to perform pattern matching against a long code, or to detect a blocker location with very high frequency resolution. For example, it can detect and recover a signal that occupies 18MHz randomly scattered anywhere in an 18GHz band with a frequency resolution of 24kHz.

5.1 Algorithm

The Sparse Fourier Transform algorithm has three components: frequency *bucketization*, *estimation*, and *collision resolution*. Below we explain these components. We use \mathbf{x} and $\hat{\mathbf{x}}$ to denote a time signal and its Fourier transform respectively. We also use the terms: the *value* of a frequency and its *position* in the spectrum to distinguish $\hat{\mathbf{x}}_f$ and f .

5.1.1 STEP 1: Frequency Bucketization

The Sparse Fourier Transform starts by hashing the frequencies in the spectrum into buckets. Since the spectrum is sparsely occupied, most buckets will be empty and can be simply discarded. The Sparse Fourier Transform then focuses on the non-empty buckets, and computes the values of the frequencies in those buckets in what we call the estimation step.

In order to hash frequencies into buckets, we use the following basic property of the Fourier transform: *sub-sampling in the time domain causes aliasing in the frequency domain*. Formally, let \mathbf{x} be a time signal of length N , and $\hat{\mathbf{x}}$ its frequency representation. Let \mathbf{b} be a sub-sampled version of \mathbf{x} , *i.e.*, $\mathbf{b}_i = \mathbf{x}_{i \cdot p}$ where p is the sub-sampling factor. Then, $\hat{\mathbf{b}}$, the FFT of \mathbf{b} is an aliased version of $\hat{\mathbf{x}}$, *i.e.*:

$$\hat{\mathbf{b}}_i = \sum_{m=0}^{p-1} \hat{\mathbf{x}}_{i+m(N/p)}. \quad (5.1)$$

Thus, an aliasing filter is a form of bucketization in which frequencies equally spaced by an interval N/p hash to the same bucket, *i.e.*, frequency f hash to bucket $i = f \bmod N/p$, as shown in Fig. 5-1. Further, the value in each bucket is the sum of the values of only the frequencies that hash to the bucket as shown in Eq. 5.1.

Now that we hashed the frequencies into buckets, we can leverage the fact that the spectrum of interest is sparse and hence most buckets have noise and no signal. The Sparse Fourier Transform compares the energy (*i.e.*, the magnitude square) of a bucket with the noise level and considers all buckets whose energy is below a threshold

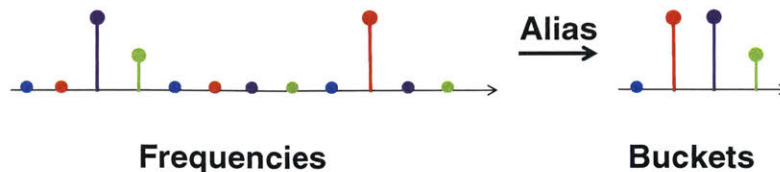


Figure 5-1: **Bucketization using Aliasing Filter:** Sub-sampling a signal by $3\times$ in the time domain, results in the spectrum aliasing. Specifically, the 12 frequency will alias into 4 buckets. Frequencies that are equally spaced by 4 (shown with the same color) end up in the same bucket.

to be empty. It then focuses on the occupied buckets and ignores empty buckets.

5.1.2 STEP 2: Frequency Estimation

Next, for each of the occupied buckets we want to identify which frequencies created the energy in these buckets, and what are the values of these frequencies. If we can do that, we then have recovered a complete representation of the frequencies with non-zero signal values, *i.e.*, we acquired the full signal in the Fourier domain.

Recall that our spectrum is sparse; thus, as mentioned earlier, when hashing frequencies into buckets many buckets are likely to be empty. Even for the occupied buckets, the sparsity of the spectrum means that many of these buckets will likely have a single non-zero frequency hashing into them, and only a small number will have a collision of multiple non-zero (or occupied) frequencies.

In the absence of a collision, the value of the occupied frequency is the value of the bucket. Said differently, the value of a bucket after aliasing, $\hat{\mathbf{b}}_i$ is a good estimate of the value of the occupied frequency $\hat{\mathbf{x}}_f$ in that bucket, since all other frequencies in the bucket have zero signal value (only noise).

Although we can easily find the value of the non-zero frequency in a bucket, we still do not know its frequency position f , since aliasing mapped multiple frequencies to the same bucket. To compute f , we leverage the *phase-rotation property* of the Fourier transform, which states that a shift in time domain translates into phase rotation in the frequency domain. Specifically, we perform the process of bucketization again, after shifting the input signal by τ . Since a shift in time translates into phase rotation in the frequency domain, the value of the bucket of changes from $\hat{\mathbf{b}}_i = \hat{\mathbf{x}}_f$

to $\widehat{\mathbf{b}}_i^{(\tau)} = \widehat{\mathbf{x}}_f \cdot e^{2\pi j \cdot f \cdot \tau / N}$. Hence, using the change in the phase of the bucket, we can estimate our frequency of interest and we can do this for all buckets that do not have collisions.

A point however is worth noting. Recall that the phase wraps around every 2π . Hence, the value of τ has to be equal to 1 to avoid the phase wrapping around for large values of f .

5.1.3 STEP 3: Collision Detection and Resolution

So far we explain how we estimate the value and the position of non-zero frequencies in the absence of collision. In the following, first we describe how we distinguish the buckets that have a single non-zero frequency from those that have a collision, and then we explain how we resolve the colliding frequencies.

Collision Detection

Again we use the *phase rotation* property of the Fourier transform to determine if a collision has occurred. Specifically, if the bucket contains a single non-zero frequency, *i.e.*, no collision, then performing the bucketization with a time shift τ causes only a phase rotation of the value in the bucket but the magnitude of the bucket does not change *-i.e.*, with or without the time shift, $\|\widehat{\mathbf{b}}_i\| = \|\widehat{\mathbf{b}}_i^{(\tau)}\| = \|\widehat{\mathbf{x}}_f\|$. In contrast, consider the case where there is a collision between, say, two frequencies f and f' . Then the value of the bucket without a time-shift is $\widehat{\mathbf{b}}_i = \widehat{\mathbf{x}}_f + \widehat{\mathbf{x}}_{f'}$, while its value with a time-shift of τ is $\widehat{\mathbf{b}}_i^{(\tau)} = \widehat{\mathbf{x}}_f \cdot e^{2\pi j \cdot f \tau / N} + \widehat{\mathbf{x}}_{f'} \cdot e^{2\pi j \cdot f' \tau / N}$. Since the colliding frequencies rotate by different phases, the overall magnitude of the bucket will change. Thus, we can determine whether there is a collision or not by comparing the magnitudes of the buckets with and without the time-shift.¹

¹Even if one occasionally falsely detects a collision when there is a single frequency, The Sparse Fourier Transform can still correct this error. This is because the collision resolution step described next will estimate the values of the presumed colliding frequencies to zero.

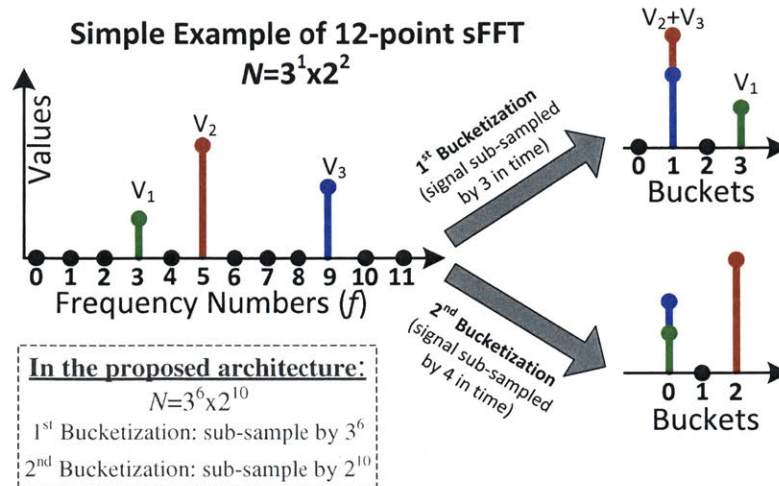


Figure 5-2: **Resolving Collisions with Co-prime Filters:** Using 2 co-prime aliasing filters, we ensure the frequencies that collide in one filter will not collide in the second. For example, frequencies 5 and 9 collide in the first filter. But frequency 5 does not collide in the second which allows us to estimate it and subtract it.

Collision Resolution

To reconstruct the full spectrum, we need to resolve the collisions *-i.e.*, for each non-zero frequency in a collision we need to estimate its value \hat{x}_f and position f .

One approach to resolve collisions is to bucketize the spectrum multiple times using aliasing filters with co-prime sampling rates (FFTs with co-prime sizes). Two numbers are co-prime if their greatest common divisor is one. Co-prime aliasing filters guarantee (by the Chinese remainder theorem) that any two frequencies that collide in one bucketization will not collide in the other bucketizations. To better understand this point, consider the example in Fig. 5-2. The first time we bucketize, we use an aliasing filter that sub-samples the time signal by a factor of 3. In this case, the two frequencies labeled in red and blue collide in a bucket whereas the frequency labeled in green does not collide, as shown in the figure. The second time we bucketize, we use an aliasing filter that sub-samples by 4. This time the blue and green frequencies collide whereas the red frequency does not collide. Now we can resolve collisions by iterating between the two bucketizations. For example, we can estimate the green frequency from the first bucketization, where it does not collide. We subtract the green frequency from the colliding bucket in the second bucketization to obtain the

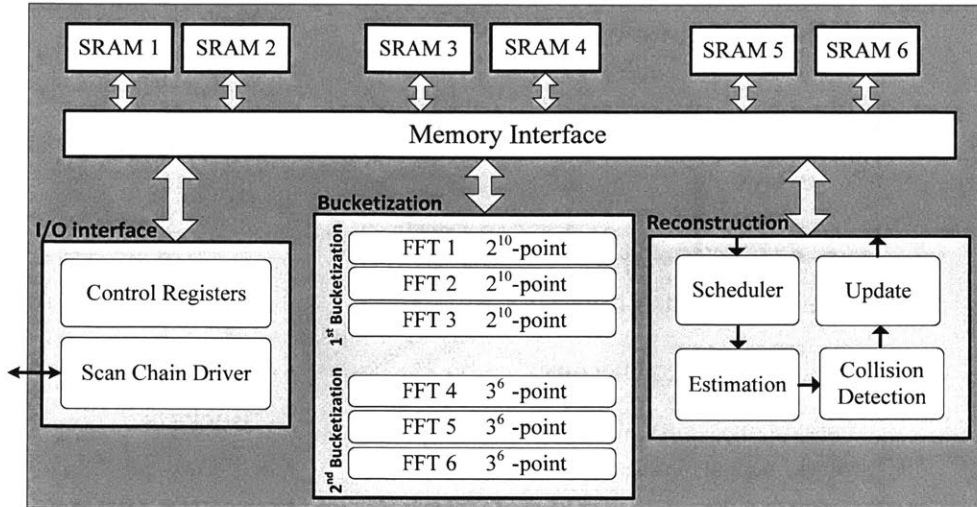


Figure 5-3: **Block Diagram of the $2^{10} \times 3^6$ -point Sparse Fourier Transform**
 The design has three main blocks: I/O interface, Bucketization and Reconstruction. They operate in parallel on three different Sparse Fourier Transform frames.

blue frequency. We then go back to the first bucketization and subtract the blue frequency from the bucket where it collides to obtain the red frequency.

Thus, by using co-prime aliasing filters to bucketize and iterating between the bucketizations –*i.e.*, estimating frequencies from buckets where they do not collide and subtracting them from buckets where they do collide– we can recover the spectrum.

5.2 Architecture

The block diagram of the Sparse Fourier Transform chip is shown in Fig. 5-3, where a 12-bit 746,496-point ($2^{10} \times 3^6$ -point) Sparse Fourier Transform is implemented. The chip includes three main blocks: I/O interface, Bucketization and Reconstruction block. First, the I/O interface block gets the digitized complex samples for the signal of the interest as an input, and it writes them in the memory. Then the Bucketization block uses these samples to perform the bucketization step of the algorithm. In order to do this task, 6 FFTs are used where they compute the Fourier transform of the sub-sampled signal. Three of them are 2^{10} -point FFTs and the other three are 3^6 -point FFTs. The inputs to the 2^{10} -point FFTs are the signal sub-sampled by 3^6 , while the inputs to the 3^6 -point FFTs are the signal sub-sampled by 2^{10} . FFTs of

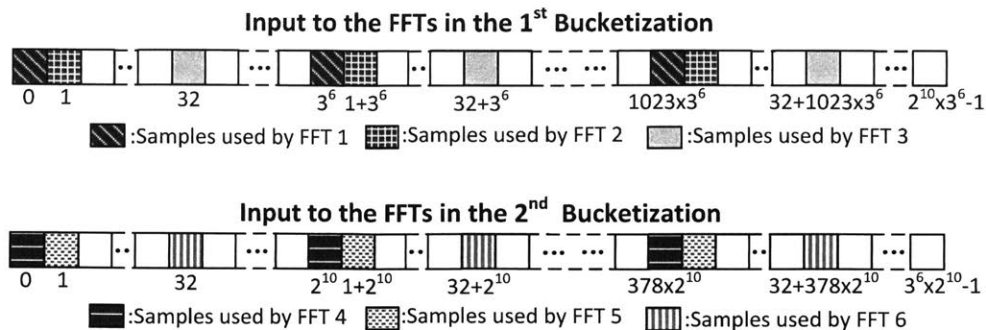


Figure 5-4: **FFT's Input.** The figure shows the input samples to 6 FFTs.

sizes 2^{10} and 3^6 are chosen since they are co-prime and can be implemented with simple low-radix FFTs. Three FFTs of each size are used in order to compute the Fourier transform of the sub-sampled signal for three different time shifts. Fig. 5-4 shows the input samples to each FFT. The first FFT computes the Fourier transform of the signal sub-sampled by 3^6 , without any time shift. The second FFT computes the Fourier transform of the signal sub-sampled by 3^6 , shifted by one samples. The third FFT computes the Fourier transform of the signal sub-sampled by 3^6 , shifted by 32 samples. The inputs to the other three FFTs are similar except that they are sub-sampled by 2^{10} samples. It is worth noting that in principle, time shifts of 0 and 1 are sufficient for the algorithm to perform. However, the third shift (i.e. shift by 32) is used to increase the robustness to noise. The output of the bucketization block is then written to the memory. Finally, The Reconstruction block reads these values and first performs Frequency Estimation step of the algorithm, and then it performs the Collision Detection and Resolution steps.

The designed architecture has six SRAMs, as shown in Fig. 5-3. One 1024-word and one 729-word SRAMs are used for three 2^{10} -point and three 3^6 -point FFTs, respectively. In order to improve the throughput, SRAMs are triplicated to enable pipelined operation of the I/O interface, Bucketization and Reconstruction blocks. Thus, three Sparse Fourier Transform frames exist in the pipeline. In the following, we explain the micro architecture of FFTs, Estimation and Collision Detection blocks in more details.

5.2.1 FFT's Micro-architecture

The micro-architecture of the 2^{10} -point FFT is shown in Fig. 5-5. Each 2^{10} -point FFT uses one radix-4 butterfly to perform an in-place FFT, which is optimized to reduce area and power consumption as follows. First, the FFT block performs read and write operations at even and odd clock cycles, respectively, which enables utilizing single port SRAMs. A single read operation provides three complex values, one for each radix-4 butterfly. The complex multiplication is computed over two clock cycles using two multipliers for each butterfly. Second, a twiddle factor (TWF) control unit is shared between the three butterflies. Third, the block floating point (BFP) technique is used to minimize the quantization error [200]. BFP is implemented using a single exponent shared between FFTs, and scaling is done by shifting in case of overflow. Round-half-away-from-zero is implemented by initializing the accumulator registers with 0.5 LSB and truncating the results. The 3^6 -point FFTs are similar, but use radix-3 butterflies.

5.2.2 Estimation's Micro-architecture

The micro-architecture of Estimation block is shown in Fig. 5-6, where it includes Phase Detector and Frequency Recovery units. The three phase detector units are implemented based on CORDIC algorithm. The micro architecture of the Phase Detector is shown in Fig. 5-7, where it computes the phase of the complex values. Note that the complex values are output of the FFTs which are in Cartesian form. The output of the Phase Detectors are shown by ϕ_1 , ϕ_2 and ϕ_3 which present phases for the FFT of the original signal, of the signal which was shifted by one samples, and of the signal which was shifted by 32 samples, respectively. Then, these phases are passed to the Frequency Recovery unit which estimates the frequency position of each non-zero frequency.

The Frequency Recovery unit operates in two steps, as shown in Fig. 5-6:

- *Accurate phase change estimation:* First, we need to estimate the exact value of the phase change caused by the time shift. Therefore, ϕ_2 and ϕ_3 are compared

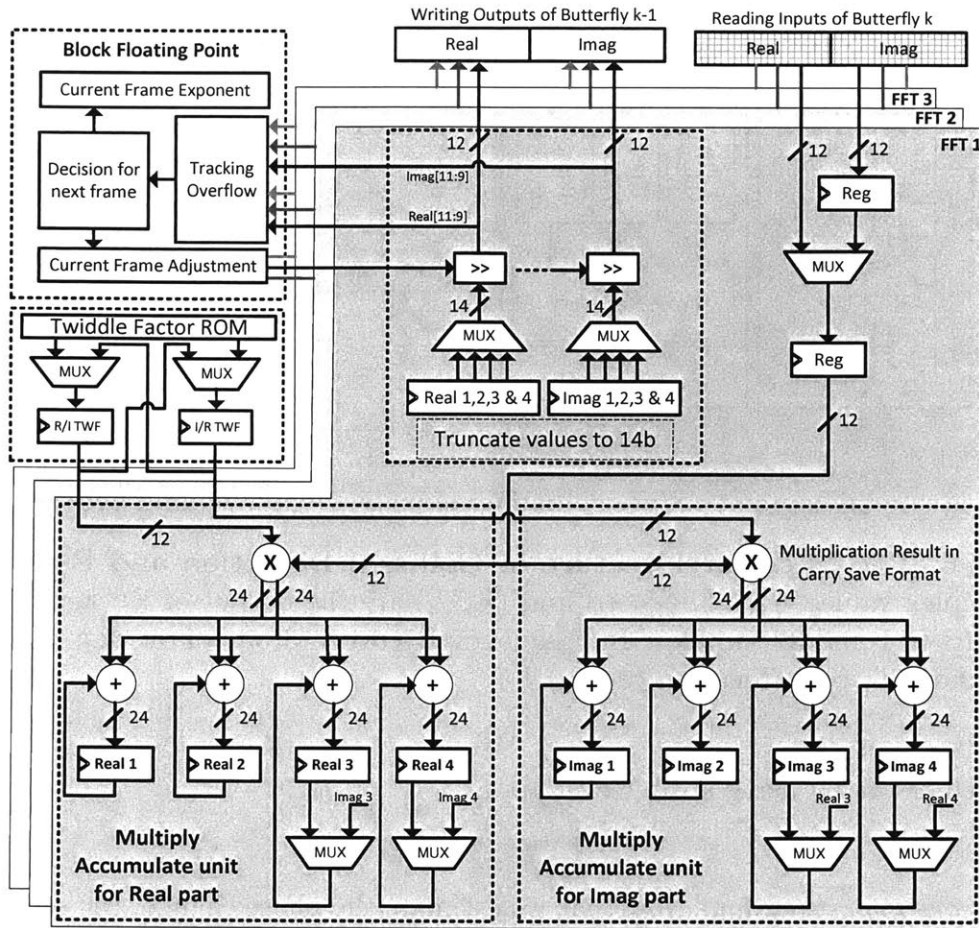


Figure 5-5: The Micro-architecture of the 2^{10} -point FFTs. Block floating point (BFP) is used to reduce the bit-width requirement during the computation of the FFT, while maintaining the required resolution at the output.

to ϕ_1 to compute the MSBs and LSBs of the phase change, respectively. Note that although comparing ϕ_2 to ϕ_1 is sufficient to estimate both MSBs and LSBs of phase change, it will be very sensitive to noise. In fact, a small amount of noise can easily cause error in the LSB of estimated phase change. To tackle this problem, we also compare ϕ_3 to ϕ_1 . Although, due to phase wrap, this comparison will not provide accurate MSBs of phase change, but it will provide LSBs which are much more robust to noise than comparing the ϕ_2 to ϕ_1 . In summary, we compare ϕ_2 to ϕ_1 to accurately estimate the MSBs of the phase change, and we compare ϕ_3 to ϕ_1 to estimate the LSBs of the phase change. Finally, note this concatenation may cause some error. We estimate and correct

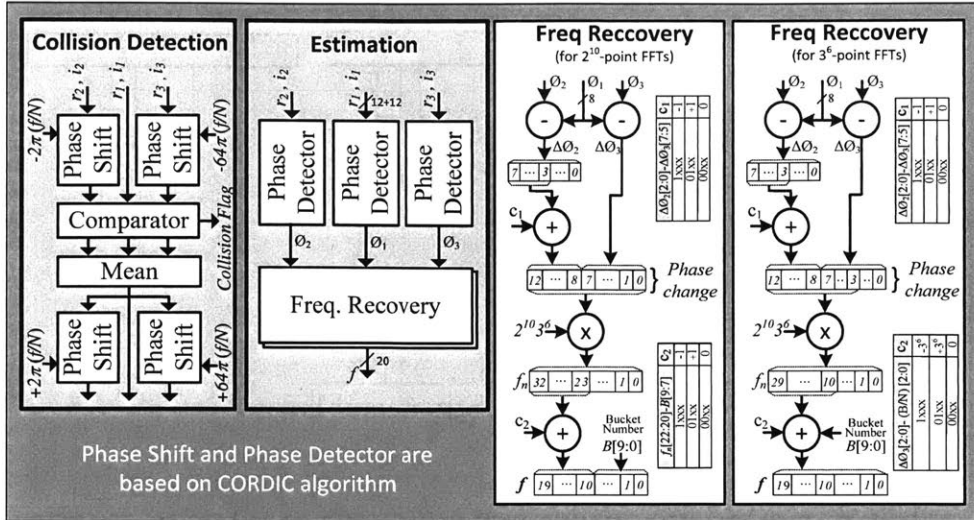


Figure 5-6: **The Micro-architecture of Collision Detection and Estimation.** The complex values (r_1, i_1) , (r_2, i_2) and (r_3, i_3) are the output of Bucketization for time-shifts 0, 1 and 32 samples. In Frequency Recovery, 3 bits of overlap are used to fix errors due to concatenation (c_1, c_2) .

such errors by using a 3-bit overlap.

