SCALABLE MILLIMETER WAVE WIRELESS NETWORKS

BY

SURAJ JOG

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2023

Urbana, Illinois

Doctoral Committee:

Assistant Professor Haitham Hassanieh, Chair
Dr. Ranveer Chandra
Professor Romit Roy Choudhury
Associate Professor Sachin Katti
Professor Josep Torrellas

ABSTRACT

Millimeter wave (mmWave) technology promises to revolutionize wireless networks. The huge abundance of bandwidth available at the mmWave frequencies (above 24 GHz) allows us to build ultra-low latency and high-data rate wireless links. This opens up completely new application domains, such as multi-user wireless VR/AR for education and professional training, large-scale robotic factory automation based on real-time video streams, and multiplayer gaming. Additionally, the availability of large bandwidth also enables very high resolution sensing and imaging capabilities, since sensing resolution is directly proportional to signal bandwidth.

However, leveraging these capabilities of mmWave networks to build communication and sensing systems at scale is challenging due to the unique characteristics of mmWave signals that set it apart from legacy wireless technologies. As a result, traditional wireless networking architectures do not translate well to mmWave networks and cannot exploit the opportunities made available by the millimeter-wave modality. This shift in paradigm that accompanies mmWave signals is also the reason why past work has been able to demonstrate the performance leaps with mmWave only in the context of single communication links, and in controlled and small scale setups.

The central question that this dissertation asks is – *"How can we design and build millimeter-wave systems that allow it to scale to realistic large-scale deployments with multiple heterogenous nodes, while also expanding the capabilities of these next-generation systems?"* This dissertation investigates the design of such scalable millimeter-wave communication and sensing systems for a number of different application domains such as Wireless LANs, Massive Multicore Processors, High Performance Computing (HPC), and IoT localization and tracking. We propose new networking architectures and protocols optimized for mmWave wireless links, that can naturally scale to many nodes in the network while providing seamless multi-Gbps connectivity. We also build mmWave sensing systems that can scale hyper-precise sensing and localization to large networks with ubiquitously deployed heterogenous nodes, without requiring any additional infrastructure support or modifications. Finally, we also show how mmWave could transform new application domains such as high-performance computing and address the practical

scalability challenges in these new fields. This dissertation introduces hardware-software co-designed systems for mmWave networks that can seamlessly scale to very large deployments, and we build proof-of-concept testbeds to demonstrate the efficacy of our proposed systems and present our learnings and insights from these real world deployments.

Dedicated to my parents and sister, for their unconditional love and support.

ACKNOWLEDGMENTS

The work presented in this dissertation would not have been possible without the help and support of a large group of people to whom I owe a lot of gratitude.

First and foremost, I am truly thankful to my advisor, Prof. Haitham Hassanieh, who, for these past six years, has tirelessly supported me and believed in me. I often tell people that doing the PhD has been the most transformative experience of my life, in terms of my thought process, resolve, and self-belief, and I genuinely believe that Haitham has been the central figure to bring about this transformation in me. He has taught me to never settle in life when it is a matter of your ambitions, and has enabled me to grow and achieve goals beyond my wildest dreams. He has been, and will always be, my role model, and I will forever be indebted to him for all that he has done.

I am very grateful to my Ph.D. committee members and mentors too – Prof. Romit Roy Choudhury, Dr. Ranveer Chandra, Prof. Josep Torrellas, and Prof. Sachin Katti. Romit has been like a second advisor to me through my Ph.D. He has supported me, encouraged me, and guided me through a lot of problems. Every interaction that I've had with him has broadened my perspective on research and has helped me appreciate the work we do and its significance. I truly admire his wisdom and his genuine eagerness to pursue knowledge. My interaction with Ranveer began when he mentored me at Microsoft Research during my summer internship in 2018. Working with him during the internship was one of the most fruitful and fun experiences I have ever had. Ranveer's friendly demeanor and positivity made him very approachable and I could brainstorm with him about completely left-field ideas. He placed immense trust in me and gave me a lot of freedom to pursue things that genuinely excited me. I am constantly amazed by Ranveer's ability to keep his sight on the big picture, and he has been a source of inspiration for me. Ranveer showed me how to have fun while working hard at research, and I am grateful to have gotten a chance to work with him early in my PhD journey. I want to express my heartfelt thanks to Josep for his guidance and support while working on the Wireless Network-on-Chip project. Without the expertise and domain knowledge that he brought to the table, it would not have been possible to complete this interdisciplinary project. Last but not the least, I also want to thank Sachin for all his suggestions and

kind support to improve the work presented in this dissertation.

I also owe my sincere gratitude to many other mentors and collaborators I have had along the way. They are Prof. Deepak Vashisht, Prof. Saurabh Gupta, Prof. Songbin Gong, Dr. Radhika Gowaikar, and Prof. R. Srikant. Deepak is, in fact, the reason why I decided to pursue a Ph.D. in the area of wireless networking. My first interaction with Deepak was when I had emailed him out of the blue in 2016, to get advice about doing a PhD and about the area of wireless networking systems. Despite not knowing me, Deepak took the time to talk to me and answered all my questions in the most patient manner. I really admire this selfless quality of Deepak, and subsequently, throughout my PhD, whenever I needed guidance on anything, I have always been able to turn to Deepak for sound advice. I am grateful to have had him as a mentor.

A lot of my dissertation work could not have been done without the help of a few close collaborators, in particular, Junfeng Guan, Sohrab Madani, Jiaming Wang, Zikun Liu, Anadi Chaman, Antonio Franques and Thomas Moon. Sohrab and Junfeng, in addition to being collaborators, have also been close friends. I treasure the memories of jamming to Bohemian Rhapsody and laughing about literally anything with Sohrab throughout the countless nights we've spent working on paper deadlines. Junfeng is among the most helpful and selfless individuals I have ever met, and he has always been the first to step up whenever I needed support of any kind. I am also grateful to the members of the Systems and Networking Research Group (SyNRG) at UIUC for their support and guidance, and to Carol, our former office assistant, whose warm and comforting presence helped me overcome some tough times.

As I complete my Ph.D. journey, I cannot help but look back and thank people who enabled me to get here in the first place. I am very grateful to Prof. D. Manjunath and Prof. Jayakrishnan Nair who mentored my undergraduate research project and helped me write my first paper. I owe deep gratitude to Prof. Manjunath for having believed in me and for taking a chance on me when I had no research experience and was just a novice. It was that experience of working with them that motivated me to pursue a Ph.D. in the first place, and subsequently defined the career trajectory that I set upon.

I also must acknowledge the National Science Foundation for supporting my Ph.D. study and research. I would like to thank Qualcomm as well, for awarding me the Qualcomm Innovation Fellowship.

I am very fortunate to have found my closest friends at UIUC in my roommates, Suryanarayana, Pranjal and Saboo. They have been a constant source of support and motivation through thick and thin, and I cannot thank them enough. Each of them are extraordinary individuals and they inspire me to aim higher. I must also thank Anadi, who I worked with in close quarters for the first two years of my Ph.D. I have learned a lot from him, both academically and in life, through my numerous discussions with him. Additional thanks to Vaishnavi for all the great food she has fed me over the last six years, Ashish for

being such a great and empathetic friend, and helping me out whenever I needed, Anjali who has taught me that no obstacles in life are too big to overcome, and to the members of my band, *White Noise*, with whom I've had a lot of fun and some truly memorable performances.

I also want to thank all the obstacles, rejections, and failures in my life. They made me stronger and helped me improve.

I also want to thank my family for all their love and support. To Suresh kaka and Usha aunty, for being such an integral part of my childhood and almost like a second set of parents. To my brother-in-law, Dr. Manoj Kumar, who I have immense respect for as a person and as a doctor. To my uncle, Raja Patwardhan, who inspires me everyday with all the social work he does for the upliftment of the underprivileged sections of society in rural India. In my life, I hope to do something that would amount to even a fraction of what Raja mama has contributed to people's lives.

Lastly, I cannot express sufficient gratitude to my parents Sucheta and Sanjay Jog and my sister Dr. Komal Jog, for their endless love, support and advice. Through the course of my Ph.D, there was a new addition to our family, my nephew Shourya, who has brought such immense joy and happiness in all our lives. I owe all my success to my family, and I could not have achieved what I did without their help. No matter what I do, I can never repay them.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

The future of networking systems that we envision goes far beyond communication alone and strives to deliver high resolution and ubiquitous sensing capabilities as well, in addition to robust and high data rate connectivity. The next-generation of wireless networks will move towards this goal by providing unprecedented new capabilities – gigabyte communication speeds, hyper-precise localization, and vision-like environmental perception. As a result, this will enable new applications – wireless virtual and augmented reality, fully-autonomous driving, space communications, precision agriculture in connected farms, and high-performance computing (HPC).

At its core, the key feature of next-generation wireless networks that enables all these applications is the availability of higher bandwidth as the technology transitions into higher frequency bands of operation, specifically the millimeter-wave (mmWave) frequency bands from 30 GHz to 300 GHz. At a fundamental level, this higher bandwidth will benefit both the communication and sensing performance of next-generation wireless networks. For instance, the latest WiFi standard of 802.11ad/ay [1, 2] operating in the millimeter wave (60 GHz) frequency bands, allocates channels spanning up to 2.16 GHz bandwidth which in turn allows for communication data rates up to 20 Gbps. In contrast, today's WiFi networks operating at sub-6 GHz frequencies can only allocate channel bandwidths up to 40 MHz, and consequently, can achieve only up to 600 Mbps data rates. Similarly, next-generation 5G signals in the millimeter-wave band can span up to 400 MHz which can enable localization and sensing accuracy of 75 centimeters. On the other hand, current 4G signals are allocated only up to 20 MHz channels, thus resulting in poor localization resolution of 15 meters (20x worse).

While the benefits of exploiting these mmWave frequency bands are significant, past work has been able to demonstrate these performance leaps only in the context of single communication links, and in controlled and small scale settings. The problem of scaling millimeter-wave technology to large network deployments with multiple heterogenous nodes while maintaining the same next-generation performance gains is still largely unsolved. For instance, while the higher bandwidth in 802.11ad WiFi can easily support a single wireless VR user in a room, being able to support multiple concurrent users is challeng-

ing given the unique characteristics of 802.11ad WiFi such as directional transmissions which requires rethinking of the interference model between links. Similarly, while 5G base stations can support transmissions up to 400 MHz in bandwidth, not every client device will be capable of receiving such high bandwidth transmissions. For example, low power IoT devices will typically have much slower Analog-to-Digital Converters (ADCs) and can capture only a very small portion of the wideband transmitted signals. As a result, we need to develop techniques that can preserve the wideband localization resolution, and scale the system to accommodate such ubiquitously deployed narrowband IoT devices. Lastly, millimeter-wave technologies are positioned to make a big impact in new application domains such as novel interconnect paradigms for massive multicore processors. While it has been shown in the computer architecture community that parallel processing performance can significantly benefit from wireless interconnects for cache coherency, to truly realize the benefits of wireless interconnects and to scale the multicore beyond 100 cores or more, we need to design novel networking protocols for the wireless network-on-chip that can make the most efficient use of the shared millimeter-wave wireless medium by optimizing for the underlying traffic statistics of the applications.

This dissertation tackles this question of scalability in next-generation millimeter-wave wireless networks. In this dissertation, I describe several end-to-end systems that we built to advance the state-of-the-art and enable multiple application domains: Wireless VR/AR streaming, Ambient Localization of Low Power IoT Devices, and Scaling Wireless Networks-on-Chip to massive multicore processors. In building these systems, we develop novel algorithms and techniques by leveraging unique insights from across all layers of the computing and network stack, from the hardware all the way to the application context. Finally, we also deployed these systems in real world environments and I present our deployment experiences and insights in this dissertation. Below, I describe the systems that we designed and built, with each system addressing a unique scalability challenge faced by millimeter-wave technologies across varied application domains.

1.1 Systems Developed

1.1.1 Many-to-Many Beam Alignment for Scaling mmWave WLANs by leveraging Dense Spatial Reuse

Millimeter-Wave (mmWave) networks can deliver multi-Gbps wireless links which will enable new applications like multi-user wireless VR and AR for education and professional training, 8K video content

streaming, and large scale robotic factory automation which relies on real-time video feeds. Enabling the above vision, however, requires scaling mmWave networks from a single communication link to a network of many links without compromising the throughput of each user. Fortunately, next-generation millimeter-wave radios offer a new dimension for scalability, since they use very directional steerable narrow beams. This allows for dense spatial reuse that can enable many links to simultaneously communicate without interfering. In order to communicate at the highest data rates, the mmWave APs and clients need to align their narrow beams towards each other. While past work focuses on developing algorithms and protocols to quickly find the best alignment for a single communication link, in [3] we show that in a network with multiple links, selfishly choosing the best alignment for each AP-client link independent of other links can create significant interference due to multipath reflections. We introduce **BounceNet** in [3], a system that addresses this scalability bottleneck and presents the first *"Many-to-Many Beam Alignment"* protocol that can enable extremely dense spatial reuse in millimeter-wave networks where many links can communicate simultaneously at multi-Gbps data rates without interfering.

BounceNet's key intuition is to leverage the sparsity in the mmWave wireless channel to reformulate the many-to-many alignment problem as a signal level routing problem at the physical layer using multi-layered graph constructs. We demonstrate that such a cross-layer protocol design which optimizes across both the sparsity in the mmWave PHY along with the network-layer configuration of the links, allows BounceNet to leverage both direct and reflected propagation paths to route the signals and densely pack as many links as possible in the confined 3D space. We show that in dense networks, BounceNet is able to deliver 3.1×-13.5× higher throughput per client. BounceNet introduces a new bridge between the link layer and PHY layer of the mmWave network stack to enable *"Physical Signal Routing"*, and, in turn, allows the network to scale easily.

1.1.2 Scaling Millimeter-Wave Wireless Networks-on-Chip for Massive Multicore Processors

Wireless Network-on-Chip (NoC) has emerged as a promising solution to scale chip multicore processors to hundreds and thousands of cores. The broadcast nature of a wireless network allows it to significantly reduce the latency and overhead of many-to-many multicast and broadcast communications, which forms the bulk of the NoC traffic. However, the traffic patterns on wireless NoCs tend to be very dynamic and can change drastically across different cores, different time intervals, and different applications. Further, due to thread synchronization primitives likes barriers and locks that are commonly used in parallel programming, the wireless NoC exhibits complex hard-to-model dependencies between packet delivery

time on the NoC and the progress of execution on the threads. As a result, traditional wireless MAC protocols perform very poorly in wireless NoCs since they remain agnostic to these domain specific dependencies and cannot adapt to the fast varying traffic.

To address this challenge, we propose a unified approach in **NeuMAC** [4], that combines networking, architecture, and deep learning to generate highly adaptive medium access protocols for wireless NoC architectures that can directly optimize for the non-trivial dependencies between threads purely through experience. NeuMAC leverages the key insight that many building block functions like FFT, graph search and sorting, repeatedly appear in many applications as common subroutines, which leads to predictability in traffic traces. NeuMAC capitalizes on this predictability by leveraging a reinforcement learning framework with deep neural networks to generate new MAC policies that can learn the structure, correlations and statistics of the traffic patterns. NeuMAC can adapt quickly to optimize performance for different applications leading to low latency, high throughput and an overall reduction in execution time of 1.37×-3.74× for a diverse set of parallel applications.

1.1.3 Scaling High-Resolution Self-Localization for Massive IoT Network Deployments using Ambient 5G Signals

Recent years have witnessed a tremendous growth in the number of connected IoT devices, which form a critical component of the network infrastructure in applications such as precision agriculture, smart city monitoring, and Industry 4.0. With such ubiquitous deployment of IoT nodes, the ability to localize and track them with high accuracy is critical. In [5, 6], we introduce **ISLA**, which enables low power IoT devices to accurately self-localize themselves simply by snooping on ambient 5G signals, without requiring any coordination or synchronization with 5G base stations. The 5G standard supports very high communication bandwidths (up to 400 MHz), which, in turn, enables very high resolution in ToF (Time-of-Flight) estimates (up to 75 cm resolution) for localization. Further, the ability to self-localize allows ease of deployment at scale since there is no need to modify the 5G base stations to support the localization feature.

However, leveraging these opportunities on power-constrained and low-cost IoT devices is challenging. IoT devices are equipped with cheap and low-speed Analog-to-Digital converters (ADCs) which cannot capture the large bandwidth of 5G signals, and, in turn, significantly lose out on the high ToF resolution. In ISLA [5, 6], we introduce the first RF-acoustic system that leverages MEMS acoustic resonators to design a new kind of RF filter that can stretch the effective localization bandwidth by 16× on these narrowband IoT devices. Specifically, we design a MEMS filter that emulates a spike-train in the frequency

domain. This allows us to subsample and sparsify the 5G signal in the frequency domain such that the end-to-end bandwidth spanned by the filtered signal is preserved. This, in turn, means that the high ToF resolution is also preserved. The sparsified signal is then subsampled below Nyquist by the IoT device which causes aliasing. To retrieve the wideband channel measurements from this aliased spectrum, we introduce a novel channel recovery algorithm that co-designs the MEMS hardware with the subsampling rate, and formulate a joint inverse problem that optimizes for the channel ToF's in the continuous domain to achieve super-resolution. Through extensive experiments in three large outdoor testbeds, we demonstrate that ISLA can improve localization accuracy by 4-11$\times$, and it achieves localization performance that is comparable to having a broadband 100 MHz receiver, despite using a narrowband IoT receiver at 16$\times$ lower sampling rates.

1.2 Contributions

In pursuit of building these next-generation millimeter-wave systems and delivering new applications, this dissertation draws on tools from diverse areas including networking, signal processing, deep learning, computer architecture and RF-acoustics microsystems. We build on a deep understanding of wireless signals and work across hardware-software boundaries to solve core problems in networking and sensing. Further, this dissertation takes an inter-disciplinary approach that couples core networking innovations with application domain specific knowledge. Specifically, this dissertation presents the first system design and protocol that could enable extreme dense spatial reuse in next-generation millimeter wave networks, to maximize the number of concurrently operating links. It also introduces the first suite of networking protocols for mmWave Wireless Networks-on-Chip that could learn and adapt to the highly dynamic traffic patterns of parallel applications. Lastly, this dissertation also contributes the first localization algorithm that allows narrowband IoT devices to accurately localize themselves using only ambient wideband 5G signals.

The work in this dissertation advances a broad array of capabilities promised by the next-generation of millimeter-wave systems, ranging from improving networking performance to deliver ultra-low latency and high-data rate communications, super-resolution in wireless localization, and transforming new application domains like high performance computing (HPC). This dissertation particularly focuses on the question of how to scale these technologies to more realistic deployments with multiple heterogenous nodes, while also extending the capabilities of these next-generation systems.

1.3 Organization

The rest of the dissertation delves deeper into the algorithms, techniques and implementation details of the systems described above. Chapter 2 describes BounceNet [3] and how it enables dense spatial reuse to allow multiple links to communicate simultaneously without interfering. Chapter 3 discusses NeuMAC [4], and how it leverages Deep Reinforcement Learning to generate new MAC protocols that allow the multicore processor to optimize for end-to-end execution of the parallel workload. Chapter 4 discusses ISLA [5, 6], and how it leverages recent advances in RF-acoustics microsystems to enable low power IoT devices to accurately self-localize by only listening to ambient 5G signals in the air. Finally, we conclude in Chapter 5 with a discussion of possible future research directions.