- *Frequency estimation:* Now that we estimate the phase change, the second step is to estimate the frequency number from the phase change. This is simply done by multiplying the estimated phase change by N which is $2^{10} \times 3^6$ in our design. However, this frequency number may have errors in the LSBs due to quantization noise. We correct any such errors by using the bucket number to recover the LSBs of the frequency number. This can be done because all frequencies in a bucket share the same remainder B ($B = f \bmod M$, where f is the frequency number and M is the FFT size), which is also the bucket number. Thus, in the Frequency Recovery block associated with the 2^{10} -point FFTs, the bucket number gives the 10 LSBs of the frequency number. However, in the Frequency Recovery for the 3^6 -point FFTs, the LSBs cannot be directly replaced by the bucket number since $M=3^6$ is not a power of 2. Instead, the remainder of dividing the frequency number by 3^6 should be calculated and subtracted from the frequency number. The bucket number is then added to the result of the subtraction. In our implementation, calculating and subtracting the remainder

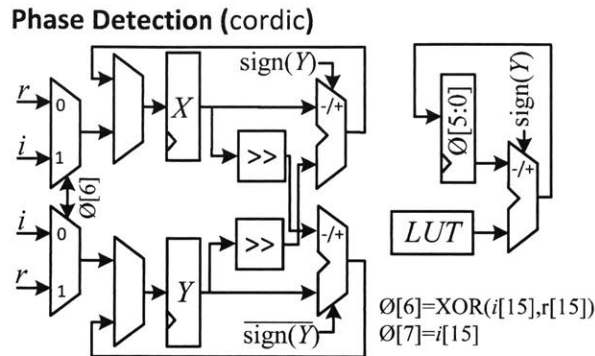


Figure 5-7: **The Micro-architecture of Phase Detection**The Phase Detection unit is based on CORDIC algorithm and used in Estimation block to find the phase of Cartesian values.

is done indirectly by truncating the LSBs of the phase change.

5.2.3 Collision Detection's Micro-architecture

The micro-architecture of Collision Detection is shown in Fig. 5-6, where it compares the values of the buckets with and without time-shifts to detect a collision. Recall that if the bucket contains a single non-zero frequency, *i.e.*, no collision, performing the bucketization with a time shift causes only a phase rotation of the value in the bucket but the magnitude of the bucket does not change. Since our complex values are in Cartesian form, instead of calculating the magnitude and comparing them, we compare their real and imaginary parts. However, in order to do this, we first need to remove the phase change due to time shifts. This is done by using two Phase Shift units which are designed based on CORDIC algorithm. After removing the phases, we compare the real and imaginary part of the three values. If a collision is detected, it flags the bucket as a collision. Otherwise, the three values are first averaged to reduce the impact of noise, and then it is shifted back using Phase Shift units. These values are used to update the output of the Sparse Fourier Transform in the memory.

5.2.4 Memory Structures

The Sparse Fourier Transform chip has six SRAMs. One 1024-word and one 729-word SRAMs are used for three 2^{10} -point and three 3^6 -point FFTs, respectively. As shown

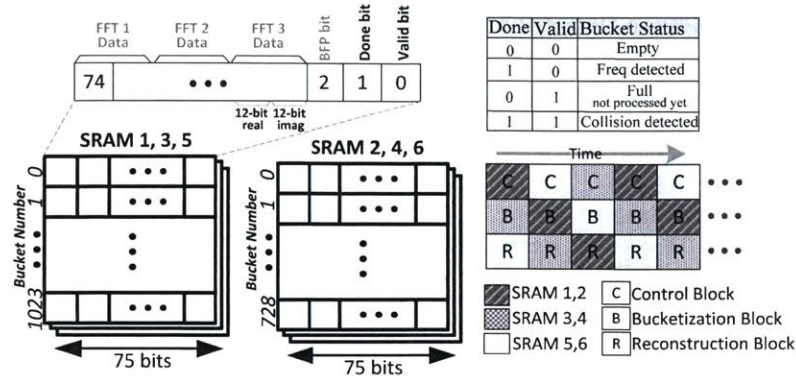


Figure 5-8: **Memory structures.** SRAMs 1, 3 and 5 (used by 2^{10} -point FFT cores) rotate between three blocks: I/O interface, Bucketization and Reconstruction. Similarly, SRAMs 2, 4 and 6 are used by 36-point FFT cores. The BFP bit indicates whether the values in the bucket are using the exponent of Block Floating Point (BFP) or not.

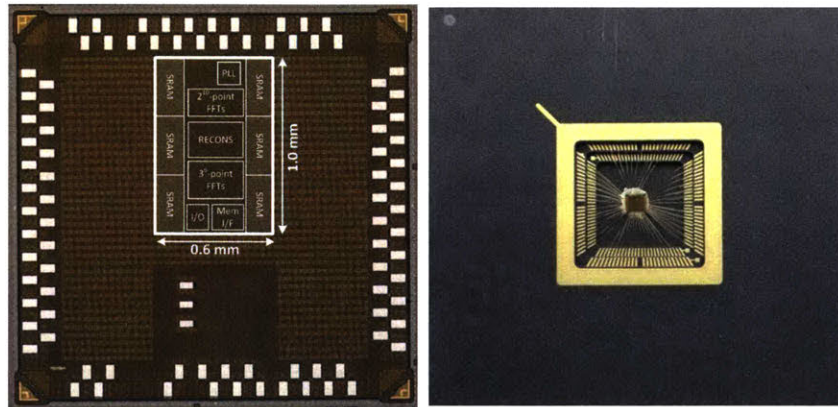


Figure 5-9: **Die Photo of the Sparse Fourier Transform Chip:**The Sparse Fourier Transform testchip is fabricated in IBM’s 45nm SOI technology and it occupies $0.6mm^2$.

in Fig. 5-8, each word is 75 bits where two bits are used to store the status of the bucket, one bit is used to indicate if BFP is used, and 72 bits are used to store the complex values (i.e. 12 bits real and 12-bits imaginary) of three FFTs output. As mentioned earlier, SRAMs are triplicated to enable pipelined operation of the I/O interface, Bucketization and Reconstruction blocks. Fig. 5-8 also shows the timing diagram of the memory usage.

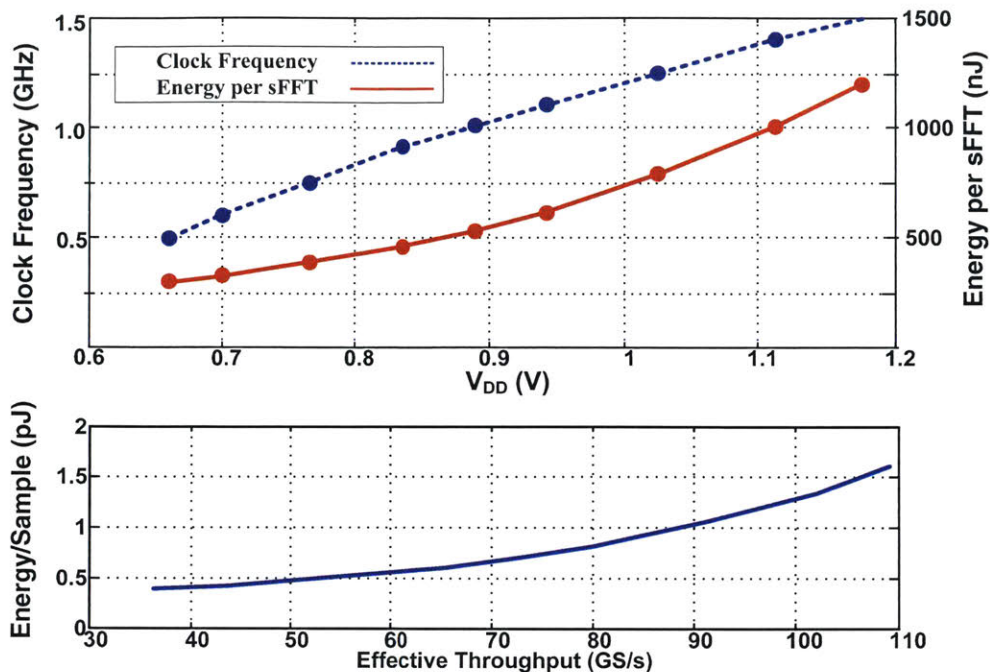


Figure 5-10: **Chip Measurement Results:**The figure shows (a) measured energy and operating frequency for a range of voltage, and (b) throughput versus energy per sample for computing a 746,496-point sparse Fourier transform

5.3 Related Work and Results

The Sparse Fourier Transform testchip is fabricated in IBM's 45nm SOI technology, as shown in Fig. 5-9. It occupies $0.6mm^2$ including SRAMs. Fig. 5-10 shows energy per Fourier transform and maximum operating frequency for a range of supply voltages. At a supply voltage of 1.18V, the chip operates at a maximum frequency of 1.5GHz, resulting in an effective throughput of 109GS/s. At this frequency, the measured energy efficiency is $1.2\mu J$ per 746,496-point Fourier transform. Reducing the clock frequency to 500MHz enables an energy efficiency of 298nJ per Fourier transform at 0.66V supply voltage. Fig. 5-10 also shows the energy per sample for different effective throughput. The energy per sample is calculated by measuring the energy consumed for computing the Fourier transform normalized by the Fourier transform size. The chip consumes 0.4pJ and 1.6pJ per sample for 36.4GS/s and 109.2GS/s effective throughput, respectively.

Since no prior ASIC implementations of the Sparse Fourier Transform exists, we

	[160]	[72]	[194]	This Work
Technology	65nm	90nm	65nm	45nm
Signal Type	Any Signal	Any Signal	Any Signal	Sparse Signal
Size	2^{10} -point	2^8 -point	2^7 to 2^{11} -point	$3^6 \times 2^{10}$ -point
Word Width	16 bits	10 bits	12 bits	12 bits
Area	$8.29mm^2$	$5.1mm^2$	$1.37mm^2$	$0.6mm^2$
Effective Throughput	240MS/s	2.4GS/s	1.25-20MS/s	36.4-109.2GS/s
Energy/Sample	17.2pJ	50pJ	19.5-50.6 pJ	0.4-1.6pJ

Table 5.1: **Comparison of Sparse Fourier Transform chip with traditional FFT chips.** The measured energy efficiency and performance of the Sparse Fourier Transform chip compared to published FFTs. For applications with frequency-sparse signals, the Sparse Fourier Transform enables 40 times lower energy per sample.

compare the performance of the Sparse Fourier Transform with recent low power implementations of the traditional FFTs [72, 160, 194]. Table 5.1 compares these Fourier transforms in terms of throughput and energy. It should be noted that other works can compute Fourier transform for both sparse and non-sparse signals while the Sparse Fourier Transform chip can compute the Fourier transform of sparse signals with sparsity up to 0.1. However, for such sparse signals, the chip delivers 40 times lower energy per sample for a 3^6 times larger Fourier transform size. Finally, the Sparse Fourier Transform chip computes a 746,496-point Fourier transform in $6.8\mu s$ which is 88 times faster compare to a C++ implementation of the algorithm running on a 3.4GHz Quad core i7 CPU.

5.4 Discussion

Many applications require computing Fourier transforms of large signals in real-time. In the context of IoT, it is essential to perform such computation in a very low energy budget. Our approach to achieve such requirement is to leverage a new algorithm called Sparse Fourier Transform which computes the Fourier transform much more efficient than traditional FFTs. In order to achieve even higher efficiency, we design an application specific integrated circuit for it. In this chapter, we presents a VLSI implementation of the Sparse Fourier Transform algorithm. The chip implements a 746,496-point Sparse Fourier Transform. The chip is fabricated in IBM's 45nm SOI

technology and occupies 0.6mm^2 of silicon area. It consumes 298 to 1206nJ of energy at a throughput of 36 to 109GS/s, utilizing scaled supply voltage of 0.66 to 1.18V. The chip works for signals that occupy up to 0.1% of the transform frequency range while it is achieving an 88 times reduction in run-time compared to a C++ implementation on an i7 CPU.

Chapter 6

An E-Toll Transponder Network for Smart Cities

Electronic toll collection transponders are simple devices consisting of a battery-powered RFID. They are perhaps among the most-widely used wireless communication technologies. In the US, depending on the state, 70% to 89% of the cars have such transponders [6, 26, 140, 178]. Further, the numbers are growing rapidly. The state of Pennsylvania has announced that E-ZPass will be mandatory on all highways in 2018. The state of California already requires drivers to have the transponder mounted on the windshield per state law in order to drive in the Express-Lanes [26, 177]. Other states are following suit motivated by Congress's decision to have a national electronic toll-collection system [19, 26]. Because of this wide-deployment and anticipated growth, multiple businesses are looking into leveraging e-toll transponders to deliver new services. For example, e-toll transponders are currently used to pay for food at some drive-through restaurants [95], and to automate payment at parking garages [20].

More generally, there is a big opportunity for using e-toll transponders to enable smart cities. For example, the city could deploy readers on traffic lights to query the transponders and track the number of cars at every intersection. It can then use the information to adapt the timing of traffic lights to minimize the average wait time for the green light. It can also leverage RF-based localization to localize cars using

their transponders' signals, detect cars that run a red-light, and automatically charge their accounts for a ticket. Readers deployed on street-lamps can detect speeding on every street in the city and ticket the offending car, without the need for car-mounted radars and hidden police officers. The same infrastructure can deliver smart street-parking systems, where a user parks anywhere on the street, the city localizes his car, and automatically charges his account.

Unfortunately, today there is a major challenge that hampers the use of e-toll transponders in smart city services, like the ones described above. Specifically, e-toll transponders are designed under the assumption that only one transponder transmits at any point in time, and hence have no MAC protocol to prevent collisions.¹ Collection systems use restricted deployments and highly directional antennas to ensure that only one car responds to the reader's query. Without this physical isolation, all transponders in range would transmit simultaneously, creating collisions. One could think of replacing the current transponders with new transponders that support a MAC protocol. Replacing the large infrastructure of deployed transponders however would take a long time and incur a major cost.² In contrast, developing for current transponders allows the cities to obtain immediate benefits even with a small installation on some of the busier streets and intersections. Yet to do that, the system has to deliver its smart services in the presence of wireless collisions.

This chapter presents Caraoke, a networked system for delivering smart services using existing e-toll transponders. Caraoke also presents a new reader design that can count, localize, and estimate the speed of the cars on the road using collision signals from their e-toll transponders. The key feature that enables Caraoke to work in the presence of collision is its ability to exploit the carrier frequency offset (CFO) of the transponders. Specifically, since e-toll transponders are active RFIDs, each device has an independent oscillator, and hence it experiences a carrier frequency offset (CFO). Traditional wireless systems view the CFO as a harmful phenomenon

¹This is unlike traditional RFIDs used in access control or retail, which have a MAC protocol.

²There were more than 26 million transponders deployed just by E-ZPass as of 2013 [5]. In addition, there are a large number of additional transponders deployed by other agencies such as FasTrak, etc.

that the receiver has to compensate for in order to correctly decode. In contrast, we show that we can leverage the CFO of the transponders to zoom in on individual transponders in the presence of collisions.

In particular, we consider the collision in the frequency domain as opposed to the time domain, and show that each collision exhibits spikes that correspond to the CFOs of the colliding transponders. Further, e-toll transponders have particularly large CFOs that span 1.2MHz [121], creating a significant separation between the spikes. Thus, we can estimate the number of transponders by counting these spikes.

We also show that we can use the differences in CFO to measure the wireless channels to the individual transponders, and hence apply RF-based localization to track cars and measure their speeds. Caraoke can also decode the IDs of the colliding transponders, say to charge a car for parking or speeding. To do this, Caraoke leverages the channels and CFO measurements to combine multiple collisions in a manner that the signals from the target transponder add up coherently, whereas the signals of other colliding transponders combine incoherently. This allows Caraoke to boost the SNR of the target transponder above the others, and enable it to decode the ID of the target transponder.

To demonstrate the practicality of Caraoke, we built Caraoke reader into a custom designed printed circuit board (PCB). Our prototype, shown in Fig. 6-8, is both small and low-cost, making it amenable to large-scale deployment. Further it is designed as a plug and play device; It connects to the Internet via an LTE modem and harvests its energy from solar power, making it easy to deploy on street-lamps.

We have evaluated Caraoke on four campus streets. We ran multiple experiments with cars that have standard E-ZPass transponders. Our results show the following:

- Caraoke can count transponders accurately despite collisions. The average error in the Caraoke estimator is 2%, and the 90th percentile is less than 5%, which is significantly more accurate than existing camera-based traffic tracking systems [135, 175].
- Caraoke can accurately localize cars into parking spots. Its average location ac-

curacy is 4 degrees. This accuracy is sufficient for detecting occupied/available parking spots between two street lamps.

- Across experiments where we varied the car speed from 10 mile/hour to 40 mile/hour, Caraoke has detected the speed to within 8% (i.e., 1 to 3 mile/hour). The same accuracy was observed in a second set of experiments conducted in an empty lot with a car speed of 50 mile/hour.³
- Caraoke successfully decodes transponder ids in the presence of collisions, but the time required to decode increases with the number of colliding transponders. In particular, decoding the ids of a pair of colliding transponders takes 4.2 ms, whereas decoding five colliding ids takes 16.2 ms.
- Measurements of the Caraoke reader show that it consumes only 9mW in average (excluding modem), which is 56× lower than what it can harvest from its solar panel.

6.1 Related Work

(a) Communication and Localization: Caraoke builds on a rich literature on RFIDs. Past research however has typically focused on EPC RFIDs, like those used in access control and inventory tracking [84, 104]. Such RFIDs *do* support a MAC protocol and hence can communicate without major collisions. In contrast, e-toll transponders use a different protocol that has no MAC support. We note however that past works [114, 185] proposed methods to decode concurrent transmissions from backscatter sensors in time domain. However, such designs are inapplicable to our scenario because they require hardware modification of the RFIDs and do not work with existing e-toll transponders.

Our work is also related to past work on RFID localization and RF-based positioning [91, 101, 186, 191, 195]. While we build on the general area of AoA localization,

³Almost all states in the US have residential speed limits below 35 mile/hour, and the maximum residential speed limit in any state is 45 mile/hour [24].

our approach differs in that it exploits CFO differences to localize the devices using colliding signals, without even decoding.

There are also commercial RFID readers which are solar powered [23]. However, unlike Caraoke, these readers can not localize, count and identify RFIDs in the presence of wireless collisions.

Finally, a vast majority of past research on issues related to CFO focuses on how to eliminate or estimate the CFO and compensate for it [152, 153, 172]. The closest to our work is [67], which advocates using the CFO of a device as an id for security purposes. None of this work however deals with collisions or the use of CFO for localization or decoding.

(b) Smart Cities: Our work is motivated by the growing interest in smart cities, where urban services are automated to improve efficiency, and reduce waste and pollution [90, 115, 134]. Past work in this area focuses on transportation research [192], software applications [78], and social and economic issues [49]. In contrast, we focus on wireless networking issues such as communication, localization, and counting in the presence of wireless collisions.

There are also a few businesses that market solutions for one of Caraoke’s applications. In particular, some apps allow a user to pay for parking using her phone [70]. Those apps however do not address the cost and overhead incurred by the city in checking for parking violations and issuing tickets. Further, they do not automatically detect the occupancy of parking spots. Alternative solutions like Streetline install a sensor in the asphalt pavement of every parking spot [167]. They need to drill the street in every spot incurring a significant cost and causing traffic disturbances. There are also traffic cameras installed in some cities for counting the cars at the corresponding traffic light and providing traffic statistics [125]. These systems are highly sensitive to occlusions, illumination, shadowing, and wind [135]. Finally, traffic radars are typically used to measure car speeds. These devices however cannot tell which speed is associated with which car. A police officer has to be around to identify the speeding car based on the orientation of the device [88]. In contrast to all of the above, Caraoke is a single system that can support all of these applications,

and address many of the drawbacks of existing solutions.

(c) **VANET:** There is a large literature on vehicle networks, or VANET. Research in that area addresses the impact of mobility on ad hoc networks [130]. It focuses on routing [127], quality of service [193], and reliability [204]. It runs on typical communication devices that support a MAC protocol, e.g., WiFi and WiMAX [82, 137]. Our work differs from this past work in objectives and techniques. Specifically, our goal is to enable smart cities to leverage the widely-deployed e-toll transponders to deliver new services such as smart-parking and real-time traffic monitoring. Our solutions target a different communication technology, namely e-toll transponders. Such transponders lack a MAC protocol, necessitating new designs that differ from those used in VANET.

6.2 Background

An e-toll transponder is an active RFID, which responds to an inquiry transmitted by the reader. The reader is typically placed in the tollbooth whereas the transponder is attached to the car's windshield. Both transponder and reader work at 915MHz. The query signal is simply a sinewave transmitted at the carrier frequency (i.e. 915MHz) for a short period of time. The transponder responds with its id, which identifies the driver's account. Fig. 6-1(a) illustrates the timing of the query signal and the transponder's response to it.

A few points are worth noting:

- In contrast to traditional RFIDs (e.g., those used in retail or access control),⁴ the active RFIDs used in e-toll transponders lack a medium access protocol (MAC). Thus, once a transponder detects the reader's signal, it immediately transmits its response. Hence, if multiple transponders are in the reader's range, they all respond leading to a collision. Toll systems avoid the need for a MAC by using highly directional antennas, and the fact that cars are separated by a

⁴Most research targets Electronic Product Code (EPC) RFIDs, which have an Aloha-style MAC protocol [85].

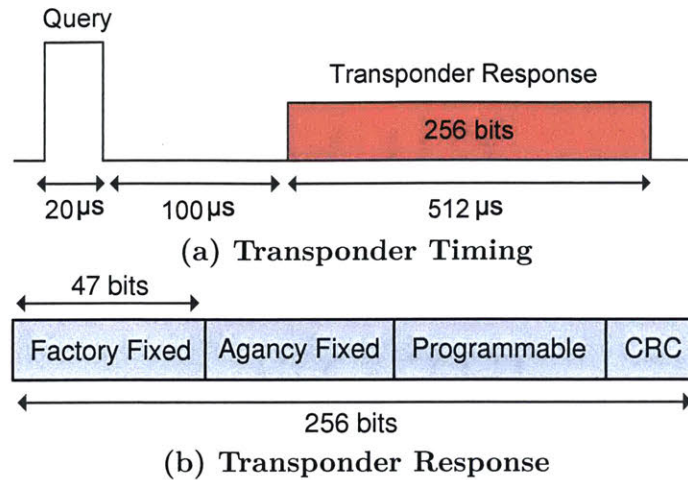


Figure 6-1: **Transponder Transmission.** The reader transmits a query signal which is a simple sine-wave at carrier frequency. The transponder responds by transmitting a 256-bit data using OOK Manchester modulation.

minimum distance.

- E-toll transponders also have a relatively large CFO. Their carrier frequencies vary between 914.3MHz and 915.5 MHz, and hence their CFO can be as high as 1.2MHz [121].
- The simplicity of the transponders results in a cheap and low power device. A transponder can work for 10 years before it runs out of battery, and it operates whether the car is on or off.
- Finally, for the purpose of this chapter, it is important to understand the properties of the transponder signal. The transponder transmits its data using on-off keying (OOK) modulation. OOK is a simple modulation, where the radio transmits a “1 bit” by transmitting the carrier frequency, and transmits a “0 bit” by staying silent. This means that the transponder’s signal corresponds to the presence and absence of the carrier sinewave. Thus, the transmitted signal can be written as:

$$x(t) = s(t) \cdot e^{j2\pi f_c t}, \quad (6.1)$$

where $s(t)$ is a binary square-wave baseband signal toggling between 0 and 1,

and f_c is the carrier frequency. The received wireless signal can be written as:

$$y(t) = h \cdot s(t) \cdot e^{j2\pi \cdot f_c \cdot t}, \quad (6.2)$$

where h is the complex channel coefficient. The receiver down-converts the signal to baseband by multiplying it with its own carrier frequency. The received baseband signal $r(t)$ then becomes:

$$r(t) = h \cdot s(t) \cdot e^{j2\pi \cdot \Delta f \cdot t} \quad (6.3)$$

$$= h \cdot (0.5 + s'(t)) \cdot e^{j2\pi \cdot \Delta f \cdot t}, \quad (6.4)$$

where Δf is the carrier frequency offset between the transmitter and the receiver, and $s'(t)$ is the same square-wave as $s(t)$ except that it toggles between -0.5 and 0.5 and has zero mean. The frequency representation of the received signal $r(t)$ can be written as:

$$R(f) = \frac{h}{2} \cdot \delta(f - \Delta f) + h \cdot S'(f - \Delta f) \quad (6.5)$$

where $S'(f)$ is the frequency representation of $s'(t)$ and $\delta(f)$ is the unit impulse function. As it can be seen from the equation, this signal has a peak at the carrier frequency offset, Δf . Further, since $s'(t)$ has a zero mean, $S'(0) = 0$.⁵ Thus, the complex value of the peak represents the channel from transmitter to receiver *i.e.*, $R(\Delta f) = \frac{h}{2}$.

6.3 Caraoke Overview

Caraoke is a networked system that enables query-response communication between a Caraoke reader and the e-toll transponders in its range. At the heart of Caraoke is a new device that we call the Caraoke reader; it counts, localizes, and decodes

⁵ $s'(t)$ has a zero mean because $s(t)$ is an on-off keying signal with Manchester encoding, and $s'(t)$ is the same as $s(t)$ but shifted by -0.5.

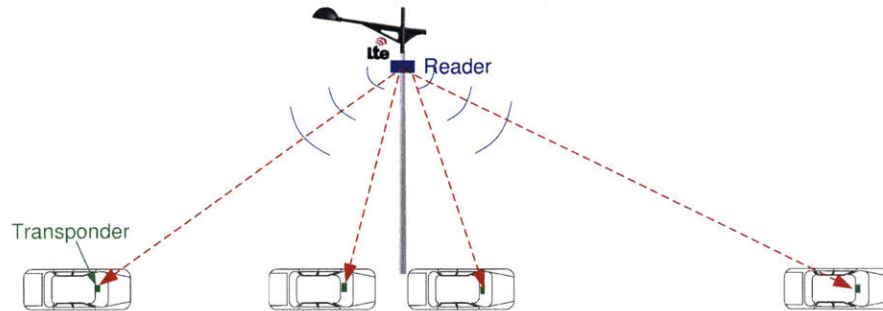


Figure 6-2: **Illustration of Caraoke** The Caraoke reader is mounted on a street-lamp. It queries nearby transponders and uses their signals to localize them and estimate their speeds.

transponders' ids from their signal collisions. It also estimates the speeds of the cars carrying the transponders. The Caraoke reader harnesses its power from solar energy and has an LTE modem to connect to the Internet. Hence, it can be easily deployed without the need for additional infrastructure. As shown in Fig. 6-2, smart cities can deploy Caraoke readers on street-lamps to support a variety of smart services including: 1) traffic monitoring, 2) speed enforcement, 3) red-light running, 4) smart street-parking, and even 4) allowing a user who forgets where he parked to query the system to locate his parked car.

This chapter is focused on the design and implementation of the Caraoke and a small-scale evaluation of the deployment of multiple Caraoke readers on a campus street. Before delving into the details of our design, we note the following two points regarding scope:

- Our objective is to automate smart services, eliminating the personnel cost, and improving the overall accuracy in comparison to the status quo. Note that the current alternatives suffer from significant errors. For example, about 10% to 30% of the speeding tickets based on traffic radars are estimated to be incorrect [21]. The errors are mostly due to the fact that radars cannot associate a speed with a particular car. This task is left to the police officer and hence is prone to human mistakes [21]. Similarly, errors in estimating the number of cars using traffic cameras vary between a few percent to 26%, depending on illumination, wind, occlusions, etc. [135]. Furthermore, the camera lenses have

to be manually cleaned every 6 weeks to 6 months [63].

- For a city to use Caraoke to deliver the above services, it needs to connect the system with its own transportation and traffic databases. For example, in order to detect a car that runs a red light, the city needs to combine the output of Caraoke with the timing of the red-light at the corresponding intersection.

6.4 Counting Despite Collisions

Estimating the number of vehicles at major intersections is critical for traffic management and city planning. In this section, we describe how a Caraoke reader counts the transponders in its radio range. Specifically, when a Caraoke reader transmits a query message, transponders in its radio range respond simultaneously with their information. We would like to use the resulting collisions to count the number of transponders.

At a high level, our approach is simple. We exploit the fact that two transponders, typically, do not have the same carrier frequency, and that their carrier frequency offset (CFO) is relatively large. In particular, the specifications of the E-ZPass transponder show that the device's CFO can exceed one MHz [121]. CFO is typically a nuisance for wireless communication systems which have to compensate for CFO before decoding. In Caraoke however, we leverage CFO for our advantage to count the number of colliding transponders. Specifically, we take the FFT of the collision signal. Since different transponders have different carrier frequencies, the Fourier transform shows multiple peaks at different frequencies that corresponds to the various transponders' CFOs. Fig. 6-3 shows the Fourier transform of a collision signal where five e-toll transponders transmitted at the same time. As can be seen in the figure, there are five peaks, each corresponds to one of five colliding transponders.

This shows that one way for counting the transponders would be to take an FFT of the collision signal and count the peaks in the Fourier domain. To understand the performance of this estimator, we need to tie it to the resolution of the FFT and whether it can distinguish the differences between the CFOs of the transponders. The

resolution of the FFT, δf refers to the width of each FFT bin and can be written as:

$$\delta f = \frac{1}{T} \quad (6.6)$$

where T is the FFT time window. Since the length of the transponder's response is $512\mu\text{s}$, the maximum FFT window is $T = 512\mu\text{s}$ and hence the resolution of the FFT is $\delta f = 1.95 \text{ kHz}$. Thus, if two transponders have carrier frequencies that differ by less than 1.95 kHz, their peaks will fall into the same FFT bin and will be counted as one. Given that the CFO range is 1.2 MHz, the peak of a transponder can fall in any of $N = 1.2 \text{ MHz}/1.95 \text{ kHz} = 615$ FFT bins. If m transponders collide, then the probability of not missing any transponder by counting FFT peaks is:

$$P(\text{not missing any transponder}) = \frac{\binom{N}{m} \cdot m!}{N^m} \quad (6.7)$$

Unfortunately this probability decreases quickly as more transponders are in range. The probability of not missing any transponder is 98%, 93% and 73% for $m = 5$, 10 and 20 cars, respectively. The above derivations shows that an estimator that simply counts the number of peaks in the FFT is acceptable at low car density but can easily miss some cars when the number of cars in range is large.

So, how can we improve the quality of our estimate in scenarios of high densities? To overcome this problem, Caraoke distinguishes whether one or more transponders have fallen into the same FFT bin while counting the number of peaks. It does this by leveraging the *phase rotation* property of the Fourier transform, which says that a shift in the time domain translates into phase rotation in the frequency domain:

$$\begin{aligned} \mathcal{F}\{r(t)\} &= R(f) \\ \mathcal{F}\{r(t + \tau)\} &= R(f) \cdot e^{j2\pi f\tau} \end{aligned} \quad (6.8)$$

where $r(t)$ is the signal in time domain and $R(f)$ is its frequency representation. Specifically, if the FFT peak contains a single transponder's response, then performing the FFT on the same signal with a time shift τ causes only a phase rotation of

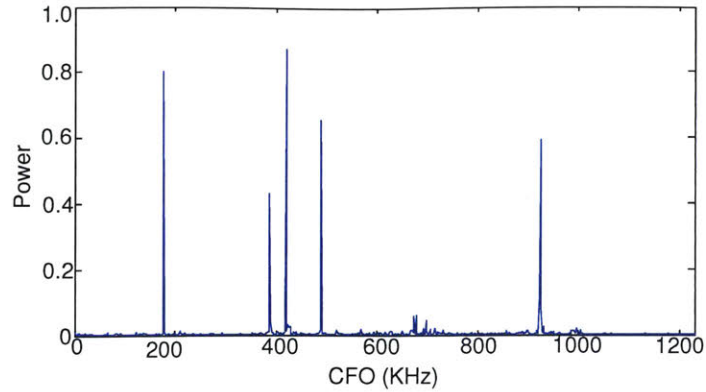


Figure 6-3: **Transponders Response in Frequency Domain.** Fourier transform of a collision signal where five e-toll transponders transmitted at the same time. The number of peaks indicates the number of transponders in the area.

the peak value but the magnitude of the peak does not change. In other words, $\|R(f)\| = \|R(f) \cdot e^{j2\pi \cdot f \cdot \tau}\|$, where $R(f)$ is the frequency representation of the received signal. In contrast, say the CFOs of two transponders, f and f' , fall into the same FFT bin, then the value of the peak in that bin without a time-shift is $R(f) + R(f')$ while its value with a time-shift of τ is $R(f) \cdot e^{j2\pi \cdot f \cdot \tau} + R(f') \cdot e^{j2\pi \cdot f' \cdot \tau}$. Since the frequencies are slightly different, they rotate by different phases and results in a change in the magnitude of the peak.

The above provides us with a mechanism to determine whether an FFT bin has one or more transponders. To do so, we compare the magnitude of the FFT bin with and without a time-shift. If the two magnitudes are different by more than a noise threshold, then multiple transponders have fallen into that bin. In the following, we explain how this detection significantly improves the probability of getting a correct count.

Probability of getting the correct count: As explained before, Caraoke counts the number of the peaks in the FFT to determine the number of cars in range. However, there is a possibility that two cars have fallen into the same bin. Hence, Caraoke considers the peaks with two or more transponders' signals as two cars when it is counting the peak. Specifically, if an FFT peak includes a single frequency, Caraoke counts it as one car and if it has two or more frequencies, it counts it as two cars. Hence, the result of counting will be incorrect only when there is at least a bin

which includes three or more cars. In another word, the probability of not missing any transponder is equal to one minus the probability of having at least one FFT bin which includes three or more transponders' signal. For m colliding transponders and N FFT bins in the 1.2 MHz range, this probability becomes:

$$\begin{aligned}
& P(\text{not missing any transponder}) \\
&= 1 - P(\exists \text{ bin with } \geq 3 \text{ transponders}) \\
&\geq 1 - \sum_{i \in \{1, \dots, N\}} P(\text{bin } i \text{ with } \geq 3 \text{ transponders}) \\
&\geq 1 - \binom{N}{1} \binom{m}{3} \frac{N^{(m-3)}}{N^m}
\end{aligned} \tag{6.9}$$

Substituting $N = 615$, the probability of not missing any transponder is at least 99.9%, 99.9% and 99.7% for $m = 5, 10$ and 20 . Thus, having the capability to detect the peaks that have two or more transponder's signals significantly improves the probability of correctly counting the cars.

Finally, note that for simplicity, our analysis has assumed a uniform distribution for CFO. However, we have also experimentally validated our solution for empirical CFO measurements collected from 155 different transponders.⁶ Our empirical results show that the probability of not missing any transponder is 99.9%, 99.5% and 95.3% for $m = 5, 10$ and 20 which are slightly worse than analytical-results.

6.5 Localizing E-Toll Transponders

Car localization is an essential function for multiple smart services such as smart parking and detecting red-light runners. The first step in localizing cars is to localize the transponder located on cars' windshield. To do so, Caraoke first calculates the angle at which the signal from the transponder arrives at the Caraoke reader. This angle indicates the spatial angle between the transponder and reader, and typically referred to as the angle-of-arrival (AoA).

⁶The mean and standard deviation of their carrier frequencies are 914.84 MHz and 0.21 MHz, respectively.

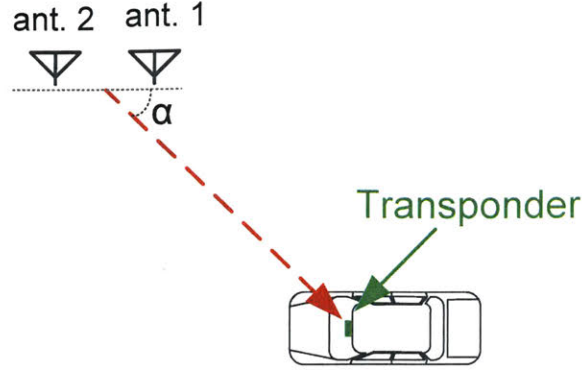


Figure 6-4: **Transponder Localization.** Caraoke calculates the spatial angle (α) of transponder by measuring the phase difference in signal received at two antennas.

In order to measure the AoA, a Caraoke reader uses two antennas separated by a distance d . It is widely-known that the angle of arrival can be computed as [191]:

$$\cos \alpha = \frac{\Delta\phi \lambda}{2\pi d}, \quad (6.10)$$

where α is the spatial angle between the transponder and the reader, shown in Fig. 6-4, $\Delta\phi$ is the phase difference between the two antennas (*i.e.* $\Delta\phi = \phi_2 - \phi_1$), and λ is the carrier wavelength ⁷.