Chapter 2

MANY-TO-MANY BEAM ALIGNMENT FOR DENSE SPATIAL REUSE IN MILLIMETER-WAVE NETWORKS

2.1 Introduction

Millimeter wave (mmWave) is emerging as the de facto technology for next generation wireless networks [7, 8]. The abundance of bandwidth available in mmWave frequencies (above 24 GHz) led to the design of wireless radios that can operate at several Gbps [9, 10, 11], and the wireless industry is constantly pushing towards incorporating these radios in wireless products [12, 13, 14, 15, 16, 8]. Hence, mmWave will significantly change the future of wireless LANs by delivering links at fiber-like speed. This will allow wireless LANs to handle the surge in IoT and mobile devices. Furthermore, it will enable new applications like multi-user wireless VR for education, professional training, and multiplayer games, where high bandwidth data must be streamed to each user in real-time [17, 18, 19]. It will also enable large scale robotic factory automation where many robots stream continuous real-time video back to servers that run AI algorithms and generate decisions to coordinate the robots [20, 21].

Enabling the above vision, however, requires scaling mmWave networks from a single communication link to a network of many links without compromising the throughput of each user. Fortunately, mmWave radios use very directional steerable narrow beams to focus their power. This presents a significant new opportunity for exploiting dense spatial reuse to enable many links to simultaneously communicate at multi-Gbps data rates without interfering. Consider the example shown in Fig. 2.1. In the current broadcast model for 802.11 WLANs, whenever a node is transmitting, all other nodes must stay silent to avoid interference. With more users, the throughput is divided since the entire medium is shared. In contrast, the use of very narrow beams in mmWave networks allows several APs and clients to transmit and receive simultaneously on the same channel without interfering as shown in Fig. 2.1(b). Hence, mmWave can potentially scale the network throughput with the number of users by adding more APs.

The directional nature of communication, however, brings its own new challenges. Millimeter wave APs and clients need to align their narrow beams towards each other in order to communicate at very

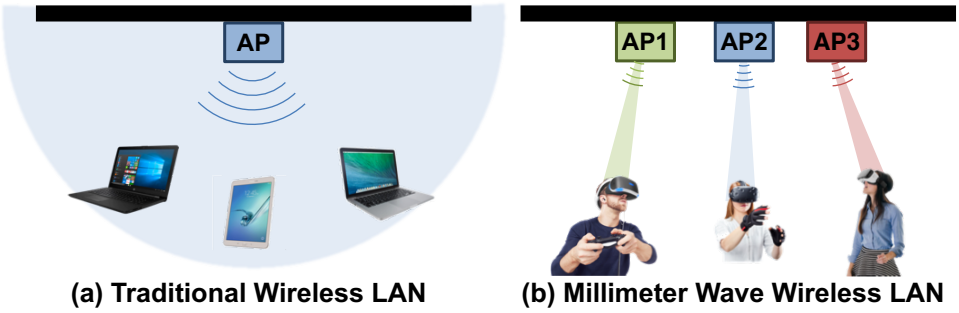


Figure 2.1: Spatial reuse in traditional WiFi vs mmWave networks.

high data rates. Past mmWave research focused on developing algorithms and protocols to quickly find the best direction to align the beams for a single communication link [18, 22, 23, 24, 25, 26]. However, in a network with multiple links, selfishly choosing the best alignment for each AP-client pair independent of other APs and clients can create interference that severely harms the throughput of interfering links. First, due to multipath reflections, even if two nodes are transmitting in completely different directions, their packets might still collide. The problem is further complicated by the fact that carrier sense is ineffective at detecting interference since the narrow beams prevent mmWave radios from hearing nearby transmissions unless these transmissions are specifically directed towards them. Hence, we can rely on neither carrier sense nor the direction in which the nodes transmit to avoid interference.

In this chapter, we introduce BounceNet, the first many-to-many millimeter wave beam alignment protocol that efficiently aligns the beams of many APs and clients in a manner that allows them to simultaneously communicate without interfering. To achieve this, we must address two key questions:

(1) How does BounceNet align the beams of all the APs and clients in 3D space to densely pack as many links as possible? The challenge arises from the fact that the choice of beam alignment at any node is intertwined with the choices at other APs and clients. To address this, BounceNet leverages the sparsity in the mmWave channel. There is much past work that shows that mmWave signals travel along a small number of paths, e.g., 2 or 3 paths [27, 28]. This means that there is a small number of paths connecting any two nodes in the network. BounceNet leverages this sparsity to reformulate the problem as a signal level routing problem at the physical layer where wireless signals are routed along different "air paths" in a manner that avoids interference and maximizes network throughput. Routing physical signals is possible in mmWave due to the lack of scattering effects at such high frequencies which ensures the signal reflects off obstacles and does not scatter in many directions [27]. Hence, BounceNet can choose to route the signal along an isolated path by aligning the narrow beam towards that path.

By choosing a combination of direct and reflected paths to route the wireless signals, BounceNet can

align the beams of all APs and clients in the network. While this allows it to maximize the number of links that can operate concurrently without interfering, it forces some APs and clients to communicate along reflected paths which typically achieve lower data rates. To address this issue, BounceNet generates several combinations of beam alignments and schedules them in different time slots; i.e., the transmissions of the links are routed along different paths in each time slot to ensure that each client gets high data rate while still maximizing the number of links that can operate simultaneously. BounceNet jointly solves the alignment and scheduling problems. We also model paths belonging to the same link as a supernode in a multilayer conflict graph and weight them by the SNR of the path. This ensures that paths which deliver higher data rates are used more often as we describe in detail in section 2.6.

(2) How does BounceNet quickly learn the paths and interference patterns in order to adapt the beam alignment in dynamic and mobile environments? In dynamic environments, the propagation paths and the interference patterns constantly change. Thus, we must periodically perform a beam search to learn the directions of the paths along which an AP and client can communicate.[1] BounceNet must also learn the propagation paths that can result in interference between two links and, hence, needs to perform the beam search between all APs and clients in the network to learn all the possible paths. Past work has shown how to leverage sparsity to quickly learn the paths without scanning all directions and reduce the search time to a millisecond [22, 23]. However, for a network of N APs and clients, this process must be performed $O(N^2)$ times. For $N = 10$, even with fast algorithms like [22, 23], the overhead is 100 ms which is prohibitively expensive especially at multi-Gbps data rates.

Instead of performing the search independently for all APs and clients, BounceNet redesigns the beam search protocol to jointly find all the paths between the nodes. BounceNet coordinates the APs' transmissions and then shares their measurements over the Ethernet which allows it to amortize the cost of the search and reduce it to $O(N)$. Since the beam search is inherent to mmWave and is required to maintain connectivity between clients and APs, BounceNet's design does not introduce additional overhead compared to current standards. This allows BounceNet to quickly learn the paths and reconfigure the beam alignment to maintain high throughput as we describe in detail in section 2.5.

Implementation & Results: We have designed BounceNet to be backward compatible with the current mmWave wireless LAN standard 802.11ad/ay making it easy to integrate into future standards. Our design also addresses several practical challenges like side-lobe leakage from imperfect beam patterns and interference estimation. We have implemented BounceNet by using extensive real measurements from three indoor wireless testbeds:

[1] Typically, the beam search is repeated every 100 ms in current standards like 802.11ad in order to track mobile users and maintain alignment.

- A 60 GHz testbed with 3° beam directional antennas.
- A 60 GHz testbed with 12° beam directional antennas.
- A 24 GHz testbed with 8-element phased arrays.

For a testbed with 10 APs and clients packed in an area of 860 sq.ft., our results show that BounceNet can scale the overall network data rate with the number of clients delivering over 39 Gbps for 10 clients. Furthermore, compared to the current 802.11ad standard that exploits spatial reuse, BounceNet can increase the average client throughput by 6.6×, 5×, and 3.1× for each of the above testbeds respectively. Compared to a baseline that aligns the beams of each link independent of other links, BounceNet increases the average client throughput by 1.27×, 2.7×, and 3.4× for each of the above testbeds respectively. BounceNet also improves the minimum data rate among all clients by up to 13.5× compared to the baseline which can create interference that severely harms some clients. Finally, Fig. 2.2 shows an example snapshot of a time slot where BounceNet exploits multipath to enable all 10 APs and clients, in the 60 GHz testbed with 12° beams, to communicate at the same time without interfering, hence demonstrating BounceNet's ability to enable extreme spatial reuse.

Contributions: We make the following contributions:

- We present the first many-to-many beam alignment protocol that can efficiently align the beams of a network of APs and clients to maximize the number of links that can operate concurrently.
- We demonstrate the opportunity of routing physical signals along different paths that bounce off the environment to improve the spatial reuse of the network. We harness this opportunity to design new algorithms that maximize network throughput while maintaining a lower bound of fairness for each client.
- We extensively evaluate our system through micro-benchmark measurements, trace-driven simulations, and experiments using 3 testbeds. Our results demonstrate the first design of a wireless LAN that can deliver more than 39 Gbps to 10 clients.

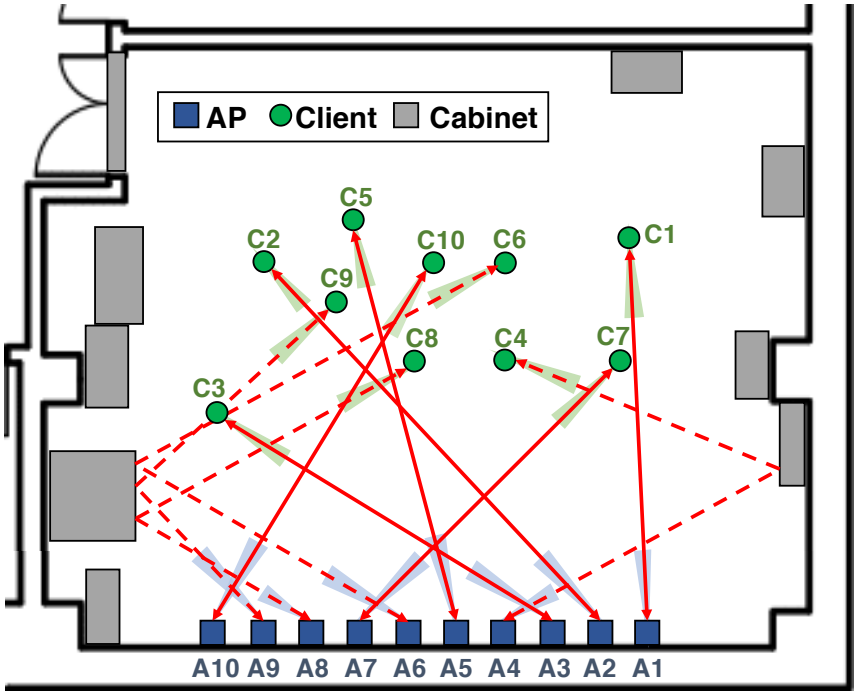


Figure 2.2: Example of BounceNet's signal routing in practice.

2.2 Related Work

Millimeter Wave Networks: BounceNet is related to recent work on increasing the speed and robustness of beam alignment in mmWave networks to enable mobility [22, 23, 24, 25, 26, 29, 30, 31, 32, 33] and avoid blockage [25, 34, 35, 36, 37, 18, 38]. All this work, however, focuses on a single communication link. BounceNet is the first to demonstrate many-to-many beam alignment. It is complementary to these systems and can benefit from faster beam search to discover the paths between nodes.

BounceNet also builds on past work in mmWave that uses 60 GHz wireless links in data centers [39, 40, 41] and leverages reflections off the ceiling to improve the throughput and avoid blockage [41]. Data centers, however, have static and known topologies with predictable interference models [41], and this does not hold in 802.11 LANs where the clients can move.

Our work is also related to recent mmWave work that deploys multiple APs to deal with blockage [42, 43]. [42] leverages multiple APs and allows clients to switch between them whenever blockage occurs in VR applications. However, it requires brute-force training to map all reflectors in the environment and relies on sensors in VR headsets to track the direction of users. [43] addresses blockage by having multiple APs jointly transmit the same signal to the clients. However, the method works only for downlink traffic and requires phase and frequency synchronization to ensure the signals sum up coherently. Achieving such level of synchronization is difficult and adds significant complexity to the design [44, 45]. BounceNet opts for a simpler design that scales the throughput of the network for both downlink and uplink traffic without requiring phase, frequency or packet level synchronization. It also learns the reflected paths in real-time.

Some recent simulation-based work for mmWave wireless PANs (Personal Area Networks) [46, 47, 48, 49, 50, 51] and mmWave mesh networks [52] tries to exploit spatial reuse. However, these solutions assume that the exact locations of the nodes are known a priori and can be used to compute the interference between links while ignoring multipath. BounceNet, on the other hand, designs and empirically tests a system that can work in the presence of multipath without prior assumptions of the clients' locations.

Finally, [53, 54] use MU-MIMO in mmWave and demonstrate concurrent transmissions to two clients from one MU-MIMO AP. BounceNet's beam alignment algorithm is complementary to MU-MIMO and can benefit from having APs that support MU-MIMO to further scale the gains.

Enterprise WiFi and WLANs with Directional Antennas: Past work has designed protocols for mobile ad-hoc networks and WLANs with directional antennas [55, 56, 57, 58]. However, past work can

support only large cone beams (e.g. 45° and 60° cones) at data rates of at most tens of Mbps. The scale of the problem is far more extreme in mmWave with narrow pencil beams of few degrees to sub-degree beamwidth at data rates of multi-Gbps. Hence, the overhead of past protocols can be prohibitively expensive in mmWave. Moreover, most of these protocols assume the locations of the nodes are known and ignore multipath [55, 56, 57].

The closest to our work is [58], which leverages directional phased arrays at 2.4 GHz to increase spatial reuse. However, [58] assumes only APs to have directional antennas which simplifies the problem since the clients can easily perform interference detection in the omnidirectional mode. Furthermore, the scheduling algorithm in [58] is exponential in the number of APs and hence is only shown to work for 3.

Past work had designed centralized scheduling algorithms for enterprise WiFi networks [59]. However, WiFi networks are omni-directional. Extending past algorithms to deal with directionality is non-trivial since the interference or conflict graph used for scheduling is itself dependent on the choices of beam alignment and there is a combinatorial number of choices as we discuss in section 2.5. BounceNet jointly solves the beam alignment and scheduling problems to deliver an efficient algorithm.

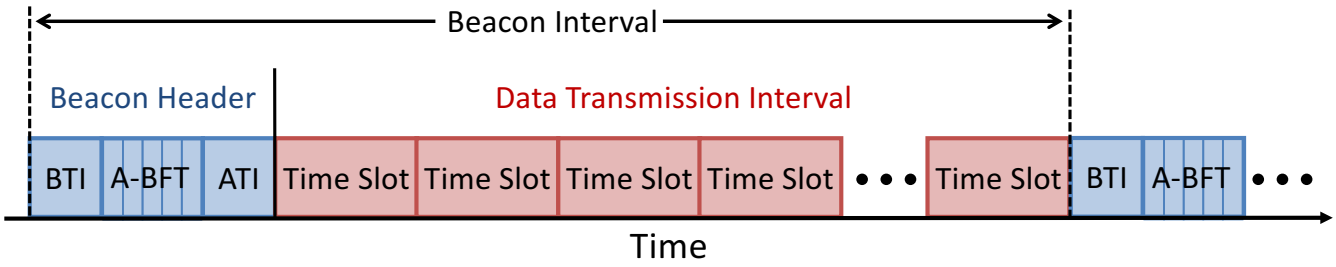


Figure 2.3: 802.11ad/ay Beacon Interval Structure.

2.3 Background

BounceNet is designed to be backward compatible with 802.11 millimeter wave standards for indoor wireless LANs. In this section, we provide a brief overview of the 802.11ad standard for 60 GHz networks [60, 1].[2]

The standards divide time into transmission cycles typically referred to as Beacon Intervals (BI) which consist of two phases, shown in Fig. 2.3. The first is the association phase which is referred to as the Beacon Header Interval (BHI). It is used to associate the clients with the AP and perform beam alignment. The second is the transmission phase which is referred to as Data Transmission Interval (DTI) where time slots are allocated for communication between the AP and associated clients. We will first describe these phases for the case of a single AP and multiple clients. We will then extend our description to multiple APs.

A. Association Phase:

The beacon header shown in Fig. 2.3 is used to associate the clients with the AP and perform beam alignment so that both the clients and the AP know which direction they should point their beam during data transmission.

The beacon header starts with a Beacon Transmission Interval (BTI) where the AP transmits announcement frames in all directions by sequentially sweeping its narrow beam along different sectors. During this time, the clients listen to the channel in all directions using a quasi-omnidirectional beam pattern so that they can receive packets from all paths. The announcement frames are marked with the sector ID along which they are sent allowing each client to discover the directions which the AP can use to send it data packets.

BTI is then followed by Association Beamform Training (A-BFT) which reverses the above operation. The AP uses a quasi-omnidirectional beam pattern so that it can hear clients from all directions while

[2]Note that another standard in the works is 802.11ay. However, it fully inherits the same PHY and MAC structure of 802.11ad. The main difference is the introduction of MIMO [61].

the clients sweep their narrow beam along different sectors. This allows the AP to discover the beam directions which the client can use to send its data packets and send it feedback to inform it of these directions. A-BFT is divided into multiple slots. Each client selects a random slot to perform its sweep. If two clients collide in an A-BFT slot, they will not get feedback from the AP and they can try again in another random slot.

The above process enables the AP and client to align their beams towards each other so that they can boost their SNR and use very high data rates for data transmission. However, during this association phase and before aligning their beams, the AP and clients use a control PHY with a low data rate of 27.5 Mbps to ensure the frames can be decoded correctly at low SNR. The beacon header finally ends with Announcement Transmission Interval (ATI), where the AP and associated clients exchange control frames such as information regarding time slots that have already been allocated to the client.

B. Transmission Phase:

The data transmission interval (DTI) is divided into time slots. The AP either uses TDMA to allocate each slot to a certain client or it allows the clients to contend for each time slot using CSMA. CSMA, however, does not work for directional networks [58, 56]. Hence, TDMA is more commonly used especially for video streaming applications where clients require dedicated slots in every beacon interval to ensure high quality and reliability.

For data transmission, the standard provides 32 different modulation and coding schemes (MCS) including single carrier modulation and OFDM modulation. Commercial products, however, adopt single carrier modulation due to the high power consumption of OFDM [62, 63]. Hence, in this chapter, we will focus on single carrier: MCS1 to MCS12 which provide data rates between 385 Mbps and 4.62 Gbps [60].

C. Multiple Access Points:

In the case of multiple APs, a lead AP is selected. The lead AP divides the beacon interval into smaller beacon intervals called beacon service periods (BSP). Each BSP has its own beacon header and data transmission period, and it is allocated to one AP. All other APs must stay silent during this service period. In order to enable spatial reuse, the lead AP can allocate a service period to two APs and request that they measure mutual interference and report back. If no interference occurs, it allocates the same service period to these APs in subsequent beacon intervals. Unfortunately, our results show that such a greedy mechanism for exploiting spatial reuse is unable to scale the network throughput with the number of clients.