To compute α , we need to substitute the value of the other parameters in Eq. 6.10. While d and λ are known, $\Delta\phi$ should be measured. In the absence of collisions, $\Delta\phi$ can be measured directly between the signals received by the two antennas on the reader:

$$\Delta\phi = \angle \frac{r_2(t)}{r_1(t)} = \angle \frac{h_2 \cdot s(t) \cdot e^{j2\pi\Delta f \cdot t}}{h_1 \cdot s(t) \cdot e^{j2\pi\Delta f \cdot t}} = \angle \frac{h_2}{h_1},$$

where $r_1(t)$ and $r_2(t)$ are baseband signals received by the first and second antenna, respectively, h_1 and h_2 are the channels to the two antennas, $s(t)$ is the transmitted baseband signal and Δf is the CFO between the transmitter and receiver⁸. However,

⁷Note that since Caraoke reader is placed outdoor on a high pole like a street lamp, it has a strong line-of-sight path to the transponder and the multipath effects which occur in standard indoor environments are significantly weaker. Hence, Caraoke does not require a large antenna array to estimate the angle of arrival. In §6.11, we empirically show that Caraoke does not suffer from multipath effects.

⁸Note that the received signals on the two antennas experience the same CFO since the antennas are connected to the same oscillator on the Caraoke reader.

since in Caraoke reader each antenna receives a collision from multiple transponders, such a direct measurement does not work. This is due to the fact that the received signal is the summation of responses from multiple transponders while each has its own channel. Thus, the received signal at each antenna when m transponders respond can be written as:

$$\begin{aligned} r_1(t) &= r_{11}(t) + r_{12}(t) + \cdots + r_{1m}(t) \\ r_2(t) &= r_{21}(t) + r_{22}(t) + \cdots + r_{2m}(t), \end{aligned} \tag{6.11}$$

where $r_{1i}(t)$ and $r_{2i}(t)$ are the received signal from the i_{th} transponder to the first and second antenna, respectively. As can be seen from the equations, one can not directly compute $\Delta\phi$ for the i_{th} transponder using $\angle \frac{r_2(t)}{r_1(t)}$ since $\angle \frac{r_2(t)}{r_1(t)} \neq \frac{h_2}{h_1}$ for a specific transponder.

Fortunately, however, we can use the same trick we used for counting the transponders. Specifically, we first take the FFT of the collision at each antenna and identify the peaks, where each peak corresponds to the response from one transponder. For each peak in the first antenna's signal, the phase value is compared to the phase value of the same peak in the other antenna's signal. These phase differences are used to calculate the spatial angle α for each transponder. Mathematically, the above approach works because the Fourier transform is linear, i.e.:

$$\mathcal{F}\{ax(t) + by(t)\} = aX(f) + bY(f) \tag{6.12}$$

Using the above property, the frequency representation of the received signal at the antennas when m transponders respond can be written as:

$$\begin{aligned} R_1(f) &= R_{11}(f) + R_{12}(f) + \cdots + R_{1m}(f) \\ R_2(f) &= R_{21}(f) + R_{22}(f) + \cdots + R_{2m}(f) \end{aligned} \tag{6.13}$$

where $R_{1i}(f)$ and $R_{2i}(f)$ are frequency representation of the received signals from the i_{th} transponder to the first and second antenna, respectively. As it was explained in §6.2, the received signal from each transponder has a peak at its CFO where the value of the peak represents the channel coefficient (i.e. $R(\Delta f_i) = \frac{h}{2}$). Hence, $R_1(f)$ and

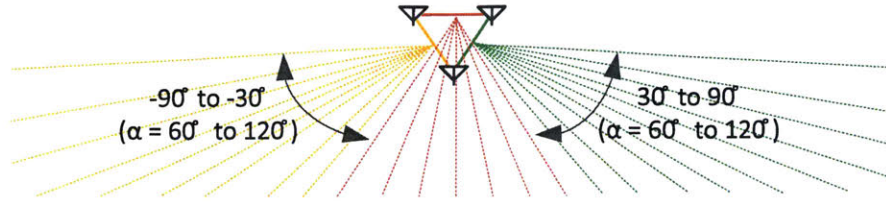


Figure 6-5: **Antenna Re-configuration.** Caraoke reader has three antennas arranged in an equilateral triangle to achieve higher accuracy in calculating spatial angle of transponder. In this case the spatial angle, α , is always between 60 to 120 degrees regardless of transponder location

$R_2(f)$ signals have multiple peaks where each peak corresponds to the response from only one transponder. Therefore, the $\Delta\phi$ for the transponder i can be calculated as follow:

$$\Delta\phi = \angle \frac{R_2(\Delta f_i)}{R_1(\Delta f_i)} = \angle \frac{h_{2i}}{h_{1i}}$$

where Δf_i is the CFO of transponder i , and $R_2(f)$ and $R_1(f)$ are frequency representation of the received signal at the the first and second antennas. Substituting the measured $\Delta\phi$ in Eq. 6.10, we can compute the spatial angle between the transponder and reader (*i.e.* AoA).

The above equation allows us to compute the spatial angle from the reader to the transponder. We can however improve the accuracy of our angle estimate with a smart choice of antenna position. Specifically, the accuracy in calculating α is best for angles around 90° and degrades for angles around 0° or 180° . This is due to the fact that $\Delta\phi$ is proportional to $\cos \alpha$, as shown in Eq. 6.10. Hence, for values close to 0 or 180, α is very sensitive to change in $\Delta\phi$. To reduce this sensitivity, we use three antennas arranged in an equilateral triangle as shown in Fig. 6-5. At any time, we use a pair of antennas out of the three antennas. We pick the pair using a programmable switch. In this setup, for any transponder position, there exists exactly one pair of antennas for which the spatial angle is always close to 90° (*i.e.*, between 60° and 120°). We compute the angle for all pairs and use the pair whose angle is close to 90° degree to localize the car.

Next we use the spatial angle to locate the transponder. The spatial angle does not correspond to a single point in the space. In fact, as shown in Fig. 6-6, this angle

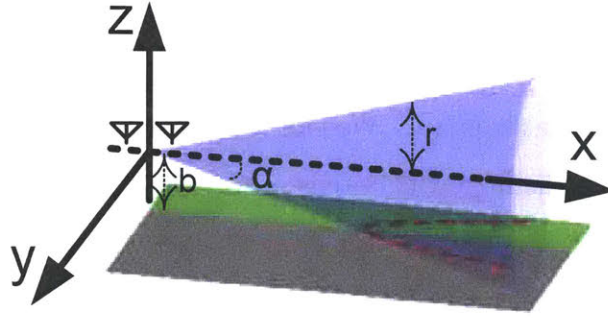


Figure 6-6: **Localizing the Car from AoA:** In this figure α is angle of arrival and b is the height of the pole. The car is located on a point on the intersection of the cone and the road plane which is a hyperbola. Hence, By using the information from two poles, one can localize the exact location of the car.

corresponds to all points on the surface of a cone where its altitude axis is parallel to the road. The cone equation can be written as follow:

$$y^2 + z^2 = r^2 = (\tan(\alpha) \cdot x)^2 \quad (6.14)$$

where x, y and z are coordinate of the car with respect to the center of measuring antennas, and α is the spatial angle between the car and reader. Cars, however, are always on the road and we can use this as another constraint. The intersection of the cone and the road plane is a hyperbola, as shown in Fig. 6-6. The equation for this hyperbola is as follow:

$$(\tan(\alpha) \cdot x)^2 - y^2 = b^2 \quad (6.15)$$

where b is a constant that corresponds to the height of the pole. While a single hyperbola is not enough to localize, we can combine information across two readers to locate the car. Specifically, using a second reader located on the other side of the road provides us another hyperbola equation. Then, by solving these two equations, one can find x and y , and localize the car.⁹

Note, in the case where antennas are tilted by 60° , the process of localizing is the same, except that the cone is tilted by 60° . Hence, the intersection of the cone and road plane is an ellipse instead of a hyperbola. One can simply replace the hyperbola

⁹ The intersection of two hyperbolas may results in more than one point, however, only one of these points is located on the road and the rests are on the sidewalk.

equation with that of an ellipse to find the intersection point.

Finally, in order to intersect location information across two readers, we need this information to be synchronized. We can leverage the readers' connection to the Internet to synchronize them to within tens of ms using the network timing protocol (NTP) [17, 138]. This synchronization level is more than sufficient for localizing parked cars. For moving cars, this introduces some error, which we will discuss in the following section.

6.6 Detecting a Car's Speed

As described earlier, Caraoke can also detect the speed of the car. Specifically, Caraoke readers can be deployed on street-lamps and detect speeding on streets in the city. The car speed can be estimated by localizing the car at two different locations and computing the total time the car took to travel between these two locations. Hence, the speed of the car can be written as:

$$v = \frac{x_2 - x_1}{delay}$$

where x_1 and x_2 are first and second locations of the car and *delay* is the amount of time it took to travel from location x_1 to location x_2 which are computed as described §6.5.

The accuracy of localizing the x_1 and x_2 and estimating the *delay* depends on the time synchronization between the readers. The error in x_1 and x_2 can be upper bounded using the hyperbola equation in the previous section independent of time synchronization. This error depends on reader's height and the number of lanes in the same direction on the street. For example, for a four lane street *i.e.* two lanes in each direction, where the antennas are attached to a street light pole whose height is 13 feet, the maximum error is 8.5 feet.¹⁰ The error in *delay* is the same as the error in timing synchronization. Since the readers are connected to the Internet via LTE

¹⁰The exact equation of error is $\frac{\sqrt{b^2} - \sqrt{b^2 + (l \cdot w)^2}}{\tan(\alpha)}$ where b is antenna's height, l is the number of lanes in the same direction on the street and w is the width of the lane (typically 12 feet).

modems, they can be synchronized up to tens of ms network timing protocol (NTP) [17, 138].

The accuracy of estimating the speed depends on the accuracy of the above parameters as well as how far x_1 and x_2 are from each other. The farther they are, the more accurate the estimate is. In particular, if x_1 and x_2 are measured at readers that are separated by 4 light poles (*i.e.* a separation of about 360 feet (≈ 110 m) [27]), for car speeds of 20 mile/hour and 50 mile/hour, the maximum error is 5.5% and 6.8% respectively. This accuracy can further be improved by taking more measurements along the street from more light poles.

6.7 Decoding Transponders' Ids

In this section, we explain how Caraoke decodes an individual transponder in the presence of collisions of multiple transponders.

At first glance, it might seem that one can decode a transponder's signal by using a band-pass filter centered around the transponder's CFO peak. This solution however does not work because OOK has a relatively wide spectrum –*i.e.*, the data is spread as opposed to being concentrated around the peak¹¹.

In contrast, our decoding algorithm is based on combining multiple collisions in a manner that ensures that the signal from the target transponder combines coherently, whereas the signals from other transponders combine incoherently. This allows Caraoke to boost the SNR of the target transponder above others, and hence decode the target transponder.

Specifically, when a Caraoke reader transmits the query signal, multiple transponders respond simultaneously. Without loss of generality, let us assume that we are interested in decoding transponder 1.

$$r(t) = h_1 s_1(t) \cdot e^{j2\pi \cdot \Delta f_1 \cdot t} + \sum_i h_i s_i(t) \cdot e^{j2\pi \cdot \Delta f_i \cdot t}$$

¹¹Intuitively this can be seen by recalling that OOK randomly toggles between 0 and 1 and hence its spectrum resembles white noise.

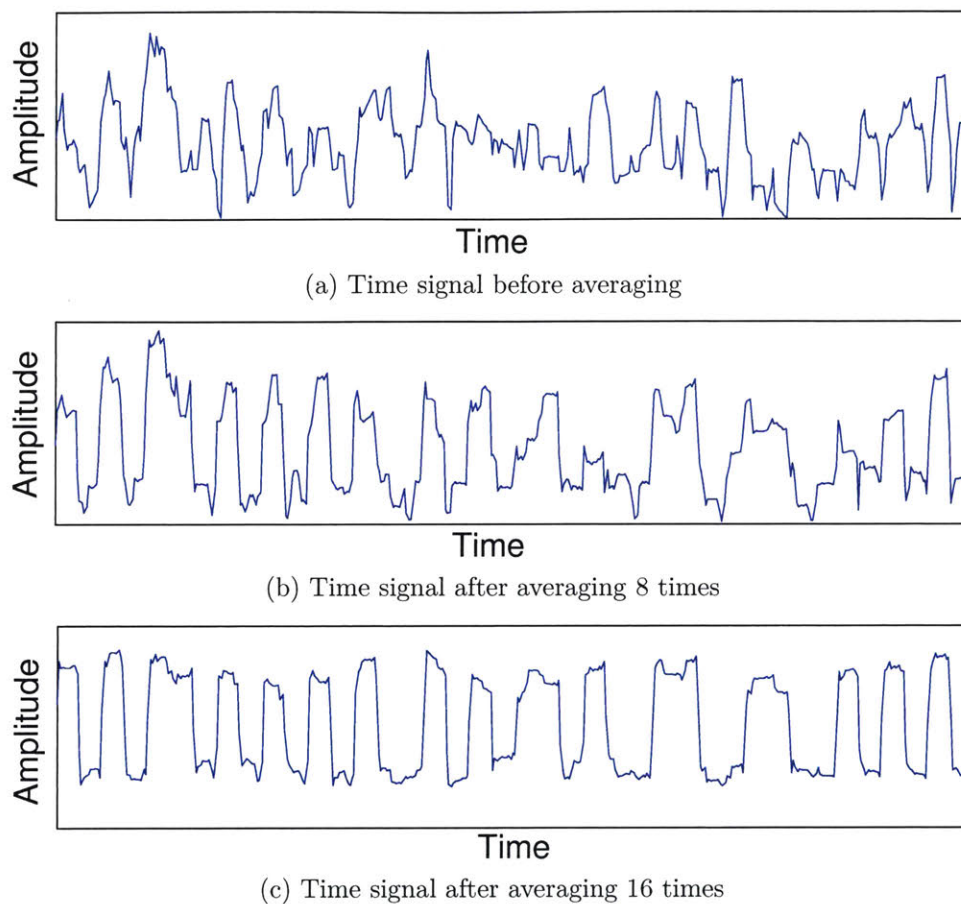


Figure 6-7: **Decoding a Transponder's Response:** (a) Received signal before averaging, (b) after averaging 8 replies, (c) after averaging 16 replies. Caraoke is able to decode a transponder's response by correcting its channel and then averaging.

where $s_i(t)$ is the signal transmitted by the i th transponder, Δf_i is its CFO, and the h_i is its channel to the reader. If the reader transmits another query, the received signal will be:

$$r'(t) = h'_1 s_1(t) \cdot e^{j2\pi \cdot \Delta f_1 \cdot t} + \sum_i h'_i s_i(t) \cdot e^{j2\pi \cdot \Delta f_i \cdot t}$$

Note that the channel coefficients have changed from the first received signal to the second one. This is due to the fact that the transponders start with a random initial phase. The channels h_1 and h'_1 as well as the CFO Δf_1 can be estimated from the peak in the frequency domain as described in §6.2. We can then compensate for the CFO and the channels of transponder 1 and sum up the received signals to obtain the averaged signal $\tilde{s}_1(t)$:

$$\begin{aligned} \tilde{s}_1(t) &= \frac{r(t)}{h_1} \cdot e^{-j2\pi \Delta f_1 t} + \frac{r'(t)}{h'_1} \cdot e^{-j2\pi \Delta f_1 t} \\ &= 2 \cdot s_1(t) + \sum_i \left(\frac{h_i}{h_1} + \frac{h'_i}{h'_1} \right) s_i(t) \cdot e^{j2\pi(\Delta f_i - \Delta f_1)t} \end{aligned}$$

By repeating this process N times we get:

$$\tilde{s}_1(t) = N \cdot s_1(t) + \sum_i \left(\sum_j \frac{h_{ij}}{h_{1j}} \right) s_i(t) \cdot e^{j2\pi(\Delta f_i - \Delta f_1)t}$$

where h_{ij} is the channel from the i -th transponder in the j -th received signal. As can be seen from the above equation, the signals from transponder 1 add coherently, while the other signals add incoherently with random phases and average out. For sufficiently large N , the signal power for transponder 1 will be much more than that for other transponders. In this case, the SNR is enough to be decoded.

Fig. 6-7 shows an example of this decoding algorithm in which the reader receives a collision of the signals from five transponders. Fig. 6-7(a) shows the time signal $r(t)$ of the received collisions before any averaging. As can be seen, the signal looks random and undecodable. Fig. 6-7(b) and (c) show the time signal $\tilde{s}_1(t)$ after averaging 8 and 16 replies respectively. The figures show that after averaging 16 times, the bits of the desired transponder become decodable and the more we average, the better our

ability to decode becomes.

6.8 Caraoke Multiple Reader Protocol

So far, we have assumed that transponders respond to a single reader at any time. However, a transponder on the road might be in the range of two or more Caraoke readers.¹² Therefore, there is a need for a MAC protocol on the readers side to avoid interference from readers. We will start by distinguishing between two interference scenarios:

1. Collision of Reader Queries: In this case, a query signal from a reader collides with a query signal from another reader. As explained in §6.2, the query signal transmitted by a reader is simply a sinewave transmitted at carrier frequency. Even if two readers interfere, the combined signal is still a sinewave at the carrier frequency, and hence a valid trigger. Thus, a collision of two queries is not harmful. Our empirical experiments confirm that transponders are still triggered to respond even when queries from different readers collide.

2. Collision of Reader Query with a Transponder Response: In this case, a query signal from a reader collides with the response of a transponder queried by another reader. This collision is harmful and needs to be avoided. To do so, Caraoke uses carrier sense. Specifically, each reader listens to the medium before transmitting a query. If the medium is available, it then transmits its query. *But how long should the reader listen to avoid a collision with a transponder response.* Recall from Fig §6-1 in §6.2, that the query signal is only $20\mu s$ and the delay between the query and the transponder response is $100\mu s$. Thus, by listening for more than $120\mu s$, if the reader does not hear any signal it can guarantee that no transponder response will be transmitted after the $120\mu s$ and it can transmit its query.

To summarize, Caraoke uses a MAC protocol for the readers based on CSMA, where each reader listens for an idle medium for $120\mu s$ before it can transmit. The main difference, however, is that there is no need for contention window since collisions

¹²The range of a Caraoke reader is 100 feet.

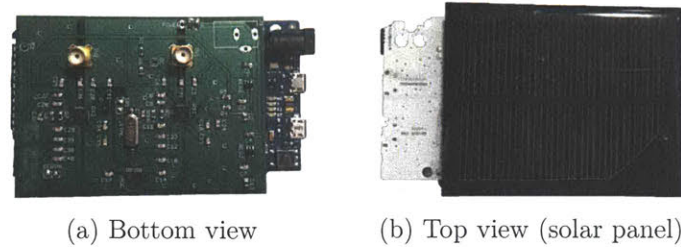


Figure 6-8: **Caraoke Reader.** The board is 3×4 in, and operates by harvesting solar power.

between queries are acceptable.

6.9 Caraoke Reader's Hardware Design

Caraoke is a software-hardware solution. In contrast to the previous sections, which focus on the algorithmic techniques underlying Caraoke, here we describe the hardware design and the optimizations we performed in order to support a low-power low-cost device.

We have developed a custom-design PCB for the Caraoke reader, as shown in Fig. 6-8. The device harvests its power from solar energy and connects to the Internet via a wireless modem. Hence, it can be attached to a light pole (or other structures) without the need for external power or wired Internet connectivity.

Fig. 6-9 illustrates the block diagram of the Caraoke reader. The device has five main block: a query generator, a receiver, a micro-controller, a modem, and a power management unit. The *query-generator* transmits a query signal that invokes a response from nearby transponders, the *receiver* receives the response signal, digitizes it, and provides it to the micro-controller. The *micro-controller* implements the standard receiver processing, i.e., packet detection, phase estimation, etc. It also implements our algorithmic solutions for counting, localization, speed measurement, etc. The processed data is then uploaded to the Internet via an LTE modem. (An alternative approach could use a WiFi modem, and have the readers forming a mesh network to connect to the Internet.) Finally, the *power management* unit includes a small solar panel that attaches on top of the PCB, and a rechargeable battery that

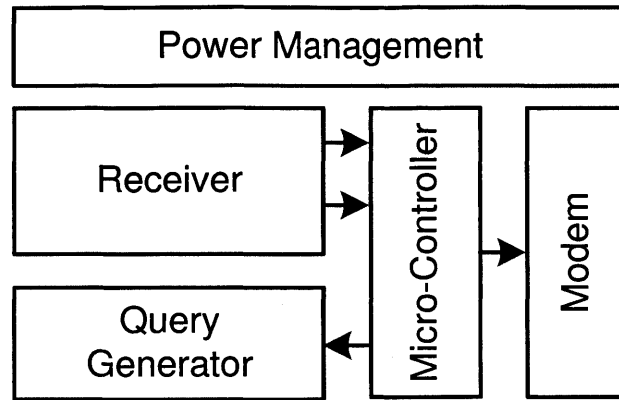


Figure 6-9: Block Diagram of the Caraoke Reader

stores the harvested energy for operation during night time and on cloudy days. Solar panels can harvest $\sim 10mW/cm^2$ on sunny days [173, 184]. We use a 6cmx7.5cm solar panel, which can provides 500mW [146]. In §6.11, we show that this number is several folds larger than the power consumption of a Caraoke reader.

Finally, the device is equipped with a USB port which is used to program the micro-controller. The same USB port can also be used to power the device and extract the data, during the development phase.

In order to minimize the cost and power consumption of Caraoke reader, we perform multiple optimizations as follows:

- **Eliminating the TX chain:** A Caraoke reader has to transmit a query signal to request a response from the transponders. One approach to generate this query is to use a standard transmitter (TX) chain, which generates the signal in the digital domain. This approach however is inefficient in terms of both power and cost. Specifically, a full TX chain includes a mixer, a frequency synthesizer (PLL), a digital-to-analog-converter (DAC) and some digital logic. These components are necessary to transmit data bits. Yet, the e-toll query signal is just a simple sine-wave and does not carry any data. Such a sine-wave can be generated using a simple frequency synthesizer (i.e., a PLL), and does not need the rest of the components in the standard TX chain. The frequency synthesizer is turned on/off by the micro-controller so that the resulting sine-wave has the proper duration of the e-toll query.

- **Replacing FFT with the Sparse FFT:** As mentioned in §6.5, Caraoke operates on the response signal in the frequency domain after taking an FFT. At any time, the number of transponders that respond to the query is typically much smaller than the FFT size. In fact in most cases only a handful of transponders respond. Remember that each transponder creates a peak at the output of the FFT. Thus, the output of Fourier transform is sparse. For such sparse signals, one can compute the Fourier transform in sub-linear time [108, 109]. To do so, we use the sFFT algorithm in [41, 110], which is both simple and has low computation complexity. This directly translates into reduction in power consumption and speedup in the computation.
- **Duty Cycling:** Caraoke reader has two operating modes: active mode and sleep mode. In the active mode, the reader transmits queries, receives responses, and processes the signals. In the sleep mode, the query generator and receiver are turned off by the micro-controller by shutting off their power supply. All peripheral blocks are also deactivated. However, the master clock continues to run. A timer (typically called sleep timer) uses this clock to wake the micro-controller up at a specific time. The micro-controller controls the scheduling of the active and sleep modes to satisfy a desired power budget. Since each query takes about 1ms, we expect the average duration of the active mode to last for 10ms, allowing for a maximum of 10 queries. By controlling the duration of the sleep mode, the total power consumption can be dramatically reduced as we show in §6.11.5.

6.10 Evaluation Setup

Implementation: We implemented Caraoke reader on a printed circuit board (PCB) using off-the-shelf components. For the RF front-ends we use Maxim Integrated MAX2117 [132] and Analog Device AD7356 [50], and the Calliope LTE modem [161]. For the micro-controller, we use an Arduino Due board. The power management

circuit includes regulator to regulate the voltage from the solar panel. For the solar panel, we use OSEPP SC10050 [146]. The RF chains share the same clock for accurate synchronization. The antennas are omni-directional and separated by $\frac{\lambda}{2} = 6.5$ inches. The ADC resolution is 12 bits. Also, the ADCs have differential inputs, and hence higher robustness to noise and interference. The total cost of the components is less than \$40, making the device amenable to large scale deployment. Further, this cost can be significantly reduced in mass production.

We note that for our setup, we extract the data using the USB port as this does not require an LTE subscription and has no impact on counting, localization, or speed estimation.

Experimental Setup: We conducted our experiments on four campus streets, A, B, C, and D, which are shown in Figs. 6-10(a) and (b). All streets have 2-way traffic. Streets A, B, and D have street parking one or both sides of the road. Street C is the busiest street on campus, and is a major street in our city. In all of our experiments, the Caraoke reader was placed on a 12.5-foot pole. We used a total of 4 such poles. The poles are portable and hence allow us to experiment with various configurations, as detailed in §6.11.2.

All experiments were conducted with standard E-ZPass transponders attached to the cars' windshields. We tried to limit our experiments to transponders and cars owned by the authors. However, certain experiments require investigating the distribution of transponder CFOs and its impact on our ability to count vehicles. For that experiment, we collected transponder responses from random cars. We measured only the CFO of the transponders and did not decode the bits. After processing the signals to extract and count the CFOs, we stored only the CFO values with no reference to the ids of the car. We do not believe that the values of the CFOs can be mapped to the owners or used to infer any private information about them. Finally, we note that our transmissions of a sine-wave in the band used by E-ZPass is in accordance with the FCC rules Part 15 [94].

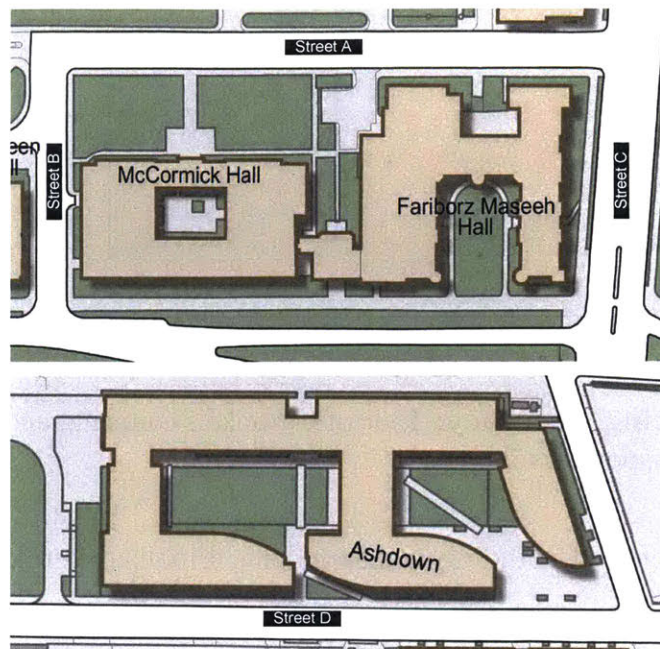


Figure 6-10: **Experimental Setup.** We ran our experiments on four streets in MIT campus, marked as streets A, B and C in the top figure and street D in the bottom figure. All streets have two-way traffic and most have parking spots on one or both sides.

6.11 Empirical Evaluation of Caraoke

We evaluate the various functions of Caraoke using outdoor experiments performed with E-ZPass transponders.

6.11.1 Counting Accuracy

In the first experiment, we aim to evaluate Caraoke’s ability to count transponders based on their CFOs. If we ran this experiment directly on collision signals, we would not know the ground truth. Hence, we needed first to estimate the CFO of each transponder in the absence of collisions. We collected signals from 155 different transponders in one of our campus parking lots. We used a directional antenna to obtain the response of each transponder without collisions. For each transponder, we took the FFT of its signal and noted the FFT bin of its CFO. We then create collisions in post-processing by summing up the time signals from a subset of the transponders. We change the number of transponders in the subset to obtain collisions

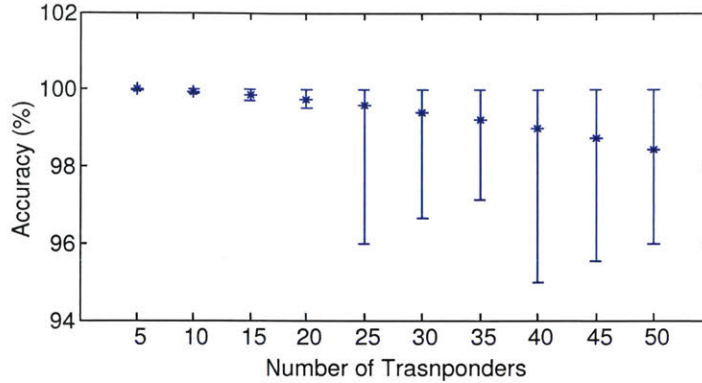


Figure 6-11: **Counting Accuracy:** Plot of Caraoke’s counting accuracy as the number of colliding transponders increases.

with different numbers of colliding transponders. Finally, we take an FFT of each collision signal and estimate the number of colliding transponders using the approach described in §6.4. We have considered collisions of 5, 10, . . . , 50 transponders, and for each case performed 1000 runs.

Fig. 6-11 plots the average accuracy in counting colliding transponders. The figure shows that when the number of transponders in a collision is relatively small, Caraoke accuracy is very close to optimal. In particular, given the empirical CFO values of e-toll transponders, Caraoke can maintain an accuracy higher than 99% when the colliding transponders are fewer than 40. Note that the overall counting accuracy depends on how often the reader would experience 40-transponder collisions vs. 5-transponder collisions, or other numbers. This depends on the intersection and the amount of traffic. Overall, the results show that Caraoke is effective at counting vehicles using e-toll transponder collisions.

In our second experiment, we deployed Caraoke at the intersection of Street A and Street C in Fig. 6-10(a), and used it to track the flow of traffic on the two streets. In this experiment, we do not know the ground truth since we do not know which cars have transponders; despite this limitation the results can indicate the ratio of traffic between the two streets and how it relates to the timing of their green-light and red-light.

Fig. 6-12 plots the number of cars observed by Caraoke as a function of time for both streets. The figure is marked with the green-yellow-red times for the traffic

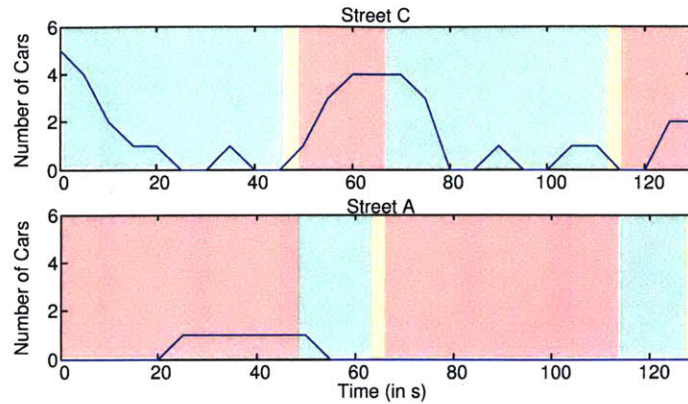


Figure 6-12: **Traffic Monitoring at an Intersection.** The figure shows the number of cars counted by Caraoke at an intersection over two periods of green-red lights.

lights on each of the streets. The figure shows how a backlog of cars accumulates during a red-light and clears during a green light. Further, road C is much busier than road A (10 times on average), but its green light is only 3 times longer than that of road A. This shows that Caraoke can help the city understand the relative congestion on various streets, and potentially adapt the timing of the traffic light to reduce car waiting time.

6.11.2 Localization Accuracy

In this experiment, we focus on localizing cars to parking spots. We ran our experiments on streets A and B, which have parking spots on one or both sides of the road, for a total of 36 spots. We use 4 poles, two deployed on street A and one at the intersection of A and B, and one on street B. We use two cars equipped with E-ZPass transponders, which we move between different parking spots to experiment with different configurations. There are other cars parked on the street, whose transponders collide with our two cars. We ran over 175 localization tests which span 35 configurations for the cars, and 5 runs per configuration which differ by the number and identity of the colliding transponders due to other parked cars and traffic dynamics on the streets. In our processing, we ignore the FFT spikes corresponding to other cars and focus on localizing our transponders using the method described in §6.5. To measure the ground truth we use a Bosch GLM50 laser distance measurement

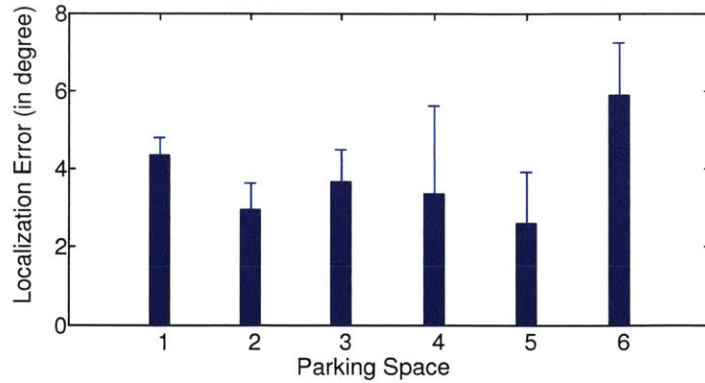


Figure 6-13: **Transponder’s Localization Accuracy.** Figure shows the accuracy of localization for cars parked in different parking spaces where the spot 1 is closest and spot 6 is the furthest spot to Caraoke pole. Caraoke enables enough accuracy to detect occupied versus available spots

tool [64], which can measure distances to within a few millimeters. We then measure the ground truth angle using our knowledge of the transponder’s distance from the pole, the pole’s height, and the transponder’s elevation.¹³

Fig. 6-13 plots the error in the angle of arrival measurement computed by Caraoke. The errors are plotted as a function of the location of the parking spot with respect to the pole carrying the Caraoke reader. The bar graph shows the average and standard deviation of the measurements from all four poles. The figure shows that the average localization error is about 4 degrees. Interestingly the error is the largest at the two ends, i.e., when the car is only 1 spot away or 6 spots away from the pole. This is because the two antennas used for computing the AoA create a 60° angle with the plane of the road. As explained in §6.5, without this tilt the error in AoA for the farthest spot, i.e., spot 6, would be significantly larger than the closer spots. The results show that our decision of positioning the antennas at 60° angle with the street produces a relatively balanced error across spots.

One may be surprised that Caraoke’s AoA accuracy is high despite that it uses a simple two-antenna array. Large antenna arrays are typically needed in multipath

¹³Measuring the ground truth for 3D angles is difficult. However, since the car is parked to the side of the road, the transponder, the pole and the two antennas are approximately in the same plane. Hence, knowing the distances we can measure the ground-truth angle with simple geometry rules. Note that this approximation means that our results are conservative –i.e., Caraoke’s actual errors could be slightly smaller.

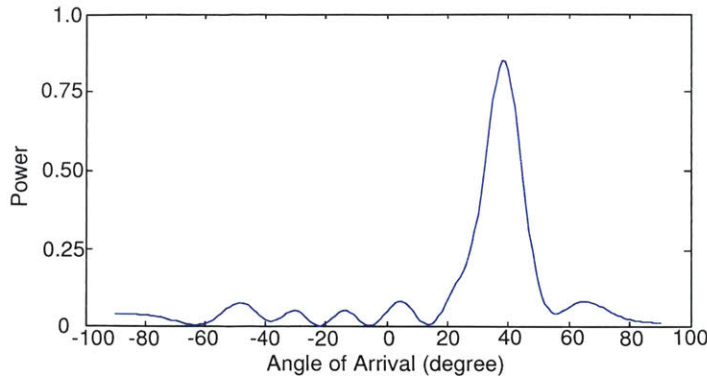


Figure 6-14: **Multipath Profile.** Caraoke readers have strong line-of-sight paths to the transponders. Therefore, multipath effects which occur in standard indoor environments are significantly weaker.

scenarios to separate signals that travel along different paths from source to destination. While multipath effect has been a big challenge for indoor localization, it becomes less prominent in Caraoke’s design, primarily because Caraoke reader is mounted on a several meter high pole in an outdoor environment, and hence, has a prominent line-of-sight path between transmitter and receiver. To ascertain that this setting indeed has low multipath, we augmented Caraoke with an antenna attached to a rotating arm of radius 70cm. Like past work [124], we use this design to emulate a large antenna array (Synthetic Aperture Radar) and obtain the multipath profile of the signal coming from the car’s transponder. As the antenna rotates, we continuously measure the wireless channel of the transponder’s signal and then use the measured channels to reconstruct the multipath profile of the transponder’s signal using standard phased array processing algorithm and the MUSIC algorithm [191].

A representative multipath profile obtained using this setup is shown in Fig. 6-14. As expected, the multipath profile has one dominant peak. To confirm that this is indeed the case across experiments, we repeat the experiment for 100 runs across different times and locations and measure the relative power of the two peaks with the highest amplitude in the multipath profile (i.e., the highest to the second highest peak power). We observe that, on average, the strongest peak has an order of magnitude higher power (specifically, 27 times higher) as compared to the second strongest peak. This confirms our hypothesis, that for line-of-sight outdoor environments, multipath

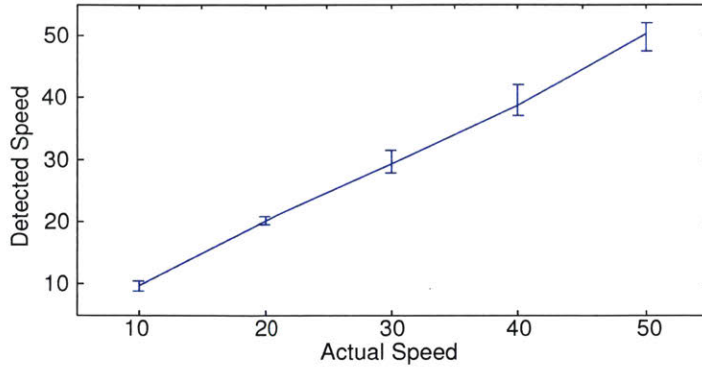


Figure 6-15: **Speed Detection Accuracy.** Figure shows the car speed detected by Caraoke versus its actual speed. Caraoke detected the speed to within 8% over a range of different speeds

effects are significantly weaker than the line of sight peak and hence, do not interfere with accurate phase based localization.

6.11.3 Speed Estimation Accuracy

Next, we evaluate Caraoke’s ability to estimate car speed. For speeds below 40 mile/hour, we ran our experiments on street A and street D in Figs 6-10(a) and(b). We used two poles to localize the car and compute the speed as explained in 6.6. We locate the two poles 200 feet apart. Experiments with speeds higher than 40 mile/hour are performed in an empty lot. We perform 10 experiments at each speed, for a total of 50 experiments. We compare the speed detected by Caraoke with the speed reported by the car. Fig. 6-15 plots the estimated speed versus the actual speed. The plot shows both the average and the 90th percentile. The figure shows that Caraoke’s estimate of the speed is within 8% of the real value –i.e., the error is 1 to 4 mile/hour across the whole range of speeds.

6.11.4 Decoding Accuracy

We evaluate Caraoke’s ability to decode the ids of the transponders in the presence of wireless collisions. We place the Caraoke reader on a pole. The reader receives the colliding responses and decodes the ids of the transponders. Specifically, as described in §6.7, to decode a particular id, the reader combines the collisions after compensat-

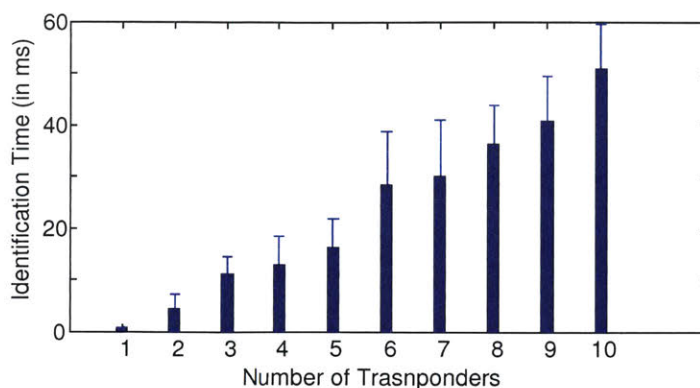


Figure 6-16: **Identification Time versus Number of Transponders.** As the number of transponders in Caraoke reader’s range increases, Caraoke requires longer sensing time to identify the tags.

ing for the channel and CFO of the desired transponder. The reader keeps combining collisions until the decoded id passes the checksum test. We use a maximum of 10 transponders whose owners agreed to the experiment. We run a total of 100 experiments with a different number of colliding transponders and different distances from the reader.