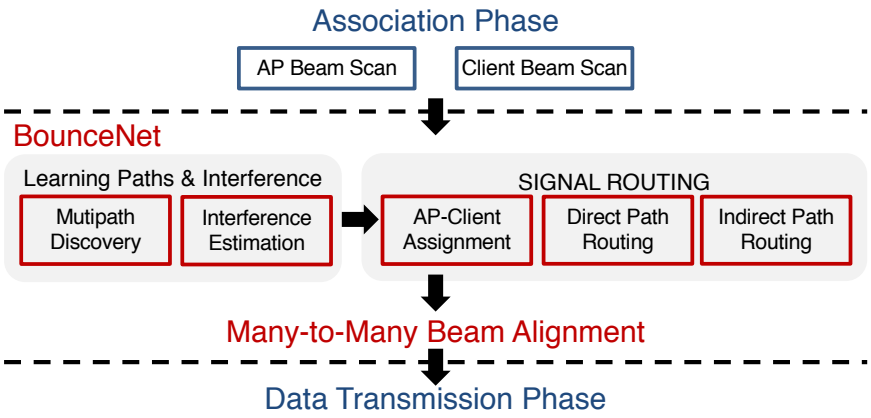


Figure 2.4: BounceNet's System Architecture.

2.4 BounceNet Overview

BounceNet's goal is to align the beams of all APs and clients in the network in a manner that maximizes spatial reuse. This allows WLANs to add additional APs to quickly scale their throughput with the number of clients.

We have designed the BounceNet protocol to support *independent flows*. This means that for an AP-client pair that is assigned to communicate along a path in a time slot, its link flow runs independently of other links for that time slot. The AP and client can transmit packets on the uplink or downlink without interfering with other links. The pair does not have to share any data packets or synchronize the individual packet transmissions with other APs or clients.

BounceNet is also backward compatible with 802.11ad/ay. It maintains the same high-level structure. BounceNet's architectural flow is shown in Fig. 3.4. It uses a controller that sits between the association phase and the data transmission phase of the protocol. BounceNet uses association phase to learn the paths and interference in the network and then runs its signal routing algorithm which dictates the many-to-many beam alignment in the data transmission phase.

BounceNet starts with an association phase similar to 802.11 where the APs and clients sweep their beams to collect information about the directions in which their signals can reach other APs and clients. This information is then aggregated at the APs, and fed to the BounceNet controller which allows it to discover all the paths connecting any two nodes in the network. We refer to this as multipath discovery (Section 2.5.1). BounceNet then uses the phased array beam patterns and the learned paths to estimate the interference created by routing signals along each path (Section 2.5.2).

BounceNet uses the results to route physical signals along propagation paths in a manner that maximizes the number of AP-clients pairs that can communicate simultaneously. Ideally, we would have liked to treat all APs as one large AP with many paths to all clients and find the optimal routing. However, this significantly increases the complexity of the problem and will require very fast handoff between APs to allow clients to switch APs within a beacon interval.[3] Hence, BounceNet assigns a single AP to each client for communication during the entire Beacon Interval.

To reduce the complexity of the system and ensure fairness, BounceNet performs signal routing in three stages:

- **Stage 1:** Associate each client to communicate with one AP for the duration of the entire beacon interval. (Section 2.6.1)
- **Stage 2:** Route the signal of each AP-client pair along their direct or highest throughput path in a manner that maximizes the number of links that can communicate in a given time slot without interfering. (Section 2.6.2)
- **Stage 3:** Route additional signals of AP-client pairs along their indirect paths to increase throughput without interfering with existing transmissions. (Section 2.6.3)

The above signal routing results in several beam alignments that are used for transmissions between APs and clients during each time slot of the data transmission phase. The entire process is repeated every beacon interval to adapt to changes in the environment and accommodate client mobility.

[3]Such fast handoffs are not feasible in mmWave networks because they require transferring the buffer at one AP to another AP at the time scale of few ms which would overwhelm the backhaul.

2.5 Learning Paths & Interference

BounceNet must first map all the paths between all nodes in the network and discover the potential interference between paths. Typically, for a network with N APs and N clients, this would require collecting $O(N^2)$ measurements. BounceNet instead redesigns the 802.11ad/ay protocol and exploits its beam alignment phase to extract all the paths from $O(N)$ measurements that are already part of the standard protocol.

2.5.1 Multipath Discovery

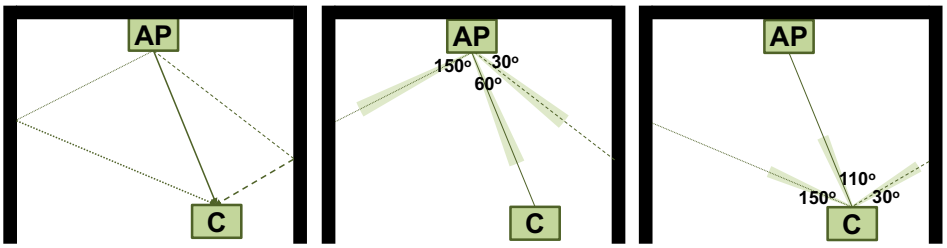
As described earlier, in case of multiple APs, the current standard divides the beacon interval into smaller beacon intervals and dedicates each interval to one AP. Instead, BounceNet aggregates them into one beacon interval with one beacon header and one data transmission interval. In particular, BounceNet only expands the BTI, shown in Fig. 2.3, to allow all APs to perform their beam scan of sequentially sweeping all sectors. While an AP is performing a sweep, all other clients and APs set their antenna to a quasi-omnidirectional mode and record the sector IDs of the frames they receive along with the SNR of the signals. A-BFT is then performed by assigning each client to a slot. While some client is performing its sweep, all other clients and APs set their beam to quasi-omnidirectional and record the sector IDs and SNRs of the frames received from the client. Algorithm 1 shows pseudocode for BounceNet's association phase.

The above process recovers a list of directions from which any node (AP or client) in the network can reach any other node. However, this might not be sufficient for discovering the paths between an AP and a client. Consider the example shown in Fig. 2.5(a) where there are three paths between an AP and a client. During BTI, we discover that the AP can reach the client by transmitting in one of three directions: 30°, 60° or 150° as shown in Fig. 2.5(b). During A-BFT, we discover that the client can reach the AP by transmitting in one of three directions: 30°, 110° or 150° as shown in Fig. 2.5(c). Unfortunately, since we do not know the position and orientation of the client, we do not know which direction at the AP corresponds to which direction at the client.

To address this, BounceNet needs to match the directions corresponding to the same paths by correlating the SNRs recorded from the client side and from the AP side. For instance, the directions corresponding to the direct path can be easily identified since typically the direct path delivers significantly higher SNR compared to indirect paths as we empirically show in Fig. 2.14(a) in section 2.8. However, in some cases, there could be two indirect paths that show similar SNR values (within 1 dB of each other). In

Algorithm 1 BounceNet Multipath Discovery

$N \leftarrow$ Number of APs
$\forall$ Clients $\rightarrow$ Set quasi-omnidirectional beam
$\forall$ APs $\rightarrow$ Set quasi-omnidirectional beam
Begin BTI:
for $m \in \{1, \cdots, N\}$ **do**
  AP(m) $\rightarrow$ Set directional beam
  for $\theta \in$ Sectors **do**
    AP(m) $\rightarrow$ Transmit frame in direction θ
    $\forall$ Clients & APs
    if Frame Received **then**
      Paths.AP(m)$\{\theta\} \leftarrow SNR$
  AP(m) $\rightarrow$ Set quasi-omnidirectional beam
Begin A-BFT:
Repeat the above process for clients.
Report Paths back to APs in transmitted frames.



(a) 3 Paths connecting AP and client

(b) Directions from AP side after BTI

(c) Directions from client after A-BFT

Figure 2.5: Multipath Discovery in BounceNet.

such situations, correlation might lead to erroneous matching due to the inherent noise in SNR measurements. Fortunately, though, as we show in section 2.8, the number of reflected paths between a pair of nodes in millimeter wave is quite small, e.g. 1 to 2 paths [27, 28]. Hence, at most, only two paths would remain ambiguous after the correlation step. BounceNet can then leverage the beam refinement option in 802.11ad which allows AP-client pairs to test pairwise directions to resolve such ambiguity. This incurs four more measurements. However, these measurements are taken while both AP and client beams are directional. Hence, they are transmitted at high data rate and incur negligible overhead.

2.5.2 Interference Estimation

Once we have discovered all the paths between the nodes in the network, we can estimate the interference caused by using any two paths simultaneously. BounceNet defines interference between paths as opposed to between nodes. If two paths interfere, then signals cannot be simultaneously routed along these two paths. We would like to keep the flows independent and avoid synchronization. Hence, at any point in time, both paths can be used to transmit uplink traffic, downlink traffic, or one path is used on the uplink while the other is used on the downlink. Consider a path between AP 1 and client 1 and another path between AP 2 and client 2 as shown in Fig. 2.6. Interference can occur in one of four cases: between AP 1 and AP 2, client 1 and client 2, AP 1 and client 2, or AP 2 and client 1 if there is a path connecting any of these pairs.

Formally, each path is defined by two angles corresponding to the direction from which it leaves one node and arrives at another node. We distinguish two types of paths:

- *Communications Paths:* defined as $(\theta_{APi}, \theta_{Ci})$ between AP 1 and client 1 as well as between AP 2 and client 2.
- *Interference Paths:* defined as (ϕ_{APi}, ϕ_{Cj}) between AP 1 and client 2 or AP 2 and client 1. They can also be defined as (ϕ_{APi}, ϕ_{APj}) or (ϕ_{Ci}, ϕ_{Cj}).

Ideally, it would be sufficient to check the directions of the paths to discover if interference occurs. Suppose AP 1 and client 1 can communicate along the path $(\theta_{AP1}, \theta_{C1})$ and AP 2 and client 2 communicate along the path $(\theta_{AP2}, \theta_{C2})$. In this case, for example, AP 2 will create interference at client 1 only if there exists an interference path (ϕ_{AP2}, ϕ_{C1}) where ϕ_{AP2} is in the direction of θ_{AP2} and ϕ_{C1} is in the direction of θ_{C1}. A similar rule can be used to detect interference between the other pairs.

Unfortunately, such a simple interference detection scheme will not work in practice. This is because the antenna beam patterns are not ideal cones. They have side lobes and can leak signal in other directions. Consider the example in Fig. 2.6, while AP 2 is transmitting in direction $\theta_{AP2} = 90°$, its signal might leak along another direction $\phi_{AP2} = 160°$ and reach client 1. To address this, BounceNet incorporates the phased array transmit and receive beam patterns into its interference estimation.[4] Specifically, to estimate interference between any pair of nodes, we consider all the interference paths between the two nodes and weight them by the beam pattern gains. Formally, when AP 2 directs its beam towards client 2 in the direction θ_{AP2}, it will have a beam pattern of $B_{\theta_{AP2}}(\phi)$. Similarly, when client 1 directs its beam towards AP 1 in the direction θ_{C1}, it will have a beam pattern of $B_{\theta_{C1}}(\phi)$. The interference created by AP 2 on

[4]Such patterns can be modeled or measured to account for imperfections in the mmWave phased arrays.

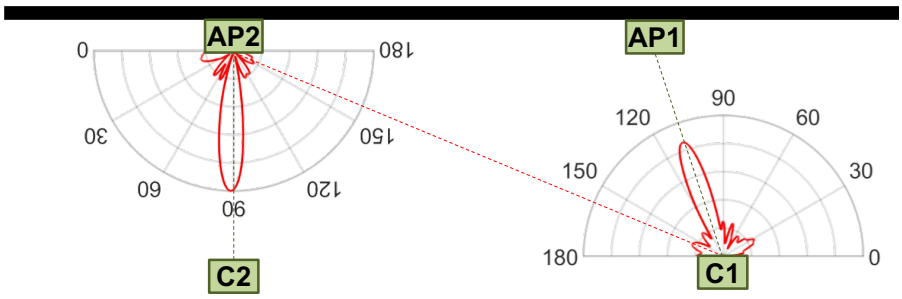


Figure 2.6: Estimating Interference using phased array beam patterns.

client 1 due to an interference path $P = (\phi_{AP2}, \phi_{C1})$ can be calculated as:

$$B_{\theta_{AP2}}(\phi_{AP2}) \cdot B_{\theta_{C1}}(\phi_{C1}) \cdot SNR(P)$$

where $SNR(P)$ is the normalized SNR[5] of the path P from AP 2 to client 1 measured during multipath discovery.

The maximum interference AP 2 causes can then be estimated as the constructive sum of leakage along all paths between AP 2 and client 1:

$$INR = \sum_{P=(\phi_{AP2},\phi_{C1})} B_{\theta_{AP2}}(\phi_{AP2}) \cdot B_{\theta_{C1}}(\phi_{C1}) \cdot SNR(P)$$

where INR is the interference-to-noise ratio. BounceNet repeats this estimation eight times: from AP 1 to AP 2 and client 2, from AP 2 to AP 1 and client 1, from client 1 to AP 2 and client 2 and from client 2 to AP 1 and client 1. BounceNet then defines the INR between the two communication paths as the maximum INR of all these 8 values.

Two points are worth noting:

- The above interference estimation does not assume to know the location or orientation of the APs or the clients. It also does not rely on knowing the room geometry or the use of ray tracing. It only requires the direction of the propagation paths (ϕ_1, ϕ_2) between nodes in the network and the associated signal strength along the paths.
- BounceNet is able to constantly maintain an up-to-date view of the multipath and interference pattern in the network since it obtains fresh measurements from the AP and client sweeps at the start of every

[5]The SNR is normalized by the antenna beam patterns used during the measurement of the SNR value in the multipath discovery phase.

Beacon Interval (which is approximately 100 ms). This feature allows BounceNet to deal with dynamic network conditions and accommodate for client mobility.

2.6 BounceNet's Signal Routing

Once BounceNet knows all the paths connecting the nodes and all the interference between the paths, it can route signals to/from clients in a manner that maximizes the number of AP-client pairs that can communicate in parallel. The choice of routing will govern the many-to-many beam alignment. BounceNet simplifies the problem by dividing it into three stages: *AP-Client Association, Direct Path Routing,* and *Indirect Path Routing*. We will elaborate on each stage below.

2.6.1 AP-Client Association

In the first stage, our goal is to associate each client to one AP for communication during the subsequent Data Transmission Phase of the Beacon Interval. Each client can associate with one AP, whereas each AP can serve multiple clients. Hence, for a network with N APs and N clients, we have N^N possible assignments. Trying all assignments is computationally infeasible. Thus, we develop an algorithm that sequentially assigns the clients to APs, with the objective of increasing throughput while minimizing the interference in the network. The intuition behind our algorithm is based on the following observations:

- In indoor settings, clients can typically achieve the highest data rate if they have a direct line-of-sight path to an AP. Hence, to ensure fairness, we should assign each client to an AP with a direct line-of-sight path.
- To maximize spatial reuse and throughput, we should avoid assigning multiple clients to the same AP unless the client cannot find any unassigned AP with a direct path.

Our algorithm works as follows. For each client, BounceNet keeps a list of best APs which have a direct path (high SNR path) to that client. BounceNet starts with the client with the least number of best APs and assigns it to one of the APs in its best AP list. It then adds this AP-Client pair to a list of already assigned links. For every subsequent client, BounceNet finds an AP from its best AP list such that: (1) the AP has not yet been assigned to a client, and (2) when communicating along their direct path, the AP-Client pair creates the minimum amount of interference on the direct paths of the already assigned links.[6] If no such AP exists, BounceNet simply picks the AP from the client's best AP list that creates the least interference.

The above algorithm is a best effort algorithm to assign each client to an AP with a direct path that creates the least amount of interference between the links. In the worst case, the best AP list of each client contains N APs. Then, while assigning the i^{th} client, BounceNet must compute the interference

[6]The amount of interference is estimated as the sum of the INRs computed in Section 2.5.2.

created by choosing one of the $N - i$ remaining APs on the i assigned links. Hence, the complexity is: $\sum_{i=1}^{N} (N - i)i = O(N^3)$. This reduces the complexity from exponential $O(N^N)$ to polynomial $O(N^3)$.

2.6.2 Direct Path Routing

Once each client is assigned to an AP, we will have N unique direct paths. BounceNet starts by routing signals to/from clients along these direct paths. Decoupling the signal routing along the direct and reflected paths simplifies the problem and allows us to ensure fairness among links when it comes to routing signals through their highest throughput paths, i.e. their direct paths. In the next section, we will show how BounceNet routes additional signals along indirect paths to enhance throughput.

A. Scheduling of Direct Paths

BounceNet uses graphs to solve the problem. It starts by building the *Direct Path Conflict Graph*: $G(V, E)$. V represents the set of vertices in the graph. Each vertex v corresponds to a direct path between an AP-client pair. E represents the set of edges in the graph. An edge $e_{u,v}$ exists between vertices u and v if the corresponding paths interfere. We use the estimation from section 2.5.2 to compute the interference between paths, and if the $INR > 0$ dB, we assign the paths as interfering.

In each time slot, BounceNet's goal is to schedule routing signals along as many paths as possible. Traditionally scheduling is modeled and solved as a minimum graph coloring problem on the conflict graph [64, 65, 66, 67]. This finds the minimum number of colors required to color the graph such that no two vertices connected by an edge share the same color. Thus, paths corresponding to vertices of the same color can be scheduled and used concurrently in the same time slot. This will minimize the number of time slots needed to schedule the paths while ensuring that each path gets one time slot to route signal to/from the client. Fig. 2.7(a) shows a possible minimum coloring of a graph which requires 3 colors. This means that we can schedule all paths within 3 time slots as shown in Fig. 2.7(b). Since there are 6 paths, this will give $2\times$ higher throughput than a scheduling which does not utilize spatial reuse and routes signals only along one path at any point in time.

B. Fairness in Millimeter Wave Networks

The above formulation can leverage spatial reuse to increase throughput while ensuring that each client gets an equal share of the time on the channel. This notion of fairness, however, is suboptimal in mmWave networks and needlessly wastes throughput. Due to the use of very directional beams in mmWave networks, the medium is no longer *"equally"* shared among all clients. Consider the example

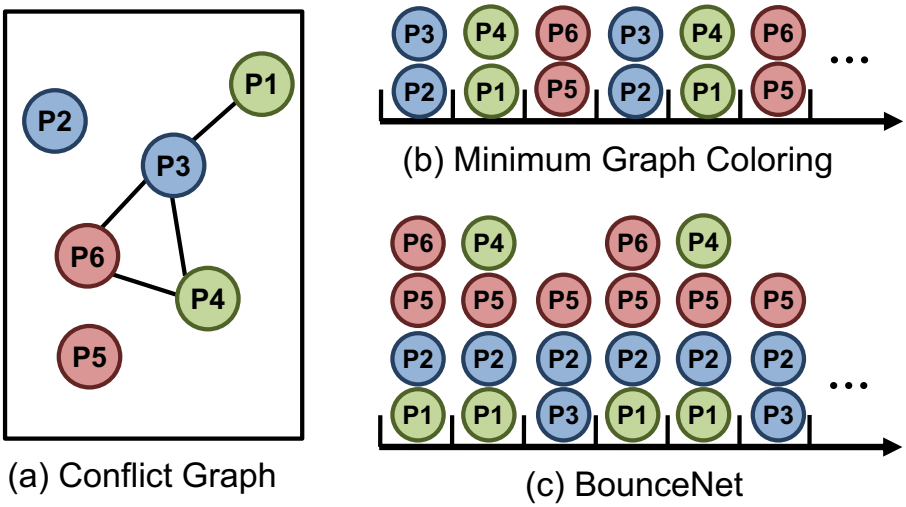


Figure 2.7: Scheduling of Direct Paths.

in Fig. 2.7(a). Paths 2 and 5 do not interfere with any other path and hence we should route signals through these paths in every time slot. Not doing so would reduce the throughput without benefiting anyone in the network. On the other hand, paths 4 and 6 share their medium with two other paths since they interfere with two other paths. Hence, a path should get a share of the medium which is at least a fraction of the number of paths it shares its medium with. For example, we should route signals through paths 4 and 6 in $1/3$ of the time slots, whereas we should route signals through paths 2 and 5 in all time slots since they interfere with no one.

Formally, if a path interferes with d other paths, it shares its medium with these d paths and hence should get a share of at least $1/(d+1)$. In the conflict graph G, d will correspond to the degree of the vertex, i.e. the number of edges that the vertex has. Using this new notion of fairness, we develop an algorithm to route signals through direct paths in a manner that achieves higher throughput while maintaining fairness.

C. BounceNet's Algorithm

BounceNet starts by trying to maximize the number of paths that can be used in each time slot. Maximizing the number of paths is theoretically equivalent to solving a maximum independent set problem. The maximum independent set refers to the maximum number of vertices that do not share any edges. For example, in Fig. 2.7(a), the maximum independent set can be formed of paths 1, 2, 4, and 5 since none of these paths share edges, i.e. none of them interfere. Routing signals through these paths in every time slot will achieve the highest possible throughput. However, it will result in starvation of some clients

Algorithm 2 BounceNet Scheduling of Direct Paths

$G(V, E) \leftarrow$ Direct Path Conflict Graph
$M \leftarrow$ Number of time slots in beacon interval
$F_1(u) = M \ \forall u \in V$
for $t \in \{1, \cdots, M\}$ **do**
    $W_t \leftarrow$ WEIGHTEDMAXINDEPENDENTSET(G, F_t)
    for $u \in W_t$ **do**
        if $F_t(u) > 2(d(u) + 1)$ **then**
            $F_{t+1}(u) = F_t(u) - (d(u) + 1)$
        else
            $F_{t+1}(u) = 0$

whose paths are never included in the maximum independent set, e.g. Path 3 in Fig. 2.7(a).

Instead, BounceNet uses a variant of the same problem referred to as the Weighted Maximum Independent Set. The idea is to give each vertex u a weight $F(u) \geq 0$. We then find the set of vertices W that maximize the sum of weights such that no two vertices in W share an edge. More formally, we find the set W that satisfies:

$$\text{maximize} \sum_{u \in W} F(u) \text{ such that } \forall u, v \in W, \ e_{u,v} \notin E \tag{2.1}$$

BounceNet solves the above optimization problem for every time slot and schedules to route paths corresponding to the vertices in W to each of the time slots. After each time slot, BounceNet decrements the weights of each of the vertices in W by an amount proportional to the interference it creates in the network, i.e. the degree of the vertex d. Hence, if we initialize all the weights equally, then for the first time slot, BounceNet will pick a Maximum Independent Set. However, as the algorithm proceeds, the weights of the scheduled paths keep getting decremented, and eventually paths that interfere with the paths in the Maximum Independent Set start to get picked in W, and in turn get scheduled.

Pseudocode of this algorithm is shown in Algorithm 2. Fig. 2.7(c) shows an example of the output of BounceNet's direct path routing. In this example, BounceNet's algorithm achieves 3.66× higher throughput while ensuring fairness, i.e. each path gets scheduled at least $1/(d + 1)$ of the time.

D. Analysis

If BounceNet wishes to schedule the nodes into M slots, it initializes all the weights to M. Then, every time a vertex u is picked, its weight is decremented by $d(u) + 1$ where $d(u)$ is the degree of this vertex. After this vertex has been picked up $M/(d(u) + 1)$ times, its weight becomes 0. Once the weight of a vertex becomes zero, its inclusion in W can no longer help maximize the sum of weights, and hence it

does not get picked up (or in our context, the path is no longer used) after that. However, by the time the weight of the vertex reaches 0, it has already been scheduled in $1/(d(u)+1)$ of the time slots and hence fairness is achieved. For example, if a vertex has degree $d = 0$, i.e. it does not interfere with anyone, it will be picked up every time since it will always help maximize the sum of weights. Every time it is picked, its weight is decremented by 1. Its weight will reach 0 only after it has been scheduled M times which means it has been scheduled in all time slots. In Appendix A, we prove the following lemma:

Lemma 2.1 *If* $t = O(M \log(NM))$*, then* $F_t(u) = 0 \; \forall u \in V$

Algorithm 2, however, requires solving a Weighted Maximum Independent Set problem which is NP-hard [68]. This would require an exponential time algorithm to find the optimal solution, which would be infeasible for any real-time implementation. We use the approximation algorithm from [68] to solve this problem. Empirically we find that the algorithm is at most two timeslots worse than optimal. However, in many cases, the algorithm achieves the optimal. This is because the sparsity renders the *Direct Path Conflict Graphs* in mmWave networks as chordal with very high probability. Chordal graphs are graphs in which all cycles of four or more vertices have a *chord*. For such graphs, [68] is optimal.

2.6.3 Indirect Path Routing

In this section, we will show how BounceNet will route additional signals along indirect multipath routes to increase the throughput without creating interference to signals being routed along the direct path.

BounceNet's indirect path routing is best understood through an example. Let us consider the direct path scheduling result shown in Fig. 2.7(c). During the first time slot, paths 1, 2, 5 and 6 were scheduled. Hence, clients 1, 2, 5 and 6 can communicate on their direct paths during this time slot. Note that a client can route its signal through only one path during any time slot. As a result, we only need to consider whether we can route signals through multipath for clients 3 and 4.

To this end, BounceNet forms an *Indirect Path Conflict Graph*. This graph includes vertices corresponding to the direct paths that have been scheduled as well as vertices corresponding to indirect paths of AP-client pairs that have not been scheduled in this time slot. Fig. 2.8(a) shows an example of this graph where client 3 has two indirect paths to its AP and client 4 has three indirect paths to its AP. Indirect path vertices corresponding to the same client are always in conflict since the client can use only one of those indirect paths. Hence, vertices corresponding to indirect paths of the same client form a fully connected subgraph which we will refer to as a *supernode*. We then estimate the interference that the indirect paths can create on direct paths that are already scheduled as well as other indirect paths.

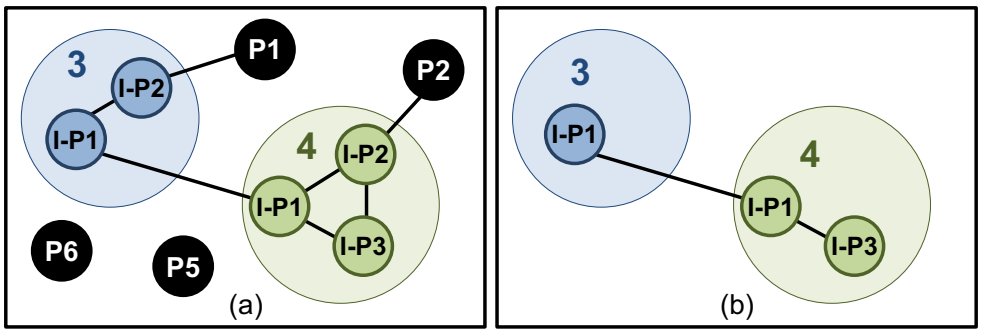


Figure 2.8: Indirect Path Conflict Graph before & after pruning.

Direct paths have already been scheduled and, hence, they are locked. Any indirect path that interferes with the direct path cannot be used in this time slot and hence can be eliminated from the indirect path conflict graph. Thus, BounceNet prunes the graph by removing all vertices that interfere with direct paths as well as vertices corresponding to direct paths as shown in Fig. 2.8(b). The resulting graph is typically much smaller and formed only of supernodes and vertices corresponding to indirect paths. BounceNet can route signals through any of the remaining indirect paths without interfering with signals being routed through the direct paths.

In order to schedule indirect paths, BounceNet uses the same algorithm as before where it maximizes throughput by solving a maximum weighted independent set problem on the *Indirect Path Conflict Graph*. However, BounceNet has to take into account two key differences:

- Unlike direct paths where there is small variance in SNR, the SNR of indirect paths can vary significantly as we will show in section 2.8. Hence, BounceNet should give indirect paths with higher SNR more weight. To do so, BounceNet gives each *supernode* a weight of M and divides this weight among its indirect path vertices in a manner proportional to the data rate that each indirect path can achieve. For example, if *supernode* 4 in Fig. 2.8 has indirect paths with SNRs 3 dB, 5 dB, and 7 dB, then it can deliver data rates of around 1.1 Gbps, 1.9 Gbps, and 2.5 Gbps respectively. Hence, its indirect paths will be weighted as $0.2M, 0.35M,$ and $0.45M$. This ensures that the higher data rate paths have a higher chance of getting picked.
- The degree d of a vertex no longer corresponds to the number of other clients it shares the medium with since vertices of the same *supernode* belong to the same client. Hence, instead of decrementing the weight of the node by $d + 1$, we decrement it by $d - s + 1$ where s is the number of other vertices that remain in the supernode after pruning the graph. For example, in Fig. 2.8(b) the indirect path in *supernode* 3 has $s = 0$ whereas in *supernode* 4 have $s = 1$.

Figure 2.9: **Indoor Experimental Space:** (a) Lecture Hall (b) Atrium (c) Lounge (d) Empty Room (e) Lab (f) Office Space.

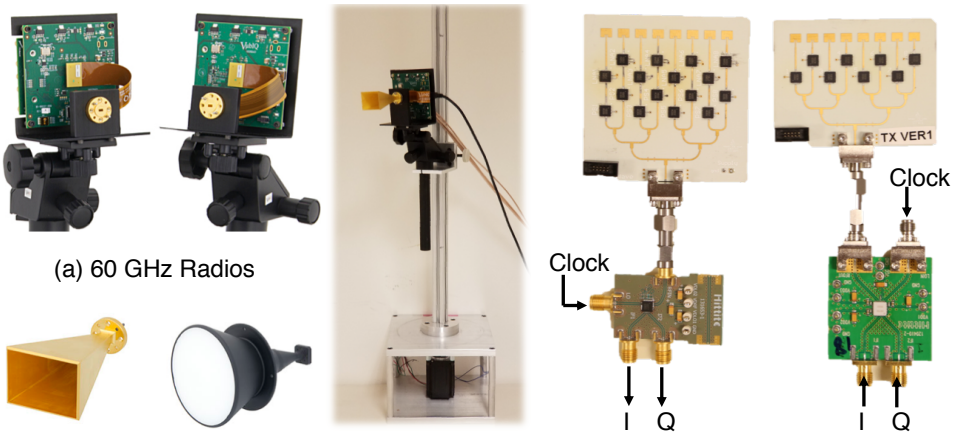


Figure 2.10: Experimental hardware used to evaluate BounceNet.

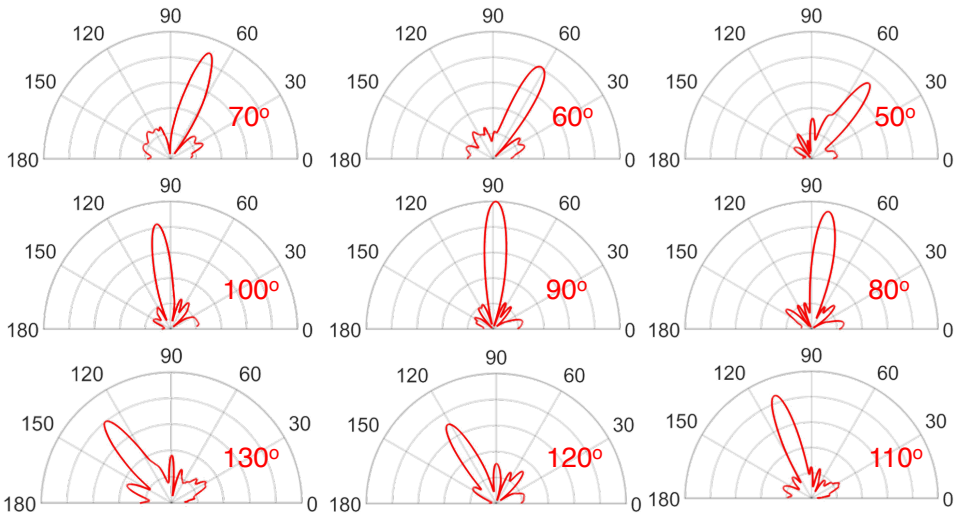
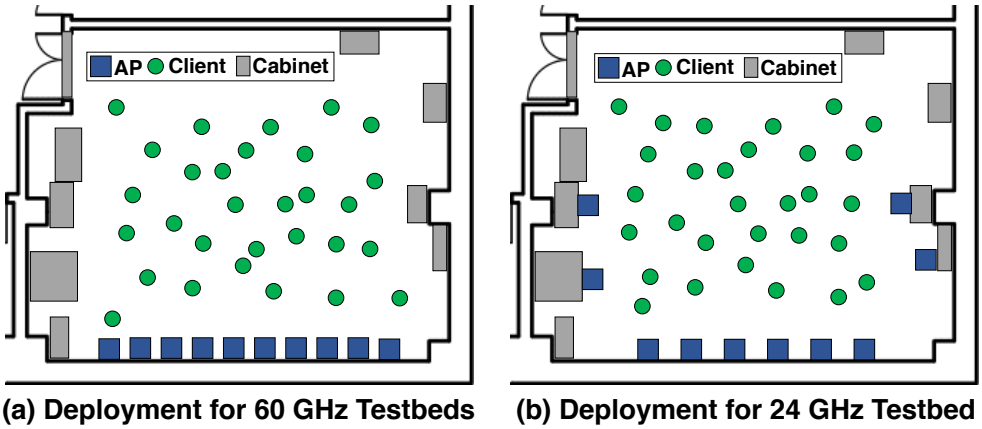


Figure 2.11: Example beam patterns of the 24 GHz phased arrays.



(a) Deployment for 60 GHz Testbeds (b) Deployment for 24 GHz Testbed

Figure 2.12: Placement of APs in the 60 GHz and 24 GHz testbeds.

2.7 Testbed and Implementation

We evaluated BounceNet using three indoor testbeds that operate at 60 GHz and 24 GHz. The 60 GHz testbeds used Pasternack PEM009 radios [69] shown in Fig 2.10(a). One testbed is equipped with directional antennas with beamwidth 3° and the other with 12° antennas shown in Fig. 2.10(b). The 60 GHz Pasternack modules are connected to USRP software defined radios through a Balun circuit to sample the signal. They are also mounted on a steerable platform shown in Fig 2.10(c) controlled through an Arduino.

The 24 GHz testbed used two radios, each equipped with an 8-element phased array shown in Fig. 2.10(d). The radios use HMC815B and HMC977 IQ up/down converters from Analog Devices which operate between 21 GHz and 27 GHz with 3.75 GHz of bandwidth. The integrated boards shown in Fig. 2.10(d) also include RF amplifiers and a frequency doubler. The boards are fed a clock in the range 10.5 GHz to 14.5 GHz from a TI LMX2594 PLL which is doubled to the 24 GHz range. The I and Q signals are connected to USRP software defined radios where the signals are collected. Fig. 2.11 shows examples of the beam patterns of the phased array that we obtain from our own empirical measurements. Note that while the beam patterns from some commercial phased arrays have much larger side lobes, we are able to achieve beam patterns as shown in Fig. 2.11 by leveraging the online algorithm for phased array calibration presented in [70].

We use the Tektronix DPS77004SX oscilloscope which samples at 200 GS/s and has a bandwidth of 70 GHz to calibrate the transmitted power of both 60 GHz and 24 GHz radios to match FCC regulations. We also use it to calibrate the measured power and noise floor of the USRPs.

Due to the large overhead of real-time processing and the limited bandwidth of USRPs, we use the software radios to measure interference and signal-to-noise ratio, which we map to the minimum achievable data rate using the receiver sensitivity table of 802.11ad [60] with 1% packet loss rate. We then used these testbed measurements to run trace-driven simulations using an 802.11ad ns3 library that takes phased array beam patterns into account [71]. We also modified this library to implement BounceNet. We then empirically verified the results by testing the interference and making sure any pair of paths used in a given time slot does not interfere. We then report the data rates per client as well as the overall network data rate. Finally, we also study the impact of our system when integrated with higher layer protocols like TCP and UDP and report application level throughput results.

We collected measurements in different rooms in order to evaluate the level of multipath and verify that BounceNet can exploit this multipath to maximize the number of links. We tested in six different types of rooms shown in Fig. 2.9: a lecture hall, an atrium, a lounge, a completely empty room, a lab space, and an office space. The full BounceNet protocol was evaluated in the lab which is 860 sq.ft. of space. The APs were deployed along the walls of the lab with the clients scattered across the room as shown in Fig. 4.3. We vary the number of APs and clients from 1 to 10. In every run, the clients are assigned randomly to these locations. We tested 5000 different configurations of locations. To emulate mobility, we move the clients in 5 cm steps along a path where we run scans and collect measurements for each step in the path.

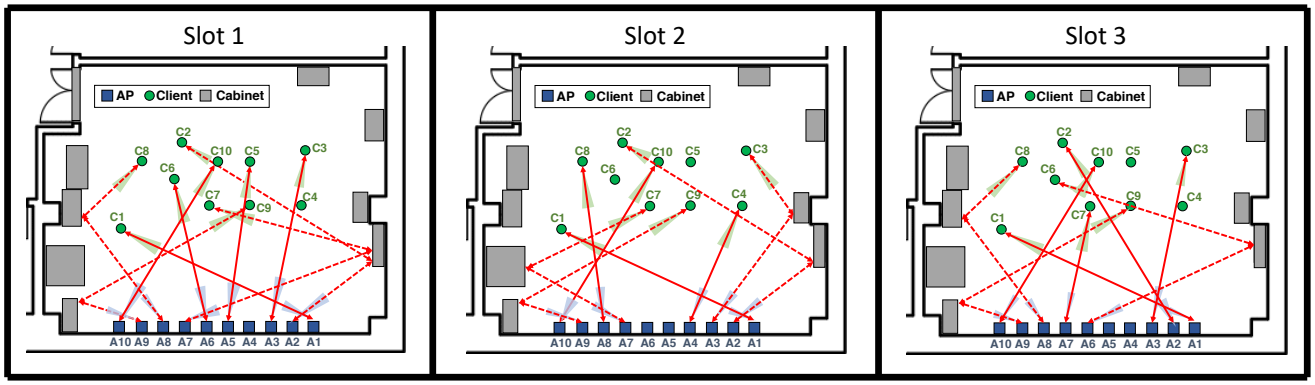


Figure 2.13: Beam Alignments computed by BounceNet for 12° beam testbed.