Fig. 6-16 shows the time taken to decode a transponder id for different numbers of colliding transponders. Recall that when decoding an id, Caraoke reader sends multiple queries, and combines the resulting collisions to decode. Since the queries are separated by 1ms, the time axis also shows the number of combined collisions in order to decode. As can be seen, the required time increases as the total number of colliding transponders increases. This is because when decoding a transponder, other transponders in the collision act as noise. However, even when there are 10 colliding transponders, Caraoke can still decode the transponder of interest within 50ms, on average.

Note that 50ms is also the time to decode all 10 transponders since one does not need to collect new collisions for individual transponders. One only needs to compensate for the CFO and channel of each of the transponders differently. Since the processing time of each query is negligible in comparison to the time it takes to transmit and receive, decoding all colliding transponders takes as much total time as decoding one transponder.

6.11.5 Karaoke Reader's Power Consumption

To profile Karaoke reader for power consumption, we removed the solar panel and the battery and used the USB port to power the device. We connected the USB to In-line Voltage and Current Meter [47], which measures the voltage and current drawn by the board. The power measurements reveal that Karaoke reader consumes 900mW in active mode and consumes only $69\mu\text{W}$ in sleep mode.¹⁴ Since the solar panel delivers 500mW in the sun, Karaoke reader would not be able to run continuously in the active mode. However, as explained in §6.9, due to duty cycling, the average power consumption of the board is much lower as the duration of the active mode is less than 10ms. Thus, if Karaoke reader takes one measurement every second, it would consume an average power of 9mW, which is $\sim 56\times$ lower than what it can harvest from its solar panel. Hence, the energy harvested from solar during 3 hours can be stored in a rechargeable battery and run the device for a week regardless of weather condition. Finally, note that this operation time can be further increased by using a larger solar panel or increasing the sleep time.

6.12 Discussion

This chapter introduces a new algorithm and a technique to solve the collision problem in a network of low-power active RFIDs where they have no MAC protocols. Further, we show how we can use our proposed techniques to build a sensor that can count, localize, and measure the speed of cars using the RF signal from their low power e-toll transponders (i.e. active RFIDs). Our sensors are small, low-cost and low-power, and hence can be easily deployed on street lamps to allow cities to deliver smart services, e.g., smart parking, traffic monitoring and speed detection, all using one infrastructure. While we focused mainly on tracking vehicles, once such

¹⁴These numbers exclude the modem module. LTE and Wi-Fi modems consumes 1-2 W and 100s of mW, respectively, while transmitting at Mbps data rates. A Karaoke reader needs to transmit only a few kbits to convey the results of processing one query (i.e., the channels and CFOs) to a backend. Furthermore, it can batch the results of multiple queries together. Hence, it can use the modem for tens of millisecond then put it to sleep for a minute or so. By duty cycling the modem, one can bring down its average power consumption to mW or hundreds of μW .

infrastructure exists, the city may use it for additional services, like locating first responders, tracking the delivery of goods, and enabling people to pay for mobile services.

Chapter 7

Conclusion

This thesis addresses two main sets of constraints in designing IoT devices: bandwidth and energy. Bandwidth constraints originate from a shortage of available radio frequency spectrum. Energy constraints stem from the small form factor and inaccessible placement of these devices, which restrict their battery's capacity and increase the frequency with which batteries must be replaced. Although the number of devices is growing rapidly, the available unlicensed spectrum remains unchanged. Moreover, certain applications, such as VR and multimedia devices, have high-data-rate requirements, which inherently require a large amount of wireless spectrum. This thesis tackles these two challenges through the creation of software-hardware systems in which new protocols and algorithms are jointly designed with new networking hardware. This thesis demonstrates how such an approach results in far more efficient systems, both in terms of energy and spectrum usage. Finally, it shows how these systems enable new applications, such as untethered VR and smart city services. Specifically, this thesis introduces the following systems:

- **Distributed Coherent Transmission:** We present AirShare a primitive that makes distributed coherent transmission seamless, enabling high-gain communication protocols. AirShare transmits a shared clock over the air and feeds it to the wireless nodes as a reference clock, hence eliminating the root cause of incoherent transmissions. In this thesis, we address the challenges in designing

and delivering such a shared clock. We also implemented AirShare in a network of USRP software radios, and demonstrate that it achieves tight phase coherence. Further, to illustrate AirShare's versatility, we use it to deliver a coherent-radio abstraction on top of which we demonstrate two cooperative protocols: distributed MIMO, and distributed rate adaptation.

- **Agile mmWave Networks:** Agile-Link is a new protocol which can find the best beam alignment in mmWave radios without scanning the space. We prove that Agile-Link finds the optimal direction in logarithmic number of measurements. Furthermore, we implemented Agile-Link in a mmWave radio to evaluate it empirically. Our results show that Agile-Link reduces beam alignment delay by orders of magnitude, making high-throughput mmWave communication suitable for mobile applications.
- **Untethered Virtual Reality:** We present MoVR a novel system that allows mmWave links to sustain high-data rates even in the presence of a blockage. MoVR introduces a self-configurable mmWave mirror that detects incoming signals from the transmitter and reconfigures itself to reflect these signals toward the receiver, in the presence of a blockage. We then implemented MoVR to empirically demonstrate its capability in enabling untethered virtual reality.
- **A million-point Fourier transform chip:** We design a chip which performs a 0.75 million-point Fourier transform for sparse signals. We address challenges in designing very-large-scale integration (VLSI) architecture of sparse Fourier transform algorithms. Furthermore, we have built an ASIC (using IBM's 45nm SOI technology), which computes a 0.75 million-point sparse Fourier transform in 6 microseconds (i.e., 88 times faster than a C++ implementation of the algorithm running on a 3.4GHz Quad-Core i7 CPU), while consuming 40 times less energy than prior VLSI implementations of traditional fast Fourier transforms (FFTs).
- **An E-Toll Transponder Network for Smart Cities:** Caraoke introduces

new techniques and algorithms to address interference in low-power active RFIDs which lack MAC protocols. To address the interference problem, Caraoke exploits hardware non-idealities, and looks at the structure of the RFID's signal and its properties in the frequency domain. We show how these techniques can be used to design a networked system capable of delivering smart services using E-toll transponders. Our design operates with existing unmodified transponders, allowing for applications that communicate with, localize, and count transponders, despite wireless collisions. We built and evaluated Caraoke on four streets on our campus and demonstrated its capabilities in delivering smart services, such as smart parking, traffic monitoring and speed detection.

7.1 Lessons Learned

This thesis details contributions that span both software and hardware in the fields of wireless communication and networking. Networking research often tends to focus on layers: the physical layer, which includes circuits and signals; the network layer, which refers to network protocols; and the application layer, which delivers the final service to the user. Traditionally, these layers have been designed in isolation, often without considering the specific application. In contrast, we have designed software and hardware jointly, with a deep understanding of the intended application. This cross-layer design approach led to important research contributions as well as several lessons learned:

- **Circuit Non-idealities: Problems or Opportunities.** Traditionally, wireless networking research has focused on designing algorithms and protocols, and considered the underlying hardware as a given component of the system. Therefore, some circuit properties and non-idealities have resulted in design challenges. To tackle these challenges, protocol designers have accepted the hardware non-idealities as harmful phenomena, and designed algorithms to correct or compensate for them. However, this approach often results in overly complex and in-efficient systems. In contrast, designing software and hardware

jointly allows us to deeply understand the root cause of circuit non-idealities and investigate their effects on the software side. This level of understanding allows for two different approaches in designing more efficient systems: 1) To modify the hardware to eliminate the root cause of these non-idealities. 2) To design protocols and algorithms that exploit these non-idealities by viewing them as features, rather than correcting for them as harmful phenomena.

For instance, prior to AirShare, cooperative PHY protocols had to use complex algorithms to compensate for carrier frequency offset (CFO), an issue which is due to the fact that radios use independent crystals and they naturally have different properties. However, these solutions depend on the details of the radio technology and works only within a particular context. In contrast, AirShare shows that by designing hardware and software jointly, the root cause of CFO can be eliminated in the hardware, enabling a generic coherence abstraction that can be leveraged by various cooperative PHY protocols, and applied broadly across technologies. On the other hand, Caraoke shows another approach where the CFO is exploited in the software side to design more efficient wireless protocols. Specifically, Caraoke leverages CFO as a unique signature of independent radios to resolve collision and separate their signals. In this thesis, we show how such designs enable far more efficient networks.

- **Restructuring the Computation:** As demonstrated in this thesis, another benefit of designing software and hardware jointly is the ability to redistribute computations between software and hardware modules depending on the specific application. This approach leverages certain mathematical properties of wireless channels and signals to relax both hardware and software requirements. For example, prior to our Fourier transform chip, real-time implementations of large size Fourier transforms have been very challenging to perform. This is because software implementations of such FFTs are too slow for real-time applications, while hardware implementations require a large silicon area and consume significant amounts of power. In contrast, we leverage the fact that in

many applications, the signal of interest is sparse; hence, using FFT might not be the most efficient way to compute a Fourier transform in these applications. Specifically, we show how signal sparsity can be leveraged, leading the design of new algorithms and hardware jointly to compute a Fourier transform more efficiently than traditional FFTs.

A similar example can be seen in Agile-Link where we exploit the fact that wireless channels are sparse in mmWave communications. This allows us to design an algorithm to find the best beam alignment faster than traditional approaches, the majority of which ignore sparsity of wireless channels. Furthermore, we show that by restructuring the computation of the algorithm between the software and hardware modules, the end-to-end performance can be improved. For example, a main part of our algorithm is a hashing function which is performed in hardware simply by adjusting the value of phase shifters. Such opportunities are only observed when software and hardware are jointly designed.

7.2 Future Work

The advent of IoT has created major opportunities. We are witnessing the infusion of computing into electrical and mechanical processes. This development will enable driverless cars, new medical devices for connected health, smarter machines, and new modes for human-computer interaction. An important future direction that builds upon the work in this thesis is to continue focusing on developing algorithms and system architectures that allow us to realize these opportunities. This requires analyzing the interfaces between hardware and software modules, and potentially restructuring computation to satisfy system constraints. It also involves re-evaluating security at all levels of the network stack, from the circuit all the way to the user interface. Below are some of the avenues which we can explore.

7.2.1 3D Printing of Sensors:

Imagine if we could 3D print all the components used in a wireless sensor. For example, can we 3D print a battery, an antenna or even a microcontroller? Even further, can we 3D print a digestible sensor that would allow doctors to customize electronic pills for their patients? Recent advances in direct digital manufacturing have enabled 3D printers which can print a wide variety of materials with very high resolution, which may make these things possible. This technology allows us to create 3D-printed component libraries that anyone could contribute to, making it possible for researchers to quickly and easily create prototype devices to match any requirements. By leveraging software-defined features, it may be possible to build sensors with unique identifiable hardware properties that can be used as a fingerprint. A system which can detect these fingerprints would be a very powerful security mechanism, capable of reliably identifying counterfeit nodes.

7.2.2 Wireless Systems for Driverless Cars:

Driverless cars is an active area of research in both academia and industry, with much investment in large-scale projects by major car manufacturers, as well as technology companies, such as Google and Uber. Although driverless cars are already entering production, their safety and reliability remains a challenge. However, wireless systems can deliver powerful solutions. Recent advances in RF sensing have shown the impressive capabilities of devices which use wireless signals to sense and track objects. Consider a system that employ many of these wireless devices to create a collaborative network capable of assembling a 3D picture of the environment. However, unlike typical wireless networks, which use wireless signals only for communication, this network would use them for both communication and sensing. This will necessitate new hardware, protocols and algorithms to enable devices to collaborate at all levels of the network stack, from the hardware layer to the application layer. Overcoming these challenges would allow us to provide driverless cars with critical information about the road and environment, even when traditional sensors like cameras might

fail.

7.2.3 Security for IoT Sensors:

In the near future, most IoT devices will communicate with each other instead of communicating to powerful access points. It is therefore increasingly important to find end-to-end, robust and scalable security solutions for these devices, especially since many of these devices communicate very sensitive information, such as health-related data. Given the tight size and energy constraints of IoT sensors, designing such security solutions is a hard problem. Unfortunately, software-only security will not be sufficient to overcome these constraints, and therefore future IoT technologies require complementary software and hardware security solutions. One powerful solution to guard IoT sensors against attacks might be to use their wireless capabilities for localization. We can extract the location of a sensor from its RF signal at the physical layer, and pass its communications to the higher layers only if the location is valid.

Appendix A

Analysis of Agile-Link System

A.1 Formal Proof of the Algorithm

A.1.1 Notation and Preliminaries

(a) Basic notation

- We use $[N]$ to denote $\{0 \dots N - 1\}$.
- We use S to denote the support of \mathbf{x} .
- We use \mathbf{F} to denote the Fourier transform matrix, and \mathbf{F}' to denote the inverse Fourier transform matrix. Also, we use \mathbf{F}_i to denote the i -th row of \mathbf{F} ; same for \mathbf{F}' . Finally, we use $\hat{\mathbf{x}}$ to denote $\mathbf{F}\mathbf{x}$.
- For two vectors \mathbf{x} and \mathbf{y} , we define the *Hadamard product* \circ of \mathbf{x} and \mathbf{y} as $(\mathbf{x} \circ \mathbf{y})_i = x_i y_i$. We will use this notion to mask out the coefficients outside of a given segment.
- We use $\mathbf{x} * \mathbf{y}$ to denote the convolution of \mathbf{x} and \mathbf{y} .
- The vectors $\mathbf{e}_0 \dots \mathbf{e}_{N-1}$ denote the standard basis. I.e., $(e_p)_i = 1$ for $p = i$ and $(e_p)_i = 0$ otherwise.

(b) Measurements and box car filter

Our measurements can be described using the notion of the *boxcar filter*, defined as follows. For parameter P , let \mathbf{H} be such that $H_i = \frac{\sqrt{N}}{P-1}$ if $|i| < P/2$ and $H_i = 0$ otherwise. It is known that $\hat{H}_j = \frac{\sin(\pi(P-1)j/N)}{(P-1)\sin(\pi j/N)}$.

Proposition A.1.1. $\hat{\mathbf{H}}$ satisfies the following properties: (i) $\hat{H}_0 = 1$; (ii) $\hat{H}_j \in [\frac{1}{2\pi}, 1]$ for $|j| \leq \frac{N}{2P}$; (iii) $|\hat{H}_j| \leq \frac{2}{1+|j|P/N}$ if $P \geq 3$.

Claim A.1.2.

$$\|\hat{\mathbf{H}}\|_2^2 = \sum_j |\hat{H}_j|^2 \leq 1 + 2N/P \sum_j 1/|j|^2 \leq C \frac{N}{P}$$

for some constant C .

We also define a *shifted* version of \mathbf{H} defined as $(H^t)_i = H_{i-t}$. By the time-shift theorem it follows that $|\hat{H}_i^t| = \hat{H}_i$.

Using this notation, we can write each measurement \mathbf{a}^b as

$$\mathbf{a}^b = \sum_{r=0}^{R-1} (\mathbf{F}_{s_r^b} \circ \mathbf{H}^{rN/R}) \omega^{t_r^b}$$

Each segment of \mathbf{a}^b , when multiplied by a row of the matrix \mathbf{F}' , can be interpreted as follows.

Claim A.1.3. $(\mathbf{F}_i \circ \mathbf{H}) \cdot \mathbf{F}'_p = \hat{H}_{i-p}$

Proof.

$$(\mathbf{F}_i \circ \mathbf{H}) \cdot \mathbf{F}'_p = \mathbf{F}_i \cdot (\mathbf{F}'_p \circ \mathbf{H}) = (\widehat{\mathbf{F}'_p \circ \mathbf{H}})_i = (\hat{\mathbf{F}'_p} * \hat{\mathbf{H}})_i = (\mathbf{e}_p * \hat{\mathbf{H}})_i = \hat{H}_{i-p}$$

□

(c) Pseudo-random permutations

We will use matrices $\mathbf{P}_{\rho,b}$ parameterized by mappings ρ of the form $\rho(i) = \sigma^{-1}i + a \pmod{N}$ for $\sigma, a, b \in [N]$ such that

- $(\mathbf{P}_{\rho,b}\mathbf{x})_{\rho(i)} = x_i\omega^{\tau(j)}$ for $\tau(j) = b(j + \sigma a)$
- $\mathbf{F}'\mathbf{P}_{\rho,b} = \mathbf{P}'_{\rho,b}\mathbf{F}$ for $\mathbf{P}'_{\rho,b}$ as defined in chapter 3.

Note that τ is a permutation assuming b is invertible mod N . We use \mathcal{R} to denote the set of all mappings ρ as defined above. For the analysis, we will assume that N is prime. This will ensure that the elements $\rho \in \mathcal{R}$ are permutations. Furthermore, in this case \mathcal{R} is *pairwise independent*, i.e., for any $i \neq j, i' \neq j'$, we have

$$Pr_{\rho \in \mathcal{R}}[\rho(i) = i', \rho(j) = j'] = 1/N^2$$

It will be convenient to assume $\|\mathbf{x}\|_2^2 = 1$. Then we can define the threshold T to be $(\frac{1}{2(2\pi)} - \frac{1}{8\pi})^2 (\frac{1}{2(2\pi)})^2 / K$.

A.1.2 Proofs

Lemma A.1.4. *Fix b and select $\rho \in \mathcal{R}$ uniformly at random. Then, for any s :*

$$E[I(b, \rho(s))] = E[|\mathbf{a}^b \mathbf{F}'_{\rho(s)}|^2] \leq CR/P$$

where C is the constant from Claim A.1.2.

Proof.

$$E[|\mathbf{a}^b \mathbf{F}'_{\rho(s)}|^2] = E\left[\left| \sum_{r=0}^{R-1} ((\mathbf{F}_{s_b^r} \circ \mathbf{H}^{rN/R}) \omega^{t_r^b}) \mathbf{F}'_{\rho(s)} \right|^2 \right] \quad (\text{A.1})$$

$$= E\left[\left| \sum_{r=0}^{R-1} \hat{H}_{s_b^r - \rho(s)}^{rN/R} \omega^{t_r^b} \right|^2 \right] \quad (\text{A.2})$$

$$= \sum_{r=0}^{R-1} E\left[|\hat{H}_{s_b^r - \rho(s)}|^2 \right] \quad (\text{A.3})$$

where in step A.2 we used Claim A.1.3 and the independence of the variables t_r^b , $r = 0 \dots R-1$. Since $i = s_b^r - \rho(s)$ is distributed uniformly at random in $[N]$, by Claim A.1.2:

$$\sum_{r=0}^{R-1} E \left[|\hat{H}_{s_b^r - \rho(s)}|^2 \right] \leq \sum_{r=0}^{R-1} 1/N \sum_i |\hat{H}_i|^2 \leq R/N \cdot N/P \cdot C = CR/P$$

□

Lemma A.1.5. *Suppose that $|s_b^r - i| \leq \frac{N}{2P}$. Then*

$$\Pr[|\mathbf{a}^b \mathbf{F}'_i|^2 \geq \frac{1}{4(2\pi)^2}] \geq 5/6$$

Proof.

$$|\mathbf{a}^b \mathbf{F}'_i|^2 = \left| \sum_{r'=0}^{R-1} \hat{H}_{s_b^{r'} - i}^{r'N/R} \omega^{t_{r'}^b} \right|^2 \quad (\text{A.4})$$

$$= \left| \hat{H}_{s_b^r - i} \omega^{t_r^b} + \sum_{r' \neq r} \hat{H}_{s_b^{r'} - i} \omega^{t_{r'}^b} \right|^2 \quad (\text{A.5})$$

$$= \left| \hat{H}_{s_b^r - i} \omega^{t_r^b} - X \right|^2 \quad (\text{A.6})$$

We know from Proposition A.1.1 that $|\hat{H}_{s_b^r - i} \omega^{t_r^b}| \geq \frac{1}{2\pi}$. We will show that the probability of $X \geq \frac{1}{2(2\pi)}$ is at most 1/6. It will follow that $|\mathbf{a}^b \mathbf{F}'_i|^2 \geq \frac{1}{4(2\pi)^2}$ with the probability of at least 5/6.