2.8 Microbenchmark Results

We start our evaluation with a few microbenchmarks that provide insights into the working of the system as well as the characteristics of mmWave networks before we present the evaluation results.

A. Multipath in mmWave Networks:

BounceNet leverages multipath in mmWave networks to maximize the number of links that can operate at the same time. Table 2.1 shows the distribution of the number of reflected multipath per link in each of the six rooms shown in Fig. 2.9. The results show that for all rooms except the atrium, in about 80% of the cases the client has 1 to 2 reflected paths through which it can route its signal to the AP. This is expected as the atrium is a large open space with limited reflectors. The results also show that very few clients see 3 or 4 indirect paths due to sparsity in mmWave.

Fig. 2.14(a) shows the CDF of the SNRs of the direct and reflected paths respectively measured from our testbeds. We observe that direct paths always provide sufficient SNR to support the highest data rate of 4.62 Gbps. The variation in direct path SNRs is small and the median SNR of direct paths is 15 dB larger than the median SNR of reflected paths which motivates BounceNet's design to split routing signals along direct and indirect paths into two stages. Furthermore, the SNRs of indirect paths can vary between 5 dB to 20 dB and hence it is important to take the SNR of indirect paths into account when deciding which indirect path to route signals through as we have described in section 2.6.3.

B. Accuracy of Interference Estimation:

Here, we evaluate the accuracy of BounceNet's ability to correctly estimate interference. We choose 100 different pairs of links from our testbed and measure the ground truth interference between every pair. For each pair, we consider both the direct path and indirect paths. To obtain the interference estimates

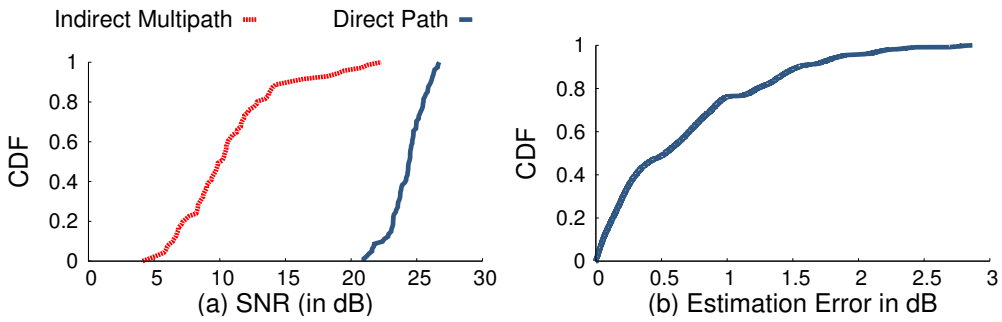


Figure 2.14: **Microbenchmarks:** (a) SNR of indirect vs. direct paths. (b) Interference estimation error.

Table 2.1: Percentage of Links with n Reflected Paths

Room	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$
Lecture Hall	0	20	46.6	26.6	6.6
Atrium	5	95	0	0	0
Lounge	0	46.6	50	3.3	0
Empty Room	0	21.0	52.6	26.4	0
Lab	0	37.4	41.4	21.2	0
Office Space	0	30	45	15	5

from BounceNet, we perform the association phase using the experimental setup. Then, we use the measurements to find all the paths and compute the INR as described in section 2.5.2. Fig. 2.14(b) shows the CDF of the absolute error between the ground truth interference measurements and the estimated values from BounceNet. BounceNet's median error is 0.52 dB and 90^{th} percentile error is 1.54 dB which is within the 3 dB tolerance for various mmWave MCSs. BounceNet is able to achieve such high accuracy in predicting the interference in the network because it accounts for both the multipath in the environment as well as the imperfections in antenna beam patterns. Furthermore, it is able to do this using only a linear number of measurements $O(N)$, therefore avoiding the need to explicitly measure interference between every pair which would be $O(N^2)$.

C. BounceNet's Signal Routing

In Fig. 2.13, we present additional examples of BounceNet's beam alignments in the 12° testbed. We pick one client configuration and plot the beam alignments computed by BounceNet for the first three time slots. We can see that BounceNet makes use of both direct and reflected paths in order to squeeze in as many links as possible for communication during the time slot. Furthermore, over the three time slots, BounceNet schedules the direct paths for different clients, thus clients get a chance to use their direct paths in different time slots. Clients that create less interference such as C1 and C10 get to use

their direct paths in all time slots whereas clients that create more interference such as C2 or C7 get to use it once.

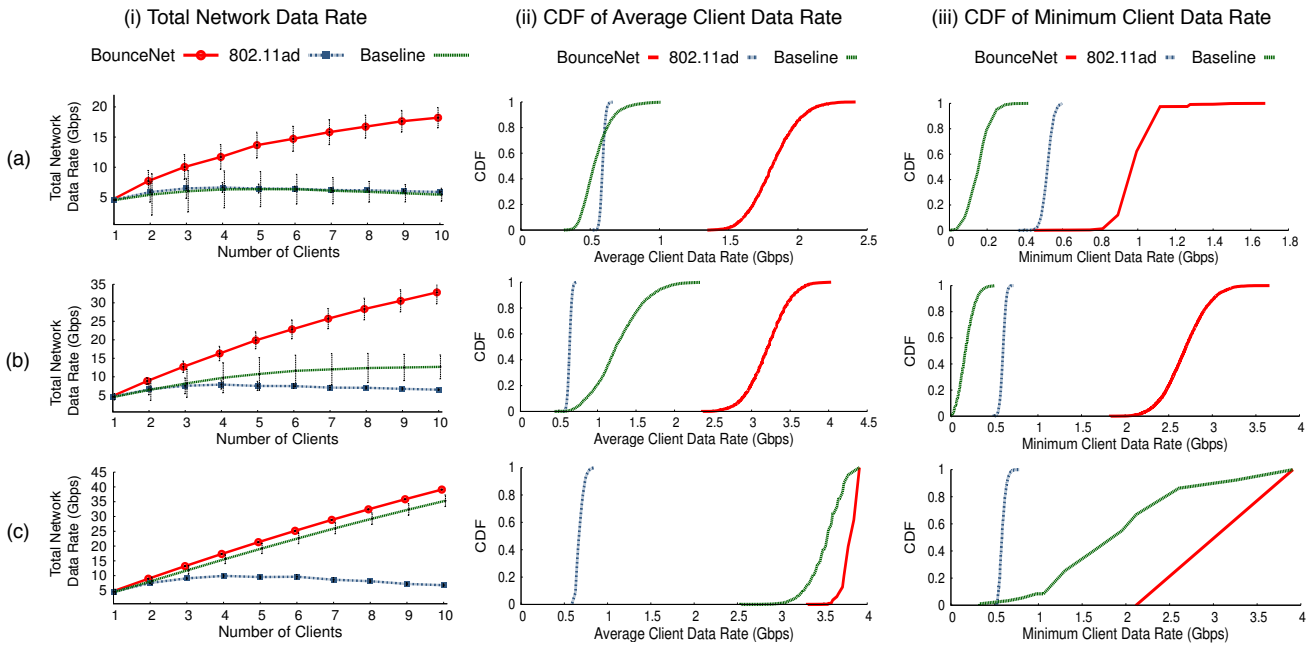


Figure 2.15: Data rates in BounceNet, 802.11ad and baseline for (a) 24 GHz phased array (b) 60 GHz with 12° beams (c) 60 GHz with 3°.

2.9 Evaluation Results

We will present our main evaluation results here. We will start by describing our baselines and evaluation metrics.

A. Compare Schemes: We compare BounceNet to:

(1) **802.11ad with Spatial Reuse:** As described in section 2.3, the current standard provides a greedy mechanism for exploiting spatial reuse by measuring pairwise mutual interference and merging links that do not interfere into the same slots. If the nodes detect changes in the interference in the network, they reset to transmitting in exclusive time slots.

(2) **Baseline:** Our baseline will consider independently aligning the beams of each AP and client and letting them transmit. To give the baseline an edge, we assume that the APs and clients can perform their beam search without creating any interference. Hence, they can find the right alignment in $O(N)$ and then use it for data transmission.

B. Metrics: We evaluate BounceNet using these metrics:

- **Total Network Data Rate:** The aggregate data rate of all the clients in the network.
- **Average Client Data Rate:** The average data rate of the clients in the network.
- **Minimum Client Data Rate:** The minimum data rate among all clients in the network.
- **Fraction of Time on the Channel:** The fraction of time slots a client gets to transmit in; used to evaluate fairness.
- **Average Client Throughput:** The average application layer throughput of a client using TCP or UDP flows.

C. BounceNet Data Rate Gain:

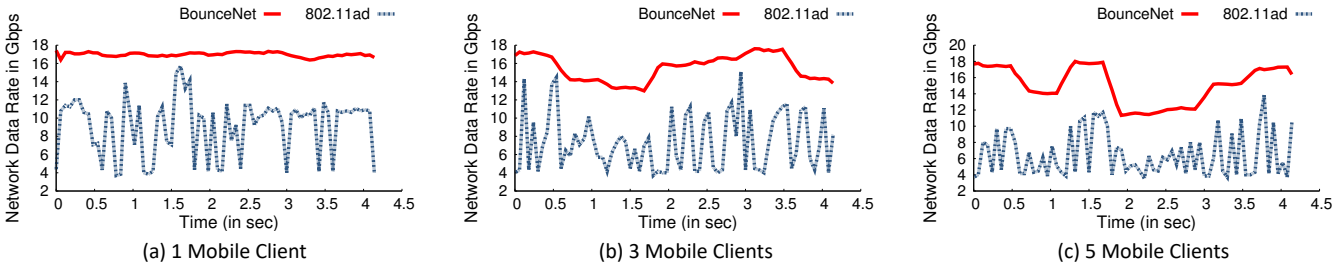


Figure 2.16: Mobility: This figure shows that BounceNet can adapt to changing and mobile clients whereas 802.11ad is unable to exploit spatial reuse in mobile networks.

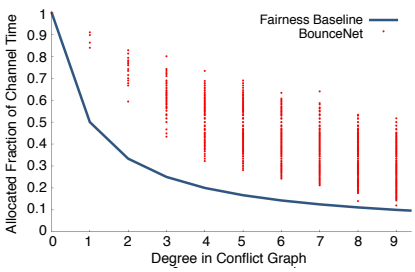


Figure 2.17: Client's share of time on the channel.

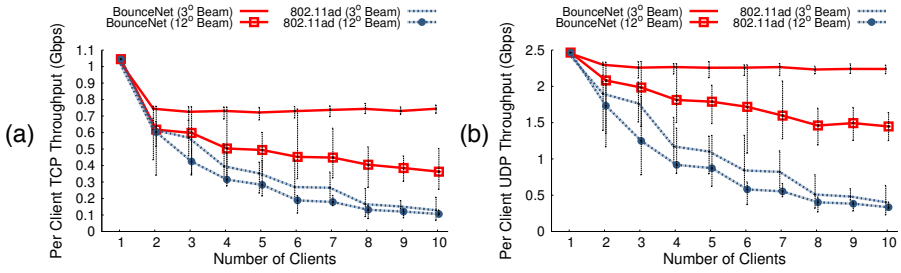


Figure 2.18: BounceNet's Application Level Average Throughput Under (a) TCP and (b) UDP.

We start by evaluating the gains in total network data rates. Fig. 2.15(i) shows the total network data rate as a function of the number of clients in a network with 10 APs for BounceNet, 802.11ad, and the baseline. As the number of clients increases, BounceNet is able to scale the total network data rate with the number of clients to deliver a total of 39.2 Gbps and 32.8 Gbps data rates for 10 clients using 60 GHz with 3° and 12° beams respectively. For 24 GHz, BounceNet is able to achieve 18.2 Gbps for 10 clients.

This is expected, as sidelobe leakage of phased arrays creates more interference in the network which limits spatial reuse.

802.11ad, on the other hand, is unable to properly exploit spatial reuse and shows limited gains. Specifically, for the case of 10 clients, BounceNet achieves $6.6\times$, $5\times$, and $3.1\times$ gain in network throughput as compared to 802.11ad for 3° beam, 12° beam, and the phased array respectively. This is due to 802.11ad's inefficiency which stems from requiring pairs of links to measure mutual interference during data transmission and merge these links during the following beacon interval only if they do not interfere. The baseline can exploit spatial reuse for 3° beam since the interference in this case is very limited. Hence, for 10 clients with 3° beam, BounceNet only achieves $1.27\times$ gain over the baseline. This gain, however, increases to $2.7\times$ and $3.4\times$ for 12° beam and the phased array respectively where there is more interference. In fact, the baseline is unable to exploit spatial reuse and scale network throughput in such cases.

In Fig. 2.15(ii) we plot the CDF of the average data rate achieved by the clients across all the runs with 10 clients in the network. A client in BounceNet can achieve a 50^{th} percentile average data rate of 3.8 Gbps for 3° beam, 3.25 Gbps for 12° beam, and 1.81 Gbps for the phased array. Whereas in 802.11ad, the 50^{th} percentile average data rate is around 0.6 Gbps in all three cases. The baseline, however, shows high average data rate of 3.4 Gbps for 3° beam which decreases to 1.26 Gbps for 12° and 0.5 Gbps for the phased array. Hence, with wider beams, simply ignoring interference would result in an even worse performance than 802.11ad.

Two points are worth noting. First, each of the 10 clients in BounceNet can achieve a 90^{th} percentile average data rate of 3.9 Gbps for 3°, 3.7 Gbps for 12°, and 2 Gbps for the phased array. This is a small deviation from the median data rate which shows that BounceNet is fair in dividing the rate across the clients. Second, while BounceNet scales the network throughput, the overhead of beam alignment starts to kick in. This, however, can be addressed by employing faster beam alignment protocols [22, 23, 26].

We also plot the CDF of the minimum data rate among all clients in Fig. 2.15(iii), across all the runs with 10 clients in the network. The figure shows that BounceNet can significantly improve the minimum and benefit worst case clients which can suffer from interference. BounceNet can improve the minimum data rate of any client in the network by $13.5\times$ for 12° beam and $7.5\times$ for phased arrays as compared to the baseline. This is because the baseline does not try to avoid interference, and hence clients that suffer from interference can really benefit from BounceNet.

In Appendix B, we present additional results when there are only 5 APs in the network. This allows us to evaluate BounceNet in scenarios where clients outnumber the APs.

D. Adapting to Changes and Mobile Clients:

To understand BounceNet's ability to adapt to mobile clients, we examine what happens to the total network data rate as clients move for both BounceNet and 802.11ad. As the baseline does not actively try to optimize for spatial reuse, we expect the total network data rate to remain smooth albeit lower than BounceNet.

We run an experiment where there are five clients in the network and we vary the number of clients that are moving. Fig. 2.16 shows the total network data rate versus time, when one client, three clients or five clients are moving. This figure shows that BounceNet can continue to maintain a high data rate as the clients move. For one client moving, BounceNet achieves almost a constant data rate. As more clients move, the interference patterns in the network change, and, hence, the maximum achievable data rate changes. The figure shows that BounceNet can quickly adapt to changes and continue to exploit spatial reuse.

On the other hand, the data rate in 802.11ad fluctuates significantly and keeps falling back to the case of no spatial reuse. This is because 802.11ad merges AP-client pairs only after measuring the mutual interference during the data transmission phase. Hence, it takes 802.11ad several beacon intervals (≈ 100ms) to exploit spatial reuse. By that time, the client has moved and the interference patterns have changed. Even if one client moves, it can affect the interference patterns of many links. Fig. 2.16 shows that as more clients move, the interference patterns change faster, and hence 802.11ad is unable to properly exploit spatial reuse.

E. BounceNet Fairness:

Recall from section 2.6.2 that fairness in mmWave networks depends on how much each client interferes with other clients. If a client interferes with d other links, it should get at least a fraction of $1/(d+1)$ of time on the channel. For each of our 5000 experiments, we compute the fraction of channel time that a client interfering with d other links in the network obtains as a result of BounceNet's algorithm. Fig. 2.17 plots this fraction for all clients against their degree in the conflict graph (equivalent to their number of interfering links). The figure shows that the algorithm guarantees that all points lie above the line denoted by $Fraction = 1/(d+1)$. Hence, every link gets at least its fair share of channel time in BounceNet.

F. Application Level throughput in BounceNet:

In order to understand whether BounceNet's gains translate to higher layer network throughput, we evaluated the application level throughput achieved using BounceNet and 802.11ad under TCP and UDP traffic flows in ns3. Fig. 2.18 shows the throughput versus the number of clients. BounceNet's scaling properties are maintained with roughly the same gain over the 802.11ad standards. For 10 links, BounceNet can achieve a UDP throughput of 1.44 Gbps for 60 GHz with 12° beamwidth and 2.23 Gbps

for 3° beamwidth. As expected, the application level throughput is lower than the MAC data rates due to the overhead of headers. For TCP the throughput is even lower with 360 Mbps for 12° beamwidth and 740 Mbps for 3° beamwidth. This is expected as TCP has larger overhead and does not perform well in wireless networks.

G. Results Summary:

802.11ad requires multiple beacon intervals to detect interference in the network and schedule concurrent transmissions. While this would work in completely static scenarios where the paths do not change, it is inefficient in mobile or dynamic environments. Our results show that in such cases, 802.11ad keeps resetting to a configuration with no spatial reuse. BounceNet, on the other hand, is able to maintain an up-to-date view of the paths and interference every Beacon Interval which allows it to achieve significant gains especially for narrower beams (e.g. 3°) where the potential for spatial reuse is very high.

The baseline, on the other hand, performs well with narrow beams (e.g. 3°) and on average achieves comparable results to BounceNet. However, the tail of the distribution is very long. Specifically, clients that experience interference would achieve significantly lower data rates than both BounceNet and 802.11ad. The performance quickly degrades for wider beams where there is more interference between links. BounceNet can achieve the best of both worlds by combining efficient path learning and interference estimation algorithms with signal routing and beam alignment. Hence, BounceNet can exploit spatial reuse for both very narrow beams and wide beams and can perform well in both static and mobile environments.

2.10 Limitations and Discussion

Few points are worth noting.

- Our current evaluation is limited by today's hardware, which makes it infeasible to implement a full-fledged real-time version of our system. Cheap commercial mmWave devices [9, 10, 11] do not provide access to the lower layers: PHY and MAC. On the other hand, the hardware we used costs around $\$14,000$ for the RF front end of one TX/RX pair, making it prohibitively expensive to scale the implementation. Note, however, that our simulations are not based on ray-tracing or any channel modeling. Rather, they are based on actual measurements of SNRs and beam scanning through a labor-intensive study that generated over 5000 configurations. We have also used two pairs of links to verify that our interference estimates are accurate. Our results show a significant opportunity to scale the throughput in mmWave networks, and we believe the protocol can be implemented on cheap commercial devices if the chip manufacturers open up the firmware.
- BounceNet's protocol is mainly designed for continuous traffic in applications like VR, 3D video streaming, and Robotics. To deal with bursty traffic, one can leverage the polling mechanism available in 802.11ad [60] to obtain a *real-time* view of the traffic demands for different clients during the Beacon Interval, and adjust the conflict graph based on the traffic.
- BounceNet's interference estimation relies on accurate measurements of the SNR. The high directionality in mmWave networks reduces multipath fading and channel fluctuations which allows us to achieve accurate estimates as we show in section 2.8. However, to address the case of noisy and unstable SNR measurements, we take a more conservative approach for determining when two links interfere (Section 2.6.2.A). The threshold to determine interference can be adjusted as a trade-off between robustness to noisy SNR estimates and maximizing spatial reuse.

2.11 Conclusion

In this chapter, we introduced BounceNet, the first many-to-many millimeter wave beam alignment system that can efficiently align the beams of many APs and clients in a manner that allows them to simultaneously communicate without interfering. We evaluated BounceNet using three experimental testbeds and demonstrated that it can enable dense spatial reuse and scale the total network throughput with the number of APs and clients.

Chapter 3

SCALING WIRELESS NETWORKS-ON-CHIP FOR MASSIVE MULTICORES USING DEEP REINFORCEMENT LEARNING

3.1 Introduction

Recently, there has been an increasing interest from both industry and academia to scale network-on-chip (NoC) multicore processors to hundreds and thousands of cores [72, 73, 74, 75]. To enable such massive networks on chip, computer architects have proposed to augment NoC multicore processors with wireless links for communication between the cores [76, 77, 78, 79, 80]. The broadcast nature of wireless networks enables the NoC to significantly reduce the number of packets that the cores need to communicate to each other as well as the latency of packet delivery [81, 82]. Both aspects play a central role in scaling the number of cores on an NoC multicore processor (See Background Section 4.3 for details) [81, 82, 83, 84, 85]. These benefits have motivated RF circuits designers to build and test wireless NoC transceivers and antennas that can deliver multi-Gbps links while imposing a modest overhead (0.4–5.6%) on the area and power consumption of a chip multiprocessor [86, 87, 88, 89].

While the use of wireless can significantly benefit NoCs, it brings on new challenges. In particular, the wireless medium is shared and can suffer from packet collisions. Designing efficient medium access protocols for wireless NoCs is, however, difficult. The traffic patterns in NoCs tend to change drastically across applications. Even during the execution of a single application the traffic pattern can change as fast as tens of microseconds [81, 90]. As a result static MAC protocols such as TDMA, FDMA and CSMA perform poorly [91, 92, 93, 94, 95, 96, 97]. Further, due to thread synchronization primitives likes barriers and locks in parallel programming, the wireless NoC exhibits complex hard-to-model dependencies between packet delivery on the network and execution time. As a result, even adaptive protocols that try to switch between TDMA and CSMA or optimize for long-term throughput [80, 98, 99], perform poorly in the context of wireless NoCs since they remain agnostic to these domain specific and intricate dependencies. Hence, the design of efficient medium access protocols has been identified as a key bottleneck for realizing the full potential of a wireless NoC multiprocessor [100, 101].

In this chapter, we present BounceNet, a unified approach that combines networking, architecture and deep learning to generate highly adaptive medium access protocols for a wireless network on chip architecture. BounceNet leverages a reinforcement learning framework with deep neural networks to generate new MAC protocols that can learn traffic patterns and dynamically adapt the protocol to handle different applications running on the multi-core processor. Reinforcement Learning (RL) has proved to be a very powerful tool in AI for generating strategies and policies that can optimize for complex objectives [102, 103]. RL allows BounceNet to make better decisions by learning from experience. In particular, many basic functions, like FFT, graph search, sorting, shortest path, etc., tend to repeatedly appear in many applications. Past work also shows that a number of unique periodic traffic patterns emerge in multiple different programs, and as the number of cores increases, the traffic patterns show increasingly predictable spatiotemporal correlations and dependencies [104, 90]. BounceNet learns these statistics and correlations in the traffic patterns, to be able to both predict future traffic patterns based on traffic history and adapt its MAC protocol to best suit the predicted future traffic. Furthermore, RL enables BounceNet to account for hard-to-model complex dependencies between execution time and delivery of packets. In particular, we carefully engineer the reward function in RL to optimize for execution time rather than to simply improve the latency and throughput of the network.

Indeed, RL has been leveraged for wireless MAC protocols in the context of heterogenous wireless networks [105, 106], sensor networks [107], and IoT networks [108]. However, bringing these benefits to wireless networks on chip faces a number of unique challenges. First, past work runs RL inference for every packet at each time step, which is not feasible for WNoCs since the time scale of operation in a multicore processor is in the order of nanoseconds. Hence, per time-slot inference would significantly delay every packet transmission. Second, due to compute resource constraints, it is also not feasible to run RL inference at every core of the wireless NoC. While the second challenge can be addressed using a centralized controller for the RL model, it would still incur significant communication overhead and latency to collect the states from the nodes (e.g. traffic injections or buffer occupancy) and to inform the nodes when to transmit.

BounceNet addresses these challenges by designing a framework where the controller is trained to generate high-level MAC policies simply by listening to on-going transmissions on the wireless medium. This allows BounceNet to eliminate any communication from the cores to the controllers. Moreover, to amortize the overhead of inference and policy updates, BounceNet only updates the cores with a new MAC policy once every interval spanning many execution cycles (e.g. ten thousand cycles). We also train BounceNet to learn policies that are highly adaptive and simple to update, to reduce communication overhead from the controller to cores.

Finally, BounceNet also needs to operate within the strict timing and resource constraints of the multicore processor. Modern deep neural networks, however, are designed with up to a billion tunable parameters and operate on high dimensional input spaces [109, 110]. Consequently, they require large amounts of memory and computational resources, and also suffer high inference latencies (tens of milliseconds) [111, 112]. To address this, we design BounceNet's RL framework such that the input and output of the neural network scale linearly with the number of cores. This ensures that BounceNet is expressive enough to service the highly dynamic network traffic while at the same time operate under the limited memory and computational resources. Specifically, BounceNet's neural network requires three orders of magnitude less parameters, and adds a small area overhead to the multicore processor. It also has an inference latency that is small enough to meet the strict timing constraints of the multicore during run-time as we show in detail in Appendix C.

We evaluate BounceNet by integrating it with a cycle-level architectural simulator for CPU-GPU heterogeneous computing that faithfully models the intricacies of multi-core processors [113]. We augmented the simulator with an on-chip wireless network that accurately models transmissions, collision handling and packet losses. We test BounceNet's performance on real applications chosen from diverse domains such as graph analytics, vision and numerical simulations. We compare BounceNet against six baselines including wired NoC, standard CSMA, TDMA, optimal CSMA protocols [114], adaptive protocols [80, 81], and an optimal oracle. Our evaluation reveals the following:

- For a 64-core NoC, BounceNet is capable of learning traffic patterns and adapting the medium access protocol at a granularity of $10\mu s$ to achieve a median gain of $2.56\times - 9.18\times$ in packet latency and $1.3\times - 17.3\times$ in network throughput over different wireless NoC baselines.
- BounceNet's throughput and latency gains translate into an average of $10\% - 47\%$ speedup in execution time over wireless NoC baselines which goes up to $1.37\times - 3.74\times$ for certain applications. The results also show a $3.4\times$ speedup on average over a purely wired NoC.
- BounceNet's gains in execution time are close to the upper bound that can be achieved by a wireless network with infinite capacity and zero latency.
- As the number of cores scale up to 1024 cores, BounceNet's performance gain increases to 3 orders of magnitude lower latency and up to $64\times$ higher throughput over baseline protocols.
- BounceNet is robust to lossy channels, and sees minimal degradation in performance with upto 10% packet losses. We also test BounceNet's sensitivity to noise in the observed state and show almost no loss in performance.

Contributions: We make the following contributions:

- We introduce the first MAC protocol that can learn and adapt to the highly dynamic traffic at very fine granularity in a wireless NoC processor. The protocol also accounts for non-trivial dependencies between packet delivery and computation speedups by optimizing for execution time.
- We design a lightweight deep reinforcement learning framework that introduces little overhead to the multi-core processor and can operate within tight timing, power and area constraints of chip multicore processors.
- We extensively evaluate our design and demonstrate significant improvement in network performance and reduction in the overall execution time on the multicore processor.

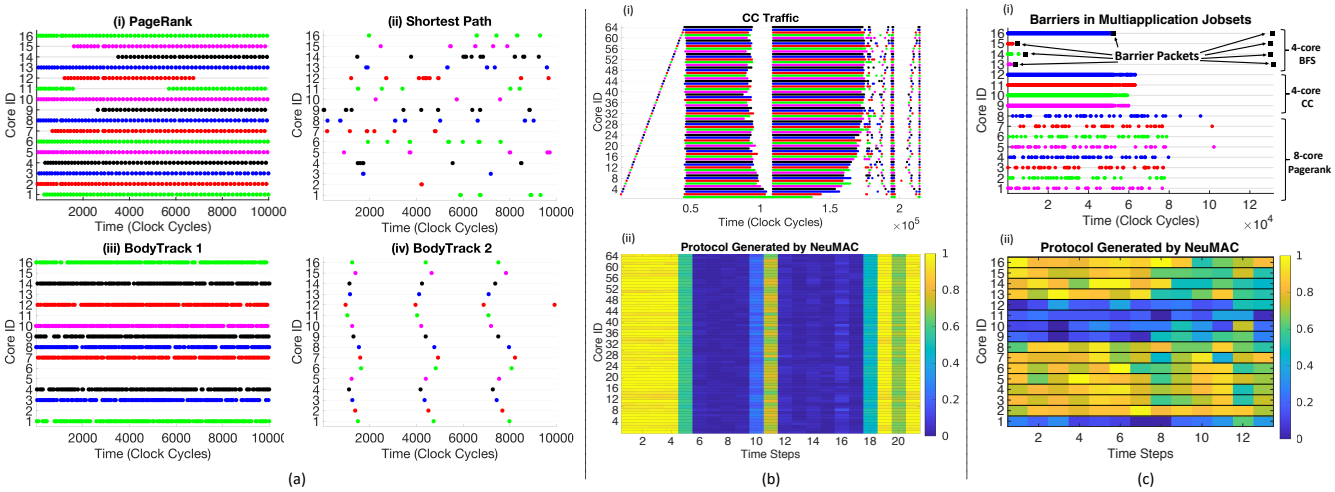


Figure 3.1: **Illustrative Examples: (a)** Traffic Pattern on a 16-core multiprocessor for different applications. The X-axis shows clock cycles, and the Y-axis corresponds to each of the 16 cores. The figures depict the scatter plots representing the packet injections into the buffer of each core. The different colors for packet injections are used for different cores. **(b)** BounceNet can quickly adapt to fast changing traffic thus ensuring efficient network utilization throughout the application's execution. In the generated protocol, high probability values (closer to yellow in colormap) represent a CSMA-like protocol whereas low probability values (closer to blue) represent a TDMA-like protocol. **(c)** BounceNet can learn and optimize for the intricate dependencies between the executions on different cores, and in turn optimize directly for end-to-end execution.

3.2 Motivation and Insights

The wireless traffic patterns on a multicore processor have been shown to vary significantly across different applications. Even for a single application, the traffic can vary across different cores (spatially) and across different time intervals (temporally) [81, 90, 115, 100, 101].

Fig. 3.1(a) shows examples of traffic traces captured from a cycle-level architectural simulator for three different common benchmark applications on a 16-core multiprocessor. The x-axis shows the time in clock cycles, the y-axis shows the core ID, and the scatter points show the injection of traffic at each core. For clarity, we only show a portion of the execution spanning ten thousand cycles. Some applications, like *PageRank* shown in Fig. 3.1(a)(i), have almost constant traffic on all cores and can benefit from a contention-free protocol like TDMA. Other applications, like computing the *Shortest Path in a Graph* shown in Fig. 3.1(a)(ii), have very bursty traffic and can benefit from a contention-based protocol like CSMA. Moreover, in most applications, the traffic pattern changes within the execution

of the application. For example, Fig. 3.1(a)(iii)-(iv) show the traffic patterns at different times in the execution of *BodyTrack*, a computer vision application for tracking body pose. In the first time interval, since there is steady injection of packets into the network on the 10 active cores, a contention-free scheme will be optimal to minimize collisions, whereas in the second time interval, a CSMA-like based scheme for all 16 cores will perform better due to the sparse traffic injection. Next, we present concrete examples showcasing the range of protocols that BounceNet can generate for different traffic patterns.

A. Adapting to Dynamic Traffic Patterns: To further appreciate the spatial and temporal changes across the execution of an entire application, we show the traffic trace for the application CC (Connected Components of a graph), running on a 64-core processor in Fig. 3.1(b)(i). Here we can see that the traffic varies significantly across the application's execution.

Fig. 3.1(b)(ii) presents the protocol generated by BounceNet. At a very high level, BounceNet's protocol is simple. Each core gets its own dedicated time slot where it can transmit with probability 1 if it has traffic. Additionally, core i can also transmit in time slots assigned to the other cores with some contention probability p_i. By setting these probability values p_i for each core, BounceNet dictates the MAC protocol on the wireless NoC. The figure shows these contention probabilities p_i's for each core generated by BounceNet. We present BounceNet's protocol design in more detail in Section 3.4.3.

From Fig. 3.1(b)(ii), we can see that BounceNet is able to adapt quickly to the changes in the traffic patterns, becoming more TDMA-like when the traffic is dense (contention probabilities p_i's are 0 and everyone transmits only in their assigned slot), and becoming more CSMA-like with sparse traffic (contention probabilities p_i's are high and cores can start transmitting in other's assigned time slots). In the case of CC, we can see that initially the traffic pattern is extremely sparse and structured such that a simple "Aloha" protocol would suffice. As a result, in the beginning the cores contend for the channel aggressively under BounceNet's protocol. However, once the traffic pattern becomes more dense, BounceNet adapts the protocol to be more TDMA-like, thus ensuring high network utilization. Finally, once the traffic pattern becomes less dense after $18 * 10^4$ cycles, the cores again start to contend for the channel with higher probability, thus emulating a CSMA-like protocol. Note that, while BounceNet is able to quickly detect traffic changes from dense to sparse at time steps 11 and 18 (From Fig. 3.1(b)(ii)), it does not immediately increase contention probabilities for the cores. Instead the change is gradual, and this is because of the outstanding packets remaining in the buffers immediately after the phase with dense traffic injection. As a result, immediately switching the probabilities would lead to large number of collisions.

The above example demonstrates that BounceNet is able to learn fine-grained highly dynamic MAC protocols that can quickly adapt to support different kinds of traffic patterns, while accounting for subtle

characteristics of network operations such as buffer build-ups even though this information is not explicitly fed into BounceNet's RL model. While there has been a lot of work on adaptive and optimal CSMA protocols [116, 117, 118, 119], these works are theoretical and make unrealistic assumptions. In particular, they optimize for long term throughput and assume that the protocol can reach a steady-state operation much faster than the variation in traffic patterns, which does not hold for wireless NoCs. As a result, these protocols perform poorly as we show in section 4.10.

B. Optimizing for Synchronization Primitives: Another challenge in designing efficient protocols stems from synchronization primitives. These primitives impose intricate dependencies between the execution of threads on different cores, leading to a non-trivial relationship between the delivery time of packets on the NoC and the progress of execution on each core. For example, in parallel computing it is common practice for software developers to use `barriers` for synchronization. These barriers are placed throughout the code of a multithreaded application in order to force each thread to stop at a certain point, blocking its execution until all participating threads catch up. Most standard libraries for parallel programming use barriers in many of its primitive routines in order to ensure the correctness of the program, such as OpenMP's `For` loop [120], or MPI's `Send/Recv` [121]. Therefore, there is complex but predictable structure in the traffic patterns caused by these synchronization primitives that can be exploited to improve parallel speedup and scalability of high performance applications. Hand tuning protocols to account for these dependencies is non-trivial. For example, the cores themselves do not explicitly know that they are involved in a barrier before they actually reach the barrier and execution halts. [122, 123]. Past work on designing MAC protocols mainly optimizes for throughput and latency, and is agnostic to such dependencies.

As a concrete example, consider the multiapplication jobset comprising of three concurrent applications, namely a 4-core BFS, a 4-core CC and a 8-core Pagerank, running on a 16-core multiprocessor as shown in Fig. 3.1(c)(i). In the traffic trace, one can observe two sets of barrier packets in the execution of BFS, denoted by black squares. The other two applications have no barriers in this portion of their executions. Here, note that core 16 has significantly more packets to transmit before arriving at its barrier, whereas core 13, 14 and 15 arrive at their barriers sooner. As a result, the execution on cores 13, 14 and 15 is blocked until core 16 clears its barrier, thus rendering the compute resources of these three cores useless as they idly wait for core 16. Additionally, at the same time core 16 also has to contend for the channel with traffic from CC, which itself has a lot of ongoing communication. Ideally, the MAC protocol in this case should prioritize traffic of the core that is falling behind, so that it arrives to the barrier and clears it as soon as possible, allowing the blocked cores to proceed execution and thus optimizing overall execution time. In Fig. 3.1(c)(ii), we can see that BounceNet can learn to account and optimize

for such dependencies. At the start, BounceNet assigns high contention probabilities to cores 13 to 16 so that it can clear the barrier point at the earliest, while assigning low contention probabilities to cores 9 to 12. Once the barrier is cleared, BounceNet increases the contention probabilities for the CC cores, so that it can transmit on the channel while the other applications go through low communication periods, thereby ensuring high network utilization.

Protocols like CSMA, TDMA and even adaptive protocols cannot optimize for such situations, as they would treat every packet in the network as equally important, thus sharing the channel equally between BFS and CC here. This would result in core 16 clearing its barrier much later, thus harming end-to-end execution time. However, since BounceNet is trained to directly optimize the high-level objective of end-to-end execution time instead of network metrics like latency, it is able to learn to prioritize the packets of some cores over others. In this example, with BounceNet's protocol, core 16 arrives at its barrier $2.4\times$ faster as compared to CSMA, and $3.75\times$ faster as compared to TDMA. This in turn leads to an overall improvement in execution time of 43% and 81% over CSMA and TDMA respectively.

3.3 Background

3.3.1 Wireless Network on Chip

Network-on-Chip (NoC) architectures have played a fundamental role in scaling the number of processing cores on a single chip which led to unprecedented parallelism and speedups in execution time [124, 125, 126, 127]. Prior to NoC, multicore processors used a shared bus architecture which had very poor scalability. As the core count increases, the power required to drive the bus grows quickly due to the increase in the capacitance of the bus wires [128]. The bus also starts to suffer from large latency [129]. As a result, shared buses become impractical for designs beyond 16 cores [130].