Recall that $s_b^0, s_b^1 \dots$ are separated by P . Therefore, for $r' \neq r$, we have $|s_b^{r'} - i| \geq P - |s_b^r - i| \geq P - \frac{N}{2P}$, which is at least $P/2$ for B large enough. By the independence of the variables $t_{r'}^b$ and by Proposition A.1.1 we have:

$$\begin{aligned} E[X^2] &= E\left[\left| \sum_{r' \neq r} \hat{H}_{s_b^{r'} - i} \omega^{t_{r'}^b} \right|^2 \right] \\ &= \sum_{r' \neq r} E\left[\left| \hat{H}_{s_b^{r'} - i} \omega^{t_{r'}^b} \right|^2 \right] \\ &\leq 2 \sum_{d=1}^R \left(\frac{2}{1 + P/N \cdot d(P/2)} \right)^2 \\ &\leq 8/(P^2/N) \sum_{d=1}^R (1/d)^2 \leq 8CN/P^2 \end{aligned}$$

Since $P = N/R$ and $R^2 = N/B$, the latter expression is bounded by $8C/B$, which

is less than $\frac{1}{6(4\pi)^2}$ for B large enough. It follows that the probability of $X^2 \geq \frac{1}{(4\pi)^2} = 6E[X^2]$ is at most $1/6$. \square

Proof of Theorem 4.1, Part (1). Suppose that $s \in S$. Select s_b^r that is closest to $\rho(s)$. Note that $|s_b^r - \rho(s)| \leq \frac{N}{2P}$, which by Lemma A.1.5 implies $|\mathbf{a}^b \mathbf{F}'_{\rho(s)}|^2 \geq \frac{1}{4(2\pi)^2}$ with the probability of at least $5/6$.

We now lower bound $T(s)$ as follows

$$T(s) \geq |\mathbf{a}^b \mathbf{F}'_{\rho, b} \mathbf{x}|^2 |\mathbf{a}^b \mathbf{F}'_{\rho, b} \mathbf{e}_s|^2 \quad (\text{A.7})$$

$$= |Yx_s - X|^2 |Y|^2 \quad (\text{A.8})$$

where $Y = \omega^{\tau(s)} \mathbf{a}^b \mathbf{F}'_{\rho(s)}$ and $X = \sum_{s' \in S - \{s\}} \omega^{\tau(s')} \mathbf{a}^b \mathbf{F}'_{\rho(s')} x_{s'}$.

We can bound $E[|X|^2]$ as follows.

$$E\left[\left|\sum_{s' \in S - \{s\}} \omega^{\tau(s')} x_{s'} \mathbf{a}^b \mathbf{F}'_{\rho(s')}\right|^2\right] \quad (\text{A.9})$$

$$= \sum_{s' \in S - \{s\}} x_{s'}^2 E[|\mathbf{a}^b \mathbf{F}'_{\rho(s')}|^2] \quad (\text{A.10})$$

$$\leq \sum_{s' \in S - \{s\}} x_{s'}^2 CR/P \quad (\text{A.11})$$

$$= \|\mathbf{x}\|_2^2 CR/P \leq C/B \leq 1/K \frac{1}{6 \cdot (8\pi)^2} \quad (\text{A.12})$$

where we used Parseval's identity, Lemma A.1.4 and that B is large enough. Therefore, we have that $\Pr[X^2 \geq \frac{1}{(8\pi)^2}] \leq 1/6$. By Lemma A.1.5 we have that, with probability at least $1 - 1/6 - 1/6$, $T(s) \geq (\frac{1}{2(2\pi)} - \frac{1}{8\pi})^2 (\frac{1}{2(2\pi)})^2 / K$. \square

Proof of Theorem 4.1, Part (2). Suppose that $s \notin S$. We have

$$\begin{aligned}
& E[T(s)] \\
& \leq \sum_{b=0}^{B-1} E_{\rho(s), \rho(s'), \tau} \left[\left| \sum_{s' \in S} \omega^{\tau(s')} x_{s'}^2 \mathbf{a}^b \mathbf{F}'_{\rho(s')} \right|^2 \left| \mathbf{a}^b \mathbf{F}'_{\rho(s)} \right|^2 \right] \\
& = \sum_{b=0}^{B-1} E_{\rho(s'), \tau} \left[\left| \sum_{s' \in S} \omega^{\tau(s')} x_{s'}^2 \mathbf{a}^b \mathbf{F}'_{\rho(s')} \right|^2 \right] E_{\rho(s)} \left[\left| \mathbf{a}^b \mathbf{F}'_{\rho(s)} \right|^2 \right] \\
& \leq CR/P \sum_{b=0}^{B-1} \sum_{s' \in S} x_{s'}^2 E \left[\left| \mathbf{a}^b \mathbf{F}'_{\rho(s')} \right|^2 \right] \\
& \leq (CR/P)^2 B \|\mathbf{x}\|_2^2 \leq C^2/B \leq T/3
\end{aligned}$$

where we assume that B is large enough. By Markov inequality it follows that $\Pr[T(s) \geq T] \leq 1/3$. \square

Bibliography

- [1] *Anker Astro 5200mAh battery*. <https://www.amazon.com/Anker-bar-Sized-Portable-High-Speed-Technology/dp/B00P7N0320>.
- [2] *CNET review for Samsung Gear VR*. <http://www.cnet.com/products/samsung-gear-vr/>.
- [3] *Datasheet DAC-7228*. <http://www.analog.com/media/en/technical-documentation/data-sheets/AD7228.pdf>.
- [4] *Datasheet Phase Shifter HMC933*. <http://http://www.analog.com/media/en/technical-documentation/data-sheets/hmc933.pdf>.
- [5] E-ZPass statistics. <http://e-zpassiag.com/about-us/statistics>.
- [6] E-ZPass taking toll on turnpike collectors. http://www.meadvilletribune.com/news/local_news/article_735f7ca1-82f9-5340-8940-29608aa499d5.html.
- [7] *Facebook Expects to Ship 2.6 Million Oculus Rifts by 2017*. <http://www.businessinsider.com/facebook-expects-to-ship-26-million-oculus-rifts-by-2017-2016-4>.
- [8] *FCC to explore 5G services above 24 GHz*. <http://www.fiercewireless.com/tech/fcc-to-explore-5g-services-auctioned-or-unlicensed-above-24-ghz>.
- [9] *HTC sold 15,000 \$800 Vive virtual reality headsets in 10 minutes*. <http://venturebeat.com/2016/02/29/htc-sold-15000-800-vive-virtual-reality-headsets-in-10-minutes/>.
- [10] *HTC sold 15,000 \$800 Vive virtual reality headsets in 10 minutes*. <http://www.ubeeinteractive.com/sites/default/files/Understanding%20Technology%20Options%20%20for%20Deploying%20Wi-Fi%20White%20Paper.pdf>.
- [11] *HTC Vive Oculus Rift Spec Comparison*. <http://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/>.

- [12] *HTC VIVE Recommended Area*. <http://www.ibtimes.co.uk/htc-vive-vr-how-much-room-space-do-i-really-need-1558494>.
- [13] *IEEE 802.16 Broadband Wireless Access Working Group*. http://ieee802.org/16/maint/contrib/C80216maint-05_112r8.pdf.
- [14] *IEEE 802.16 Broadband Wireless Access Working Group*. http://www.keysight.com/upload/cmc_upload/All/22May2014Webcast.pdf?&cc=US&lc=eng.
- [15] *Look Inside the HTC Vive's Positional Tracking System*. http://http://www.gamasutra.com/view/news/273553/An_expert_look_inside_the_HTC_Vives_positional_tracking_system.php.
- [16] *mmWave 24GHz Transceivers*. <http://www.infineon.com/cms/en/product/rf-and-wireless-control/mm-wave-mmhc/channel.html?channel=db3a304339d29c450139d8bdb700579d>.
- [17] NTP org: NTP over cellular. http://support.ntp.org/bin/view/Support/ConfiguringNTP#Section_6.15.
- [18] *Optoma's wireless VR headset frees you from PC cables*. <http://www.pcworld.com/article/3044542/virtual-reality/optomas-new-wireless-vr-headset-frees-you-from-pc-cables.html>.
- [19] PA turnpike to make E-ZPass mandatory. <http://www.bizjournals.com/philadelphia/news/2013/08/29/pa-turnpike-to-make-e-zpass-mandatory.html>.
- [20] Paying with E-ZPass. <http://www.panynj.gov/airports/lga-paying-with-e-zpass.html>.
- [21] The problems with police radar. <http://www.ibiblio.org/rdu/a-btrust.html>.
- [22] *Samsung Gear VR*. <http://www.samsung.com/us/explore/gear-vr/>.
- [23] Solar powered RFID system. <http://www.gizmag.com/savi-technology-solar-rfid/8373/>.
- [24] Speed limits in the united states. http://en.wikipedia.org/wiki/Speed_limits_in_the_United_States.
- [25] *Sulon sneak peak*. <http://sulon.com/blog/sulon-q-sneak-peek>.
- [26] Tunpike tolls to be all-electronic by 2018. <http://paindependent.com/2013/08/no-e-zpass-heres-a-bill-electronic-tolling-planned-for-pa-turnpike>.

- [27] Urban design division/planning and development review department. http://austintexas.gov/sites/default/files/files/Planning/Urban_Design/great_street_site_overview_0512.pdf.
- [28] *Virtual Apple Airport Express*. <http://www.apple.com/airport-express/>.
- [29] *Virtual Reality in Entertainment*. <http://www.vrs.org.uk/virtual-reality-applications/entertainment.html>.
- [30] *Virtual Reality in Healthcare*. <http://www.vrs.org.uk/virtual-reality-healthcare/>.
- [31] *Visus VR*. <http://www.visusvr.com/>.
- [32] *VR for everyone*. <https://vr.google.com/>.
- [33] *Wireless HDMI*. <http://www.cnet.com/news/wireless-hd-video-is-here-so-why-do-we-still-use-hdmi-cables/>.
- [34] *WorldViz*. <http://aecmag.com/technology-mainmenu-35/1130-news-worldviz-brings-warehouse-scale-vr-to-unreal-and-unity-engines>.
- [35] Wilocity 802.11ad Multi-Gigabit Wireless Chipset. <http://wilocity.com>, 2013.
- [36] Short SSW Format for 11ay. <https://mentor.ieee.org/802.11/dcn/16/11-16-0416-01-00ay-short-ssw-format-for-11ay.pptx>, 2016.
- [37] 60 GHz: Taking the VR Experience to the Next Level. <http://www.sibeam.com>.
- [38] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. Cutting the cord in virtual reality. In *HotNets*, 2016.
- [39] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. Cutting the cord in virtual reality. In *HotNets*, 2016.
- [40] Omid Abari, Dinesh Bharadia, Austin Duffield, and Dina Katabi. Enabling high-quality untethered virtual reality. In *NSDI*, 2017.
- [41] Omid Abari, Ezz Hamed, Haitham Hassanieh, Abhishek Agarwal, Dina Katabi, Anantha P Chandrakasan, and Vladimir Stojanovic. A 0.75-million-point fourier-transform chip for frequency-sparse signals. In *IEEE ISSCC*, 2014.
- [42] Omid Abari, Haitham Hassanieh, and Dina Katabi. Millimeter wave communications: From point-to-point links to agile network connections.
- [43] Omid Abari, Haitham Hassanieh, Michael Rodreguiz, and Dina Katabi. Poster: A Millimeter Wave Software Defined Radio Platform with Phased Arrays. In *MOBICOM*, 2016.
- [44] Omid Abari, Hariharan Rahul, Dina Katabi, and Mondira Pant. Airshare: Distributed coherent transmission made seamless. In *INFOCOM*, 2015.

- [45] Omid Abari, Deepak Vasisht, Dina Katabi, and Anantha Chandrakasan. Caraoke: An e-toll transponder network for smart cities. In *SIGCOMM*, 2015.
- [46] Abracon “Crystal Oscillator”. www.abracon.com/Resonators/abm3c.pdf.
- [47] Adafruit. USB Voltage and Current Meter. <http://www.adafruit.com/product/1852>.
- [48] Ahmed Alkhateeb, Omar El Ayach, Geert Leus, and Robert W Heath. Channel estimation and hybrid precoding for millimeter wave cellular systems. *Selected Topics in Signal Processing, IEEE Journal of*, 2014.
- [49] Alexander Amini, Kevin Kung, Chaogui Kang, Stanislav Sobolevsky, and Carlo Ratti. The impact of social segregation on human mobility in developing and industrialized regions. *EPJ Data Science*, 2014.
- [50] Analog Device. ADC 7356. http://www.analog.com/static/imported-files/data_sheets/AD7356.pdf.
- [51] Christopher R. Anderson and Theodore S. Rappaport. In-Building Wideband Partition Loss Measurements at 2.5 and 60 GHz. *IEEE Transactions on Wireless Communications*, 3(3), May 2004.
- [52] Don Anderson, Tom Shanley, and Ravi Budruk. *PCI express system architecture*. Addison-Wesley Professional, 2004.
- [53] Daniel C Araújo, André LF de Almeida, Johan Axnas, and Joao Mota. Channel estimation for millimeter-wave very-large mimo systems. In *Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European*, pages 81–85. IEEE, 2014.
- [54] Khanh Do Ba, Piotr Indyk, Eric Price, and David P Woodruff. Lower bounds for sparse recovery. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1190–1197. SIAM, 2010.
- [55] W. Bajwa, J. Haupt, A Sayeed, and R. Nowak. Compressive wireless sensing. In *Information Processing in Sensor Networks, 2006. IPSN 2006. The Fifth International Conference on*, pages 134–142, 2006.
- [56] Horia Vlad Balan, Ryan Rogalin, Antonios Michaloliakos, Konstantinos Psounis, and Giuseppe Caire. Airsync: Enabling distributed multiuser mimo with full spatial multiplexing. 2012.
- [57] Constantine A Balanis. *Antenna theory: analysis and design*. John Wiley & Sons, 2012.
- [58] Paolo Baronti, Prashant Pillai, Vince WC Chook, Stefano Chessa, Alberto Gotta, and Y Fun Hu. Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. *Computer communications*, 30(7):1655–1695, 2007.

- [59] Dinesh Bharadia and Sachin Katti. Fastforward: Fast and constructive full duplex relays. In *SIGCOMM*, 2014.
- [60] Rudolf HJ Bloks. The IEEE-1394 high speed serial bus. *Philips Journal of Research*, 50(1):209–216, 1996.
- [61] BM Breakthrough Could Alleviate Mobile Data Bottleneck. <http://cacm.acm.org/news/164893-ibm-breakthrough-could-alleviate-mobile-data-bottleneck/fulltext>.
- [62] Carlo Alberto Boano, Marco Zúñiga, James Brown, Utz Roedig, Chamath Kerpitiyagama, and Kay Römer. Templab: a testbed infrastructure to study the impact of temperature on wireless sensor networks. In *ACM/IEEE ISPN*, 2014.
- [63] James A Bonneson and Montasir Abbas. Intersection video detection manual. Technical report, Texas Transportation Institute, Texas A & M University System, 2002.
- [64] Bosch Tools. Gln 50 laser distance measurer. <http://www.boschtools.com/>.
- [65] S. Boyd. *Lecture 12: Feedback control systems: static analysis*. <https://stanford.edu/~boyd/ee102/ctrl-static.pdf>.
- [66] Matt Branda. Qualcomm Research demonstrates robust mmWave design for 5G. Qualcomm Technologies Inc., November 2015.
- [67] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *ACM MobiCom*, 2008.
- [68] Cripps Steve C. *Advanced Techniques in RF Power Amplifier Design*. Artech House, 2002.
- [69] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52:489 – 509, 2006.
- [70] CBS Local. New app allows Boston drivers to pay for parking on smartphone. <http://boston.cbslocal.com/2015/01/14/new-app-allows-boston-drivers-to-pay-for-parking-on-smartphone/>.
- [71] Ranveer Chandra. Towards 7x improvement in battery life of mobile devices. In *Proceedings of the Eighth Wireless of the Students, by the Students, and for the Students Workshop*, pages 1–1. ACM, 2016.
- [72] Yuan Chen, Yu-Wei Lin, Yu-Chi Tsao, and Chen-Yi Lee. A 2.4-gsample/s dvfs fft processor for mimo ofdm communication systems. *IEEE Journal of Solid-State Circuits*, 43(5):1260–1273, 2008.

- [73] Chia-Lung Cheng, Fan-Ren Chang, and Kun-Yuan Tu. Highly accurate real-time GPS carrier phase-disciplined oscillator. *Instrumentation and Measurement, IEEE Transactions on*, 54(2):819–824, 2005.
- [74] Krishna Chintalapudi, Bozidar Radunovic, Horia Vlad Balan, Michael Buetener, Srinivas Yerramalli, Vishnu Navda, and Ramachandran Ramjee. wifinc:wifi over narrow channels. *NSDI'12*.
- [75] Cisco. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013.
- [76] Cisco, “Cisco Aironet 2600 Access Point”. http://www.cisco.com/en/US/prod/collateral/wireless/ps5678/ps12534/data_sheet_c78-709514.pdf.
- [77] Yong Cui, Shihan Xiao, Xin Wang, Zhenjie Yang, Chao Zhu, Xiangyang Li, Liu Yang, and Ning Ge. Diamond: Nesting the Data Center Network with Wireless Rings in 3D Space. In *NSDI*, 2016.
- [78] Welington M. da Silva, Alexandre Alvaro, Gustavo H. R. P. Tomas, Ricardo A. Afonso, Kelvin L. Dias, and Vinicius C. Garcia. Smart cities software architectures: A survey. In *ACM SAC*, 2013.
- [79] Ltd DMC R&D Center, Samsung Electronics Co. 5G Vision White Paper - Samsung. 2015.
- [80] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289 – 1306, 2006.
- [81] John C Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory*. Courier Corporation, 2013.
- [82] S. Eichler. Performance evaluation of the iee 802.11p wave communication standard. In *Vehicular Technology Conference*, 2007.
- [83] Mohammed E. Eltayeb, Ahmed Alkhateeb, Robert W. Heath, and Tareq Y. Al-Naffouri. Opportunistic Beam Training with Hybrid Analog/Digital Codebooks for mmWave Systems. In *GLOBESIP*, 2015.
- [84] EPCglobal Inc. Electronic Product Code (EPC): An overview. http://www.gs1.org/docs/epcglobal/an_overview_of_EPC.pdf.
- [85] EPCglobal Inc. EPCglobal class 1 generation 2 v. 1.2.0. <http://www.gs1.org/gsmf/kc/epcglobal/uhfc1g2>.
- [86] Sinem Coleri Ergen. ZigBee/IEEE 802.15.4 summary. <http://pages.cs.wisc.edu/~suman/courses/838/papers/zigbee.pdf>, 2004.

- [87] Definition and assessment of relay based cellular deployment concepts for future radio scenarios considering 1st protocol characteristics, Ch5. <https://www.ist-winner.org/DeliverableDocuments/D3.4.pdf>.
- [88] Escort Radar. The truth about speed enforcement. https://www.escortradar.com/pdf/radar_report.pdf.
- [89] Ettus, Universal Software Radio Peripheral. <http://www.ettus.com>.
- [90] European Commission. Smart Cities: Digital agenda for Europe. <http://ec.europa.eu/digital-agenda/en/smart-cities>.
- [91] Zahid Farid, Rosdiadee Nordin, and Mahamod Ismail. Recent advances in wireless indoor localization techniques and systems. *Journal of Computer Networks and Communications*, 2013.
- [92] FCC, “FCC online table of frequency allocation, April 16, 2013”.
- [93] FCC, Second Memorandum Opinion & Order 10-174.
- [94] Federal Communications Commission. Title 47, code for federal regulations, part 15.
- [95] FOX News. Company lets drive-thru customers pay for fast food with E-Zpass. <http://www.foxnews.com/leisure/2013/12/16/fast-food-drive-thrus-get-faster-company-lets-customers-pay-with-e-zpass/>.
- [96] Bo Gao, Zhenyu Xiao, Changming Zhang, Depeng Jin, and Lieguang Zeng. Joint SNR and Channel Estimation for 60 GHz Systems using Compressed Sensing. In *WCNC*, 2013.
- [97] Bo Gao, Zhenyu Xiao, Changming Zhang, Depeng Jin, and Lieguang Zeng. Sparse/dense channel estimation with non-zero tap detection for 60-GHz beam training. *IET Communications*, 8(11):2044–2053, 2014.
- [98] A. Gilbert, M. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal space fourier representations. In *SPIE*, 2005.
- [99] Anna C Gilbert, Sudipto Guha, Piotr Indyk, S Muthukrishnan, and Martin Strauss. Near-optimal sparse fourier representations via sampling. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 152–161. ACM, 2002.
- [100] Anna C Gilbert, Piotr Indyk, Mark Iwen, and Ludwig Schmidt. Recent developments in the sparse fourier transform: a compressed fourier transform for big data. *IEEE Signal Processing Magazine*, 31(5):91–100, 2014.
- [101] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. Phaser: enabling phased array signal processing on commodity WiFi access points. In *MobiCom*, 2014.