Unlike a shared bus, wired NoCs use packet-switched communication with every core connected to a router as shown in Fig. 3.2 [131]. As the packet moves from source to destination, it is buffered, decoded, processed, encoded, and retransmitted by each router along the multi-hop path. However, as we scale the number of cores, computation slows down due to the high communication latency and overhead of the network [132, 133, 134]. This problem is known as the "Coherency Wall" [135], where the execution on each core is faster than the NoC's ability to ensure that the memory caches of the cores are coherent. Hence, the speedup gained by parallelism and multithreading is outweighed by the network's communication cost for keeping the caches coherent [135, 136, 85].

Recent work proposes to augment NoC multicore processors with wireless links for communication between the cores [76, 77, 78, 79, 80]. Wireless links benefit chip multicore processors in two important aspects:[1]

- **Lower Latency:** Wireless enables every core to reach every other core in just a single hop. In contrast, in a purely wired NoC, a packet must go through multiple NoC routers, incur queuing, transmission, and processing delay at every hop which ends up taking multiple execution cycles [82]. Hence, as the number of cores increase, wireless can deliver packets with significantly lower latency and within the tight timing requirements of execution on the cores [82].
- **Broadcast:** Since wireless is a broadcast medium, transmitted packets are directly heard at all other cores which significantly simplifies the NoC's ability to ensure the coherency of the memory caches. In particular, any local changes in the memory cache of a core can instantaneously be replicated at all other cores through a single packet transmission [81]. In contrast, today's wired NoCs must send multiple parallel unicast/multicast transmissions to synchronize the caches, which leads to a large overhead that

[1] Note that other technologies such as optical links have poor performance [79, 137, 138].

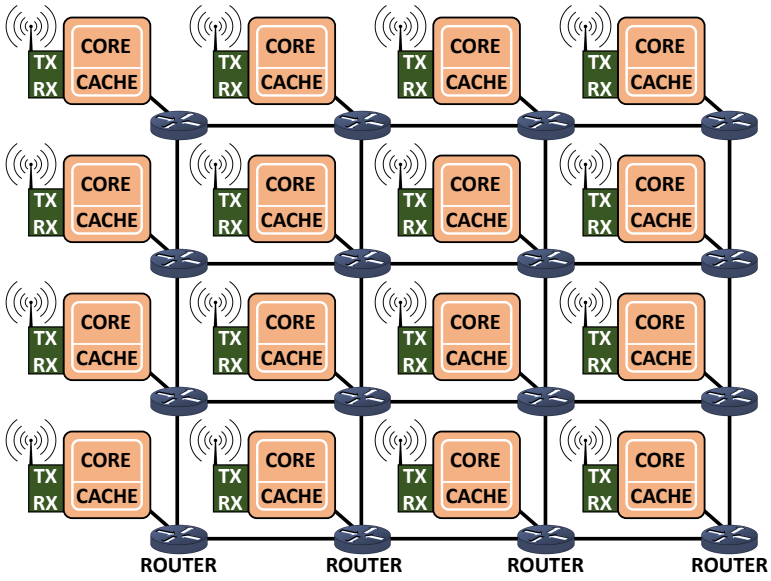


Figure 3.2: NoC Architecture with Wireless Links

scales poorly as the number of cores increases [83, 84, 85].

Several wireless NoC transceivers and antennas have been built and shown to deliver 10 to 50-Gbps links while imposing modest overhead (0.4–5.6%) on the area and power consumption of a chip multiprocessor [86, 139, 87, 88, 89]. The wireless transceivers typically operate in the millimeter-wave and sub-THz spectrum which enables miniaturizing the antennas and avoids antenna coupling. Antennas are either planar integrated dipoles or vertical monopoles drilled through the silicon die [140, 141]. The wireless signals propagate through the enclosed chip packaging and attenuate by few tens of dBs [142, 140]. On-Off Keying (OOK) is the choice of modulation since it requires significantly lower powerand achieves a very low Bit Error Rate (BER) for on-chip wireless links [89, 143, 139]. We adopt the collision and packet loss handling protocols from past work [81, 82].

3.3.2 Deep Reinforcement Learning

We provide a brief primer on RL based on [144]. In RL, an *agent* interacts with an *environment*, and learns to generate a policy directly from experience as shown in Fig. 3.3. In our case, BounceNet is the *agent*, the multiprocessor is the *environment*, and the generated MAC protocol is the policy.

• **Agent & Environment:** The *agent* starts with no apriori knowledge. Then, at each time step t, the agent

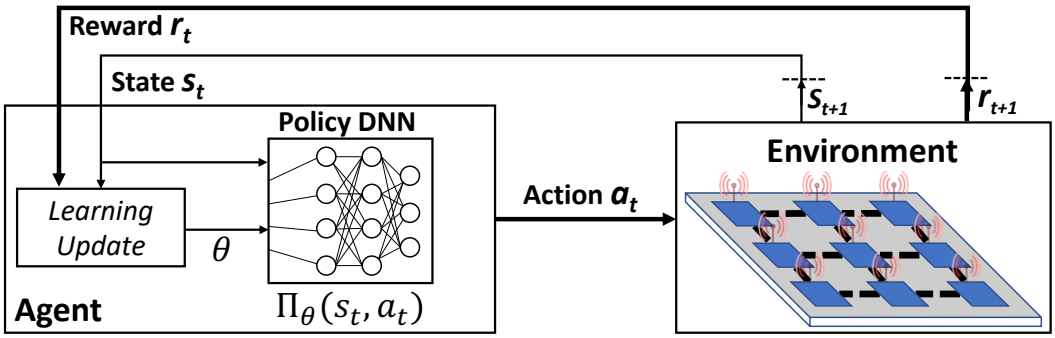


Figure 3.3: Deep Reinforcement Learning Framework.

observes the state s_t of the environment, and takes an action a_t. Following the action, the environment transitions to state s_{t+1}, and the agent receives a reward r_t. The state transitions and the rewards are stochastic and assumed to have the Markov property. During training, the *agent* gains experience by taking actions and observing the state transitions and rewards in response to these actions. The actions the *agent* takes aim to maximize an objection function known as the expected cumulative discounted reward: $\mathbb{E}\big[\sum_{t=0}^{\infty}\gamma^t r_t\big]$, where $\gamma \in (0,1]$ is the discount factor for future rewards.

• **Policy:** The action a_t picked by the *agent* is dictated by a *policy* π, where π represents a probability distribution over the space of actions and states : $\pi(s,a) \rightarrow$ [0,1]. That is, $\pi(s,a)$ is the probability that action a is taken in state s by the agent following policy π. For most large-scale practical problems, the policy π is modeled with a Deep Neural Network (DNN), as they are very powerful function approximators. The DNN is parameterized by θ, which are the learnable parameters of the model, and we represent the policy as $\pi_\theta(s,a)$. θ is also referred to as the *policy parameters*.

• **Training:** The objective of training in RL is to learn the policy parameters θ so as to maximize the expected cumulative reward received from the environment. Towards this end, we focus on a class of RL algorithms called *policy gradient algorithms*, where the learning takes place by performing *gradient descent* on the policy parameters. In practice, the training methodology follows the *Monte Carlo method* where the agent samples multiple trajectories obtained by following the policy π_θ, and uses the empirically computed cumulative discounted reward as an unbiased estimator of the expected value. This empirical value is then used to update the policy parameters via the gradient descent step. The result is a known algorithm: REINFORCE which we use in this chapter. For more details, we refer the reader to [144].

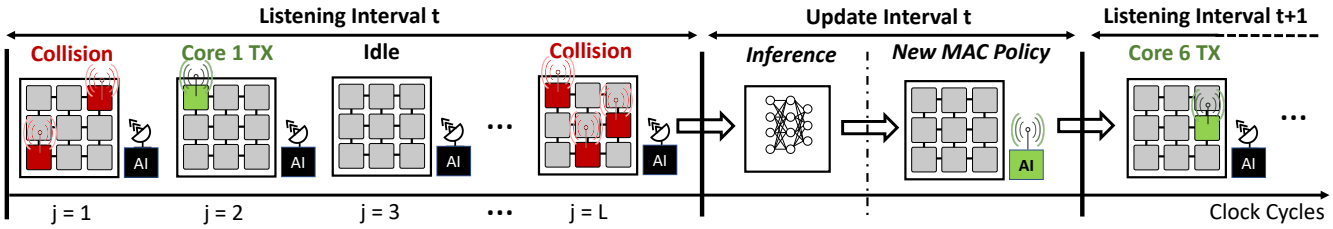


Figure 3.4: An Overview of BounceNet's Protocol

3.4 BounceNet Design

3.4.1 Overview

BounceNet consists of two components. (1) A standard NoC multicore processor with N cores where each core has been augmented with a wireless transceiver as shown in Fig. 3.2. (2) A BounceNet agent that periodically generates new medium access policies based on the traffic patterns it sees on the wireless NoC. The agent is housed in a simple neural accelerator that resides on the same chip with a small area and power overhead (See Appendix C for hardware details).

Fig. 3.4 shows the working of BounceNet. The BounceNet agent is equipped with a wireless transceiver through which it can listen on the channel, and also send protocol updates to the cores. The BounceNet agent listens on the wireless channel for a period called the "*Listening Interval*" where it collects traffic data about core transmissions, collisions, and idle slots. It, then, feeds this data to a trained RL neural network that implicitly predicts the future traffic patterns and generates a new policy to be used as the medium access protocol during the next *Listening Interval*. BounceNet updates the policies at the cores by sending an update message with the policy parameters. Each *Listening Interval* and *Update Interval* constitute a single step in the RL framework.

One point to note is that, although the cores share a common clock for their normal CPU operation[2], it is infeasible to coordinate medium access for each clock cycle through a shared centralized scheduler, since the exchange of control messages between the cores and the scheduler would itself incur latencies of multiple clock cycles. [101]

[2]Unlike a distributed system of machines, a shared clock for a manycore system is feasible since all cores are housed on the same silicon die.

3.4.2 Design Challenges

The above design is governed by several strict timing and resource constraints of wireless NoC. In particular, it must address the below challenges while at the same time ensuring BounceNet's ability to generate versatile and expressive medium access protocols to service the dynamic and fast-varying traffic patterns.

C1. *Centralized Agent:* Ideally, we would have wanted BounceNet to adopt a distributed design where every core is equipped with its own BounceNet agent that dictates its own MAC protocol. However, introducing a neural accelerator at every core would be prohibitively expensive in terms of area and power. Hence, BounceNet is constrained to a centralized approach with a single agent.

C2. *Cores to Agent Communication Overhead:* To obtain an accurate view of traffic patterns, BounceNet must obtain the packet injection rate and buffer occupancy across time at each core in the network. However, relaying this information from every core back to the centralized agent would result in huge communication overhead. Instead, BounceNet leverages the broadcast nature of wireless networks to collect traffic patterns simply by listening for transmissions on the wireless medium. While the collected information is less expressive than the history of packet injection and buffer occupancy at each core, it retains sufficient information to allow BounceNet to predict traffic patterns while at the same time completely eliminating communication overhead from the cores to the centralized agent.

C3. *Agent to Cores Communication Overhead:* One option is to have the agent tell each core whether to transmit or not at every CPU clock cycle. However, this would require running inference and relaying information to each core at every clock cycle which would lead to prohibitively large communication overhead. To address this, BounceNet amortizes the communication overhead (*Update Interval*) from the agent to the cores by performing inference once every *Listening Interval* spanning thousands of clock cycles. In our implementation, we use an interval of L =10,000 clock cycles ($10\mu s$) which is large enough to reduce the overhead to less than 6% and small enough to ensure that the traffic patterns remain stable and can be learned by the RL agent.

C4. *Complexity of the MAC Policy:* BounceNet generates a policy that dictates the MAC protocol of each core for the following *Listening Interval*. Ideally, BounceNet would generate a deterministic transmission schedule for every core to follow. Such a design is extremely expressive since it could allow BounceNet to generate any possible schedule. However, such a design would require the RL deep neural network to output an action space with $N \times L$ dimensions where N is the number of cores and L is the number of clock cycles (e.g. 10,000). Such a neural network would be unsuitable for a resource-constraint setting like NoC. To address this, we carefully design a parameterized MAC policy that can support a flexible range of medium access protocols while ensuring that the neural network only needs to output a few

parameters to dictate the desired policy.

C5. *Reward engineering:* The reward during training needs to be designed so as to guide BounceNet towards the high-level objective. While most past work on learning link-layer and network-layer protocols only use network-level metrics such as throughput and latency for the reward signal, in our case we need to choose domain specific rewards so as to optimize for the end goal, which is application execution speedup on the multicore.

C6. *Low Footprint Neural Network:* BounceNet's neural network must adhere to strict timing, power and area constraints of a chip multiprocessor. Thus, our design cannot simply adapt a known RL model as it would require large amounts of memory and computational resources, and would also suffer high inference latencies (tens of milliseconds) [111, 112]. To address this, we design BounceNet's RL framework such that the state space (input to the neural network) and action space (output) scale linearly with the number of cores. Our design ensures that BounceNet is expressive enough while at the same time can operate under NoC's resource constraints.

3.4.3 BounceNet's MAC Policy

As discussed above, the MAC policy that the agent dictates to the cores should have the following properties:

1. The policy should span a wide range of protocols, all the way from TDMA to CSMA.
2. It should be possible to describe the policy with few parameters to reduce the communication overhead and the output of the neural network.
3. It should allow for a simple neural network architecture to learn a mapping from observed traffic patterns to the most efficient MAC protocol.

In order to achieve these properties, we adopt a two-layer protocol design. The first layer consists of a deterministic underlying TDMA schedule, where each core is assigned a unique time slot for transmission in a round-robin fashion. For example, for time slots $j \in [1, \cdots, L]$, core i is assigned the slots $\{j \mid j \mod N = i\}$ where N is the number of cores. The second layer consists of a probabilistic transmission schedule like CSMA, where each core is assigned a contention probability. Specifically, during its assigned time slot, core i transmits on the channel with probability 1 if it has an outstanding packet in its buffer. During other cores' assigned time slots, core i can transmit with probability p_i. In the event of a collision, exponential backoff is implemented by halving p_i of the colliding cores similar to CSMA. On the other hand, if a transmission is successful, p_i is reset to its initial value.

Algorithm 3 BounceNet Protocol

$L \leftarrow$ Number of Clock Cycles in Listening Interval
$[a_{1,t}, a_{2,t}, \ldots, a_{N,t}] \leftarrow$ Action space generated by RL agent at time step$t$
$[p_1, p_2, \ldots, p_N] \leftarrow [a_{1,t}, a_{2,t}, \ldots, a_{N,t}]$

At core i:
for $j \in \{1, \cdots, L\}$ **do**
  $Buffer_i(j) \leftarrow$ Outstanding packet in the buffer for core i
  if $Buffer_i(j) \neq \emptyset$ **then**
    if $j \mod N = i$ **then** ▷ TDMA Slot Assigned to Core i
      Transmit with probability 1
    else
      Transmit with probability p_i
  if Transmission from Core i collides **then**
    $p_i = p_i/2$
  else
    $p_i = a_{i,t}$

To generate this policy for an NoC with N cores, the RL neural network needs to output an action space that can be defined as $a_t = [a_{1,t}, a_{2,t}, \ldots, a_{N,t}]$ where $a_{i,t} \in [0, 1]$ represents the initial contention probability of core i during *"Listening Interval"* t (i.e., time step t in the RL framework). The contention probability of core i is then initialized as $p_i = a_{i,t}$. Different choices of a_t result in different protocols on the multicore. For instance, setting $a_{i,t} = 0$ for all i results in a simple TDMA protocol since every core only transmits on the channel during its assigned slot. On the other hand, $a_{i,t} = c > 0$ for all i mimics a CSMA-like protocol with varying degrees of aggressiveness on the channel. The pseudo code for BounceNet's protocol is presented in Alg. 3.

The above formulation satisfies our design objectives. First, it enables BounceNet to gracefully shift between a pure TDMA and a CSMA scheme, while supporting all intermediate protocols. The design also gives the flexibility to control each core individually, so that the BounceNet can potentially increase contention probabilities for cores that observe high traffic intensity. Second, since the MAC protocol at core i is characterized by only one number (the contention probability $a_{i,t}$), there is very small communication overhead during the *Update Interval*, where the BounceNet agent has to transmit a single broadcast packet with N numbers. Each core, receives the packet and extracts it own contention probability. Finally, the design keeps the action space constrained and linear in the number of cores, which allows for a simple neural network that can be easily trained and is more likely to converge.

3.4.4 RL Formulation and Training

Given the above design, we now formalize the state space, reward, policy and training of BounceNet's RL framework.

• **State Space Design:** The BounceNet agent takes state information s_t as input and generates a MAC policy characterized by the action space a_t described above. The state information is generated purely by listening to ongoing transmissions on the channel. As described earlier, this allows us to eliminate all communication overhead from the cores to the RL agent. However, it only provides information about the activity on the channel rather than the traffic injection into the network. Moreover, in the event of a collision, BounceNet cannot know which cores attempted to transmit. Despite these limitations, BounceNet's state space retains enough information to infer traffic patterns. In particular, during each CPU cycle, BounceNet will either detect an idle channel, a collision, or a successful transmission from some core i. We define our state at time step t, s_t, as an $(N+1) \times 1$ vector that keeps track of the number of successful transmissions from each core and the number of collisions observed during the cycles in the RL time step (*Listening Interval*). Specifically, the i^{th} element of s_t counts the number of successful packet transmissions by core i, and the $N + 1^{th}$ element counts the number of collisions. The number of idle slots is implicitly encoded in the state since it is equal to $L - \sum_{i=1}^{N+1} s_{i,t}$ where L is the number of cycles in a *Listening Interval*. The state s_t is then used by the BounceNet agent to generate the MAC protocol policy for the next time step.

• **Reward Engineering:** The reward signal is designed to guide the agent towards policies that optimize for the desired objective. Most past work that uses RL for learning networking protocols employs network-level metrics like throughput or latency as the reward signal. However, in our case, we need the reward signal to directly represent our end goal, which is to optimize for speedups in application execution time on the multicore. While network-level metrics like throughput are correlated to the execution time, they do not always capture the intricate dependencies between the execution on threads and packet delivery on the network. In Section 4.10, we see that there are instances where a protocol performs significantly worse in terms of average network throughput, but still has better end-to-end application execution time.

As a result, we design our reward signal to reflect our high level objective of minimizing application execution time. Specifically, for each time step t, the reward is set to $-L_t$ where L_t represents the number of clock cycles where the application was executing. Hence, for all but the last time step, the reward signal r_t is set to $-L$. For the last time step, reward is set to $-k$, where k is the number of clock cycles at which the application terminates execution. The intuition behind this choice for the reward

signal is as follows. Recall that the objective of reinforcement learning is to maximize the cumulative reward, i.e. $-\sum_t L_t$. This is equivalent to minimizing $\sum_t L_t$, which ultimately means the application utilizing fewer CPU clock cycles for execution. While this choice of reward signal does correlate with improving network-level metrics such as packet latency and throughput, it is not the central objective and thus it is possible that sometimes the BounceNet agent compromises on network performance for improvement in execution time. Note that in our formulation, we set the discount factor $\gamma = 1$.