- [102] Shyamnath Gollakota, Matthew S Reynolds, Joshua R Smith, and David J Wetherall. The emergence of rf-powered computing. *Computer*, 47(1):32–39, 2014.
- [103] P. R. Gray and R. G. Meyer. Mos operational amplifier design—a tutorial overview. *IEEE Journal of Solid-State Circuits*, 1982.
- [104] GS1. Electronic Product Code Implementation. <http://www.gs1.org/epcglobal/implementation>.
- [105] Muhammad K. Haider and Edward W. Knightly. Mobility resilience and overhead constrained adaptation in directional 60 GHz WLANs: protocol design and system implementation. In *MobiHoc*, 2016.
- [106] Daniel Halperin, Srikanth Kandula, Jitendra Padhye, Paramvir Bahl, and David Wetherall. Augmenting Data Center Networks with Multi-Gigabit Wireless Links. In *SIGCOMM*, 2011.
- [107] Shuangfeng Han, ChihLin I, Zhikun Xu, and Corbett Rowell. Large-Scale Antenna Systems with Hybrid Analog and Digital Beamforming for Millimeter Wave 5G. *IEEE Communications Magazine*, January 2015.
- [108] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Nearly optimal sparse fourier transform. In *STOC*, 2012.
- [109] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Simple and practical algorithm for sparse FFT. In *SODA*, 2012.
- [110] Haitham Hassanieh, Lixin Shi, Omid Abari, Ezzeldin Hamed, and Dina Katabi. Bigband: Ghz-wide sensing and decoding on commodity radios. In *IEEE INFOCOM*, 2014.
- [111] Ken’ichi Hosoya, Narayan Prasad, Kishore Ramachandran, Naoyuki Orihashi, Shuya Kishimoto, Sampath Rangarajan, and Kenichi Maruhashi. Multiple sector id capture (mide): A novel beamforming technique for 60-ghz band multi-gbps wlan/pan systems. *IEEE Transactions on Antennas and Propagation*, 2015.
- [112] HP. Fundamentals of quartz oscillators. Technical report, Appl. note 200.
- [113] An-swol Hu and Sergio D Servetto. dFSK: Distributed frequency shift keying modulation in dense sensor networks. In *IEEE ICC*, 2004.
- [114] Pan Hu, Pengyu Zhang, and Deepak Ganesan. Fully asymmetric backscatter communication. In *ACM SIGCOMM*, 2015.
- [115] IEEE. IEEE Smart Cities. <http://smartcities.ieee.org/>.
- [116] IEEE Standards Association. IEEE Standards 802.11ad-2012: Enhancements for Very High Throughput in the 60 GHz Band, 2012.

- [117] Mark Iwen, Aditya Viswanathan, and Yang Wang. Robust sparse phase retrieval made easy. *Applied and Computational Harmonic Analysis*, 42(1):135–142, 2017.
- [118] Jackson Labs, Fury GPSDO. <http://jackson-labs.com/>.
- [119] Kishore Jaganathan, Yonina C Eldar, and Babak Hassibi. Phase retrieval: An overview of recent developments. *arXiv preprint arXiv:1510.07713*, 2015.
- [120] Yalei Ji, Cedomir Stefanovic, Carsten Bockelmann, Armin Dekorsy, and Petar Popovski. Characterization of coded random access with compressive sensing based multi-user detection. *CoRR*, abs/1404.2119, 2014.
- [121] Kapsch TrafficCom. Active TDM LMS specifications. <http://tdm.kapschtraffic.com/>.
- [122] John Kilpatrick, Robbie Shergill, and Manish Sinha. 60 GHz Line of Sight Backhaul Links Ready to Boost Cellular Capacity. Analog Devices Inc.
- [123] Joongheon Kim and Andreas F. Molisch. Fast Millimeter-Wave Beam Training with Receive Beamforming. *Journal of Communications and Networks*, October 2014.
- [124] Swarun Kumar, Ezzeldin Hamed, Dina Katabi, and Li Erran Li. LTE radio analytics made easy and accessible. In *ACM SIGCOMM*, 2014.
- [125] KY3. Springfield’s high tech cameras count cars to ease traffic congestion. http://articles.ky3.com/2012-06-20/springfield-traffic_32340154.
- [126] Bin Li, Zheng Zhou, Weixia Zou, Xuebin Sun, and Guanglong Du. On the Efficient Beam-Forming Training for 60GHz Wireless Personal Area Networks. *IEEE Transactions on Wireless Communications*, 12(2), February 2013.
- [127] Fan Li and Yu Wang. Routing in vehicular ad hoc networks: A survey. *IEEE Vehicular Technology Magazine*, 2007.
- [128] Sung Hoon Lim, Young-Han Kim, Abbas El Gamal, and Sae-Young Chung. Noisy network coding. *IEEE Transactions on Information Theory*, 57(5):3132–3152, 2011.
- [129] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R Smith. Ambient backscatter: wireless communication out of thin air. *ACM SIGCOMM Computer Communication Review*, 2013.
- [130] Yue Liu, Jun Bi, and Ju Yang. Research on vehicular ad hoc networks. In *CCDC*, 2009.
- [131] Ivana Maric, Nan Liu, and Andrea Goldsmith. Encoding against an interferer’s codebook. In *46th Annual Allerton Conference on Communication, Control, and Computing, 2008*, pages 523–530. IEEE, 2008.

- [132] Maxim Integrated. MAX2117. <http://datasheets.maximintegrated.com/en/ds/MAX2117.pdf>.
- [133] Maxim Integrated, "VHF/UHF Low-Noise Amplifiers." <http://datasheets.maximintegrated.com/en/ds/MAX2664-MAX2665.pdf>.
- [134] McKinsey & Company. The smart-city solution. http://www.mckinsey.com/insights/public_sector/the_smart-city_solution.
- [135] Juan C Medina, Madhav Chitturi, Rahim F Benekohal, and Transportation Research Board. Illumination and wind effects on video detection performance at signalized intersections. *Urbana*, 2008.
- [136] Renato E Mirolo and Steven H Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, 1990.
- [137] L.S. Mojela and M.J. Booyen. On the use of WiMAX and Wi-Fi to provide in-vehicle connectivity and media distribution. In *ICIT*, 2013.
- [138] Cristina D Murta, Pedro R Torres Jr, and Prasant Mohapatra. Characterizing quality of time and topology in a time synchronization network. In *GLOBECOM*, 2006.
- [139] Bobak Nazer and Michael Gastpar. The case for structured random codes in network capacity theorems. *European Transactions on Telecommunications*, 19(4):455–474, 2008.
- [140] New Jersey Turnpike Authority. Travel resources: E-ZPass information. <http://www.state.nj.us/turnpike/ez-pass.html>.
- [141] NIST. NIST Radio Station WWVB. <http://www.nist.gov/pml/div688/grp40/wwvb.cfm>.
- [142] NIST. Radio Station WWV. <http://www.nist.gov/pml/div688/grp40/wwv.cfm/>.
- [143] Thomas Nitsche, Guillermo Bielsa, Irene Tejado, Adrian Loch, and Joerg Widmer. Boon and bane of 60 ghz networks: Practical insights into beamforming, interference, and frame level operation. In *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*. ACM, 2015.
- [144] Thomas Nitsche, Carlos Cordeiro, Adriana B Flores, Edward W Knightly, Eldad Perahia, and Joerg C Widmer. IEEE 802.11 ad: directional 60 GHz communication for multi-Gigabit-per-second Wi-Fi. *IEEE Communications Magazine*, 2014.
- [145] Thomas Nitsche, Adriana B Flores, Edward W Knightly, and Joerg Widmer. Steering with eyes closed: mm-wave beam steering without in-band measurement. In *INFOCOM*, 2015.

- [146] OSEPP. Monocrystalline Solar cell. <http://osepp.com/products/solar-cells-arduino-compatible/>.
- [147] A. Ozgur, R. Johari, D.N.C. Tse, and O. Leveque. Information-theoretic operating regimes of large wireless networks. *Info. Theory Trans.*, 2010.
- [148] Zhouyue Pi and Farooq Khan. An introduction to millimeter-wave mobile broadband systems. *Communications Magazine, IEEE*, 2011.
- [149] Eric Price and David P Woodruff. $(1 + \epsilon)$ -approximate sparse recovery. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 295–304. IEEE, 2011.
- [150] Franeois Quitin, Muhammad Mahboob Ur Rahman, Raghuraman Mudumbai, and Upamanyu Madhow. A scalable architecture for distributed transmit beamforming with commodity radios: Design and proof of concept. *Wireless Communications, IEEE Transactions on*, 12(3):1418–1428, March 2013.
- [151] Hariharan Rahul, Haitham Hassanieh, and Dina Katabi. Sourcesync: A distributed wireless architecture for exploiting sender diversity. *SIGCOMM*, 2010.
- [152] Hariharan Rahul, Haitham Hassanieh, and Dina Katabi. SourceSync: A distributed wireless architecture for exploiting sender diversity. In *ACM SIGCOMM*, 2010.
- [153] Hariharan Rahul, Swarun Suresh Kumar, and Dina Katabi. MegaMIMO: Scaling wireless capacity with user demand. In *ACM SIGCOMM*, 2012.
- [154] Dinesh Ramasamy, Subramanian Venkateswaran, and Upamanyu Madhow. Compressive tracking with 1000-element arrays: A framework for multi-gbps mm wave cellular downlinks. In *Allerton*, 2012.
- [155] Sundeep Rangan, Theodore S Rappaport, and Elza Erkip. Millimeter-wave cellular wireless networks: Potentials and challenges. *Proceedings of the IEEE*, 2014.
- [156] Theodore S Rappaport, James N Murdock, and Felix Gutierrez. State of the art in 60GHz integrated circuits and systems for wireless communications. *Proceedings of the IEEE*, 2011.
- [157] Wonil Roh, Ji-Yun Seol, JeongHo Park, Byunghwan Lee, Jaekon Lee, Yungsoo Kim, Jaeweon Cho, Kyungwhoon Cheun, and Farshid Aryanfar. Millimeter-Wave Beamforming as an Enabling Technology for 5G Cellular Communications: Theoretical Feasibility and Prototype Results. *IEEE Communications Magazine*, February 2014.
- [158] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Ranveer Chandra, and Jie Liu. Snow: Sensor network over white spaces. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 272–285. ACM, 2016.

- [159] Scott Y Seidel and Theodore S Rappaport. 914 mhz path loss prediction models for indoor wireless communications in multifloored buildings. *IEEE Trans. on Antennas and Propagation*, 1992.
- [160] Mingoo Seok, Dongsuk Jeon, Chaitali Chakrabarti, David Blaauw, and Dennis Sylvester. A 0.27 v 30mhz 17.7 nj/transform 1024-pt complex fft core with super-pipelining. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, pages 342–344. IEEE, 2011.
- [161] Sequans Communications. Chipset solution for category 1 LTE devices for the internet of things. <http://www.sequans.com/products-solutions/streamlitelte/calliope-lte-platform/>.
- [162] Clayton Shepard, Hang Yu, Narendra Anand, Li Erran Li, Thomas Marzetta, Richard Yang, and Lin Zhong. Argos: Practical Many-Antenna Base Stations. In *MobiCom*, 2012.
- [163] Clayton Shepard, Hang Yu, and Lin Zhong. ArgosV2: A Flexible Many-Antenna Research Platform. In *MobiCom*, 2013.
- [164] SiBeam, Lattice Semiconductor. www.sibeam.com.
- [165] O. Somekh and S. Shamai. A shannon-theoretic view of wyner’s multiple-access cellular channel model in the presence of fading. In *IEEE Int. Symp. on Info. Theory*, Aug 1998.
- [166] M. Speth et al. Optimum receiver design for wireless broad-band systems using OFDM- I. *Comm, IEEE Trans. on*, 47(11):1668 –1677, November 1999.
- [167] Streetline. Streetline Inc. <http://www.streetline.com/>.
- [168] Sanjib Sur, Vignesh Venkateswaran, Xinyu Zhang, and Parameswaran Ramanathan. 60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling. In *SIGMETRICS*, 2015.
- [169] Sanjib Sur and Xinyu Zhang. BeamScope: Scoping Environment for Robust 60 GHz Link Deployment. In *Information Theory and Application Workshop*, 2016.
- [170] Sanjib Sur, Xinyu Zhang, Parameswaran Ramanathan, and Ranveer Chandra. BeamSpy: Enabling Robust 60 GHz Links Under Blockage. In *NSDI*, 2016.
- [171] Vamsi Talla, Bryce Kellogg, Benjamin Ransford, Saman Naderiparizi, Joshua R Smith, and Shyamnath Gollakota. Powering the next billion devices with wi-fi. *Communications of the ACM*, 2017.
- [172] Kun Tan, Ji Fang, Yuanyang Zhang, Shouyuan Chen, Lixin Shi, Jiansong Zhang, and Yongguang Zhang. Fine grained channel access in wireless LAN. In *SIGCOMM*, 2010.

- [173] Texas Instruments. Energy harvesting. http://www.ti.com/corp/docs/landing/cc430/graphics/slyy018_20081031.pdf.
- [174] Texas Instruments, "True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications." <http://www.ti.com/lit/ds/symalink/cc2530.pdf>.
- [175] Zong Tian, Michael Kyte, and Hongchao Liu. Vehicle tracking and speed measurement at intersections using video detection systems. *ITE Journal*, 2009.
- [176] TPLink. Talon AD7200 Wireless Wi-Fi Tri-Band Gigabit Router. <http://www.tp-link.com/en/products/details/AD7200.html>.
- [177] Transportation Corridor Agencies. FasTrak on new I-10 and I-110 ExpressLanes. <https://www.thetollroads.com/communications/switchable-transponders-110.php>.
- [178] Transportation Corridor Agencies. What is FasTrak? how do i use it to pay tolls? <https://www.thetollroads.com/ontheroads/commonquestions/fastrak.php>.
- [179] Trimble, Thunderbolt GPSDO. trimble.com/timing/thunderbolt-e.aspx.
- [180] Y. Ming Tsang, Ada S. Y. Poon, and Sateesh Addepalli. Coding the Beams: Improving Beamforming Training in mmWave Communication System. In *IEEE GLOBECOM*, 2011.
- [181] D. Tse and P. Vishwanath. *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.
- [182] UMTS Forum. Mobile traffic forecasts: 2010-2020 report, 2011.
- [183] Deepak Vasisht, Z Kapetanovic, Jongho Won, Xinxin Jin, Ranveer Chandra, Sudipta Sinha, Ashish Kapoor, Madhusudhan Sudarshan, and Sean Stratman. Farmbeats: An iot platform for data-driven agriculture. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, pages 515–529. USENIX Association.
- [184] Ruud JM Vullers, RV Schaijk, Hubregt J Visser, Julien Penders, and Chris Van Hoof. Energy harvesting for autonomous wireless sensor networks. *IEEE Solid-State Circuits Magazine*, 2010.
- [185] Jue Wang, Haitham Hassanieh, Dina Katabi, and Piotr Indyk. Efficient and reliable low-power backscatter networks. In *ACM SIGCOMM*, 2012.
- [186] Jue Wang and Dina Katabi. Dude, where's my card?: RFID positioning that works with multipath and non-line of sight. In *ACM SIGCOMM*, 2013.

- [187] Junyi Wang, Zhou Lan, Chang-Woo Pyo, Tuncer Baykas, Chin-Sean Sum, M Azizur Rahman, Jing Gao, Ryuhei Funada, Fumihide Kojima, Hiroshi Harada, and Shuzo Kato. Beam Codebook Based Beamforming Protocol for Multi-Gbps Millimeter-Wave WPAN Systems. *IEEE Journal of Selected Areas in Communications*, 27(8), October 2009.
- [188] Xiao Y Wang, Rajeev K Dokania, and Alyssa Apsel. Pco-based synchronization for cognitive duty-cycled impulse radio sensor networks. *IEEE Sensors Journal*, 11(3):555–564, 2011.
- [189] Xiao Y Wang, Rajeev K Dokania, Yi Zhuang, Waclaw Godycki, Carlos I Dorta-Quinones, Michael Lyons, and Alyssa B Apsel. A self-synchronized, crystal-less, $86\mu\text{w}$, dual-band impulse radio for ad-hoc wireless networks. In *Radio Frequency Integrated Circuits Symposium (RFIC), 2011 IEEE*, pages 1–4. IEEE, 2011.
- [190] AD. Wyner. Shannon-theoretic approach to a gaussian cellular multiple-access channel. *IEEE Trans. on Info. Theory*, 1994.
- [191] Jie Xiong and Kyle Jamieson. ArrayTrack: A fine-grained indoor location system. In *NSDI*, 2013.
- [192] Zhang Xiong, Hao Sheng, WenGe Rong, and DaveE. Cooper. Intelligent transportation systems for smart cities: a progress review. *Science China Information Sciences*, 2012.
- [193] Shouzhi Xu, Pengfei Guo, Bo Xu, and Huan Zhou. QoS evaluation of VANET routing protocols. *JNW*, 2013.
- [194] Chia-Hsiang Yang, Tsung-Han Yu, and Dejan Markovic. Power and area minimization of reconfigurable fft processors: A 3gpp-lte example. *IEEE journal of solid-state circuits*, 47(3):757–768, 2012.
- [195] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices. In *ACM MobiCom*, 2014.
- [196] Xuhang Ying, Jincheng Zhang, Lichao Yan, Guanglin Zhang, Minghua Chen, and Ranveer Chandra. Exploring indoor white spaces in metropolises. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 255–266. ACM, 2013.
- [197] Wenfang Yuan, Simon M. D. Armour, and Angela Doufexi. An Efficient and Low-complexity Beam Training Technique for mmWave Communication. In *PIMRC*, 2015.
- [198] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

- [199] Jialiang Zhang, Xinyu Zhang, Pushkar Kulkarni, and Parameswaran Ramanathan. OpenMili: A 60 GHz Software Radio Platform With a Reconfigurable Phased-Array Antenna. In *MobiCom*, 2016.
- [200] Guichang Zhong, Fan Xu, and Alan N Willson. A power-scalable reconfigurable fft/iff ic based on a multi-processor ring. *IEEE Journal of solid-state circuits*, 41(2):483–495, 2006.
- [201] Liang Zhou and Yoji Ohashi. Efficient Codebook-Based MIMO Beamforming for Millimeter-Wave WLANs. In *PIMRC*, 2012.
- [202] X. Zhou, Z. Zhang, Y. Zhu, Y. Li, S. Kumar, A. Vahdat, B. Y. Zhao, and H Zheng. Mirror on the Ceiling: Flexible Wireless Links for Data Centers. In *SIGCOMM*, 2012.
- [203] Yibo Zhu, Zengbin Zhang, Zhinus Marzi, Chris Nelson, Upamanyu Madhow, Ben Y. Zhao, and Haitao Zheng. Demystifying 60GHz Outdoor Picocells. In *MOBICOM*, 2014.
- [204] Tetiana Zinchenko, Hugues Tchouankem, Lars Wolf, and André Leschke. Reliability analysis of vehicle-to-vehicle applications based on real world measurements. In *VANET*, 2013.