• **Policy:** We represent our policy π as a deep neural network (also called policy network) which takes as input the state s_t, and maps it to a_t in the action space. Note that in our problem, the action space is continuous. In such cases, it is common to discretize the continuous action space $a \in [0, 1]^N$ similar to [145], and convert the problem into a classification problem where the agent now chooses which combination of a_i's to pick. However, an obvious issue with this approach is the curse of dimensionality. Even with 2 quantization levels for each a_i, the total number of discretized actions in $a \in [0, 1]^N$ becomes 2^N. Thus the neural network architecture needs to have an output dimension of 2^N which becomes infeasible for our resource constrained environment.

Therefore, we avoid discretizing the action space and, instead, model the actions as following a Gaussian distribution with mean μ and variance σ. The deep learning model is now trained to output the parameters of this Gaussian distribution, as described in [144]. The BounceNet agent picks the action for the next time step simply by sampling from the distribution $\mathcal{N}(\mu, \sigma)$. In BounceNet, the policy network outputs N parameters μ_i corresponding to N distributions, one for each core i. The variance σ is set to 1 at the start of training to encourage exploration, and annealed down to 0.05 as BounceNet's policy improves. Finally, during inference, the variance σ is set to 0.05, the action $a_{i,t}$ for core i is sampled from the corresponding distribution $\mathcal{N}(\mu_i, \sigma)$, and clipped to ensure that $a_{i,t} \in [0, 1]$.

• **Training Algorithm:** We train our policy network end-to-end in an *episodic* setting. In each episode, an instance of an application is executed on the multicore, and the wireless network on chip follows the MAC protocol as dictated by the BounceNet's policy network. The episode terminates when the application completes execution. In order to learn a policy that generalizes well, we train the network for multiple episodes with each episode observing a different application trace. For every episode, we run M separate Monte Carlo simulations to explore the probabilistic space of possible actions using the current policy, and use the resulting data to improve the policy for all applications. Specifically, we record the state, action, and reward information for all time steps of each episode. We then use this data to train our policy using the popular REINFORCE algorithm along with a baseline subtraction step, as described in [146].

3.4.5 Neural Network Architecture

Our network is composed of three fully connected layers with 128, 128 and 64 neurons respectively. The first two layers are followed by ReLU activation units, whereas the final layer is followed by a sigmoid unit to output the probability values a_i's between 0 and 1. During training, the weights use 16 bit floating points. Once trained, the learned weights are quantized to 8 bit fixed points for the inference stage. This is standard for run-time optimization in deep learning [147] and does not adversely affect performance.

The proposed fully connected network architecture here is simple and ties in very well with our design objectives. Recall that BounceNet performs one inference step every 10,000 CPU clock cycles, and we require the inference step to add little overhead. The architecture here is composed of 32,000 learnable parameters, and at 8-bit quantization, it can be stored in a 32 KB on-chip SRAM cache to ensure fast memory accesses. Since inference latencies in most neural network architectures tend to be memory bound (including Fully connected and CNN architectures) [148, 147], improving memory access latencies plays a big role in speeding up overall inference time. Further, the simple structure of a fully connected network allows for straightforward memory access patterns, since the inference step is a straightforward computation amounting to consecutive matrix multiplications. In Appendix C we provide energy-delay characterization of this architecture.

One point to note is that BounceNet's deep RL agent is trained offline, and does not undergo any training during run-time since training is resource intensive. However, retraining can be triggered periodically depending on performance requirements, and this retraining will be performed offline. The updated model parameters can then be migrated to the neural hardware accelerator by simply rewriting the SRAM memory blocks on the accelerator corresponding to the neural network's model parameters. This update can happen through the multicore's wireless NoC communication channel and will not add much overhead, since our model is restricted to just 32,000 parameters, each of 8 bits.

Name	Description
BFS [149]	Breadth-first search
Bodytrack [150]	Tracking a body-pose through images
Canneal [150]	Compute optimal routing for gates on a chip
CC [149]	Compute connected components of a graph
Pagerank [149]	Compute pagerank for nodes in a graph
SSSP [149]	Single source shortest path
Volrend [151]	Rendering of 3D objects
StreamCluster [150]	Cluster streams of points
Community [149]	Compute modularity of a graph

Table 3.1: Summary of Applications

3.5 Implementation

Evaluation Environment: We evaluate BounceNet on a cycle-level execution-driven architectural simulator, Multi2sim [113]. Multi2sim is a popular end-to-end heterogenous system simulator tool used in the architecture community to test and validate new hardware designs with standard benchmarks. We evaluate BounceNet for multicores with core count $n = 64$ at 22nm technology running at 1GHz. We use the same architecture parameters as [81]. We augment Multi2sim with an on-chip wireless network that accurately models transmissions, collision handling and packet losses.

While BounceNet could be potentially trained directly using multi2sim, it is extremely slow and would result in prohibitively large training times. Therefore, for BounceNet's training phase, we use a light-weight custom-built Wireless Network-on-Chip simulator along with traffic traces captured from Multi2sim. Our custom simulator models the data dependencies and synchronization primitives (such as locks and barriers) in the applications, so as to faithfully mimic the behavior of multi-threaded applications.

In order to evaluate BounceNet's generalizability and effectiveness for a broad use case, we test BounceNet on 9 different applications chosen from diverse domains such as graph analytics, vision, and numerical simulations (Summary in Table 3.1). Additionally, we also test with multi-application jobsets where different groups of cores are executing different multithreaded applications. While training is performed using our custom simulator, we evaluate BounceNet using Multi2sim. We integrate Multi2sim with BounceNet's trained RL agent, and our evaluations account for the RL agent's DNN inference latency and communication latency between the multicore and RL agent.

Training and Evaluation Details: For each application, we collect 500 different traces, each generated with different inputs to the applications in order to capture the variations between different runs. We

evaluate BounceNet using k-fold cross validation, where we train the model on 8 applications and test performance on the ninth application. Thus, we ensure that the BounceNet agent is never explicitly trained on the application it is being evaluated on, and our results show that BounceNet can generalize well to different applications. We train BounceNet for a total of 4000 episodes, and for each episode we run $M = 16$ Monte Carlo simulations in parallel. The policy network is trained using ADAM optimizer [152] with a learning rate of 0.001.

3.6 Evaluation Results

3.6.1 Baselines

We compare with the following baselines:

(1) **CSMA with Exponential Backoff:** CSMA/CA protocol from 802.11 networks, with backoff window ranging from 1 to 1024. [82, 97] use CSMA MAC in the context of WNoCs.

(2) **TDMA:** Cores are allocated fixed slots for transmission in round-robin fashion. [136, 153] evaluate TDMA for WNoCs.

(3) **Switch-thresh:** [81, 80] propose a protocol that switches between a static CSMA and a static TDMA protocol based on per-core preset thresholds for channel activity and buffer occupancy. The optimal threshold values vary across applications and we choose values that are best in the average case.

(4) **Optimal CSMA Algorithm:** There is a large body of work that designs throughput optimal CSMA algorithms. However, most of these works are theoretical, and make simplifying assumptions like ignoring collisions or static traffic arrival rates, due to which they perform significantly worse than even regular CSMA protocols in practice. Among the optimal CSMA algorithms we tested, we found queue-based algorithms to perform best. We implement an extension of the popular Q-CSMA algorithm [114], where each node uses its buffer queue buildup to infer its transmission aggressiveness on the channel. While this algorithm is not truly distributed in nature, we ignore the global communication overheads in evaluations to favor the baseline performance.

(5) **Wired Baseline:** We also compare performance against a purely wired baseline, where all cache coherency traffic is serviced through the wired network-on-chip.

(6) **Infinite Capacity Channel:** We also compare BounceNet's performance against an oracle with infinite channel capacity where the wireless medium can support multiple concurrent transmissions without suffering collisions, and every packet can be transmitted immediately without any channel contention delays. This baseline gives us an upper bound on how much improvement in end-to-end execution time is possible from improving the wireless NoC performance.

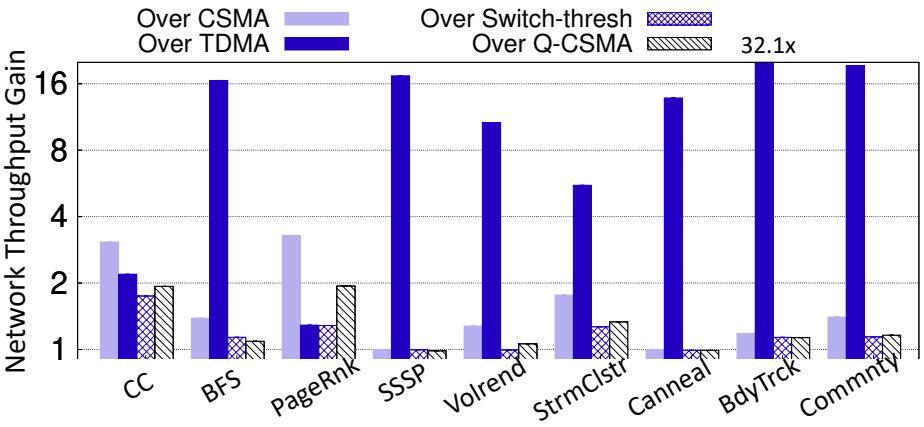


Figure 3.5: Gains in Wireless Network Throughput. (y axis in logscale)

3.6.2 Quantitative Results

We first evaluate BounceNet's performance against baselines on single application executions, followed by evaluations on the more realistic scenarios where multiple applications are running on the multicore. We also test BounceNet's performance under lossy network conditions, and conclude by presenting scaling results where we demonstrate that BounceNet's gains increase as the multicore scales to thousands of cores.

A. Single Application Wireless Network Performance:
We begin by evaluating the wireless network performance against baselines along three metrics – (i) Wireless network throughput, (ii) Packet latency on the wireless network, and (iii) Number of collisions on the channel. We note that while BounceNet is not explicitly trained to optimize for network metrics, their performance is correlated to faster execution times on the NoC.

(i) Network Throughput: In Fig. 3.5, we plot the gains in average network throughput achieved by BounceNet against the baselines. Compared to CSMA and TDMA, BounceNet achieves a mean improvement of $1.8\times$ and $9.63\times$ respectively across the benchmarks, and a maximum improvement of $3.3\times$ and $32.1\times$ respectively. TDMA has poor performance for average network throughput since cores have to wait for their turn to transmit even when the traffic is sparse, which leads to underutilization of channel.

Compared to Switch-thresh and Q-CSMA, BounceNet achieves a mean improvement of $1.2\times$ and $1.33\times$, and a maximum improvement of $1.7\times$ and $1.9\times$ respectively. While these protocols are improve over CSMA and TDMA, they still cannot react and adapt quickly enough to accommodate the fast

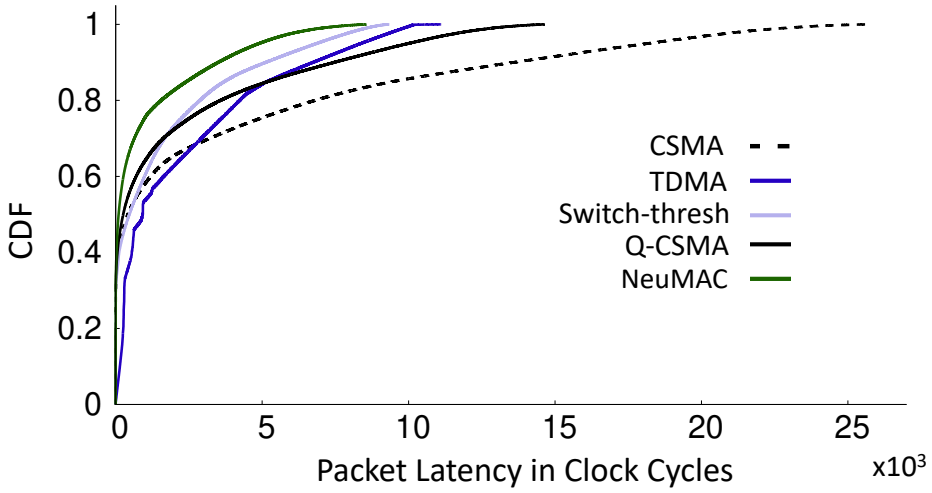


Figure 3.6: CDF of packet latency

changing traffic patterns on the multicore.

(ii) Packet Latency: In Fig. 3.6, we plot the CDF of packet latency due to queuing in the Wireless Network-on-Chip across all applications. It is interesting to note that while at the tail TDMA performs better than CSMA, in the median case TDMA performs significantly worse than CSMA. This is because the high packet latencies at the tail are due to dense traffic in the network which TDMA is better suited for, whereas at the median where traffic is less dense, TDMA leads to much higher packet latencies. BounceNet, on the other hand, is able to adapt to all these different scenarios and provides an improvement in packet latency across all baselines. Over CSMA and TDMA, BounceNet improves median packet latency by $4.11\times$ and $9.18\times$, and improves 90^{th} percentile latency by $3.89\times$ and $1.92\times$ respectively. Over Switch-thresh and Q-CSMA, the gains respectively are $4.66\times$ and $2.56\times$ at the median, and $1.47\times$ and $2.13\times$ at 90^{th} percentile.

(iii) Collisions on Wireless Channel: In Table 4.1 we show $\%$ of collisions on the wireless channel across different benchmarks. We omit TDMA here, since TDMA by design does not suffer from collisions. As observed, BounceNet has significantly fewer collisions than the CSMA algorithms. Switch-thresh is the next best performing protocol, but BounceNet in most cases still has fewer collisions.

B. Single Application End-to-End Execution Speedup:

(i) Speedups over Purely Wired Network-on-Chip: In Table 3.3, we show application speed-ups achieved by BounceNet and the Infinite Capacity baseline respectively, over the purely wired NoC. BounceNet can speed up benchmarks by up to $9.7\times$ for StreamCluster and $6.53\times$ for BFS, and on average provides a

Apps	CSMA	Switch-thresh	Q-CSMA	BounceNet
CC	75.30%	55.58%	76.24%	8.72%
BFS	50.42%	28.28%	49.57%	3.81%
Pagernk	77.36%	11.26%	77.79%	2.19%
SSSP	11.08%	9.48%	9.44%	8.88%
Volrend	44.17%	7.93%	46.11%	2.49%
Strmclstr	62.57%	19.21%	62.69%	31.24%
Canneal	2.55%	2.87%	2.09%	2.04%
Bdytrck	30.5%	29.06%	29.8%	28.87%
Cmmnty	46.76%	32.02%	49.24%	5.8%

Table 3.2: % of Collisions

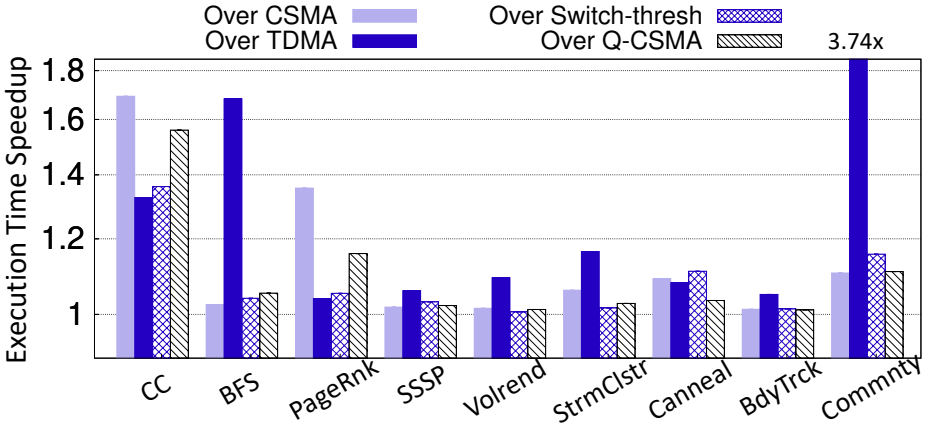


Figure 3.7: Execution Time Results (y axis in logscale)

speedup of $3.42\times$ across benchmarks. Additionally, we see that BounceNet gets very close to the upper bound of the speedup value, achieving up to 99.5% of the maximum speedup possible in the case of BFS, and 98% of the maximum speedup possible on average. This result demonstrates that BounceNet is able to fully exploit the potential offered by the wireless NoC.

(ii) Speedups over Baselines: Fig. 3.7 shows execution time gains of BounceNet over the baselines on the wireless NoC. As can be observed, there is no one baseline protocol that performs well across all applications. While in applications like Pagerank, TDMA performs the best, in other applications such as BFS it is significantly worse. BounceNet, on the other hand, performs well across all benchmarks. In Table 3.4, we see that BounceNet achieves a maximum of 69.18% speedup over CSMA for CC and 274.56% speedup over TDMA for Community, and compared to Switch-thresh and Q-CSMA, BounceNet offers speedups up to 37.09%-55.94%.

Apps	BounceNet	Inf. Cap. baseline	% Achieved
CC	1.96x	2.06x	95%
BFS	6.53x	6.56x	99.5%
Pagerank	1.07x	1.11x	96.4%
SSSP	2.24x	2.25x	99.5%
Volrend	1.32x	1.33x	99.2%
Strmclstr	9.70x	9.77x	99.28%
Canneal	1.14x	1.15x	99.13%
Bodytrack	1.37x	1.38x	99.3%
Community	3.77x	3.82x	98.6%

Table 3.3: Speedups over Purely Wired Network-on-Chip.

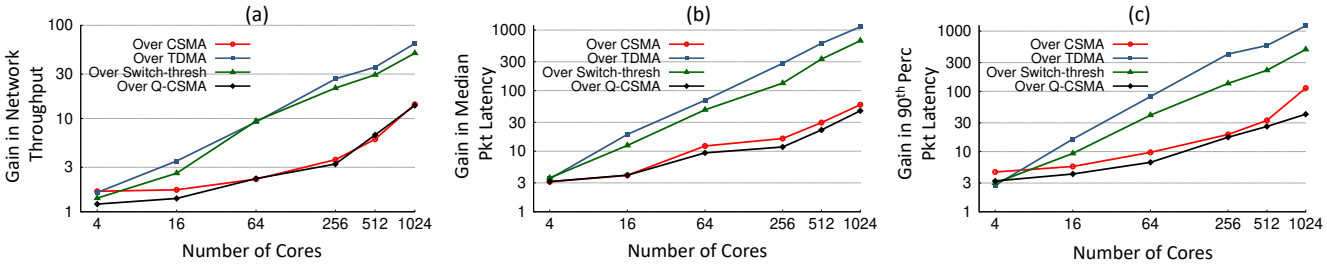


Figure 3.8: Scaling Trends in BounceNet's Gains for (a) Wireless Network Throughput (b) Median Packet Latency and (c) 90^{th} Percentile Packet Latency

C. Multi-Application Jobs: In Table. 3.5, we present execution time speedup results for multiapplication runs on the multicore. For each run, we randomly choose one application among the 9, and execute it using either 4, 16 or 32 threads. We choose a sufficient number of applications such that all 64 cores are utilized, and in total we test on 100 different multiapplication jobsets. Note that the BounceNet agent was never explicitly trained on such multiapplication traffic traces. From Table. 3.5, we can see that BounceNet's gains increase over the baselines compared to single benchmark experiments (Table. 3.4), and goes as high as $6.15\times$ (515.04%) speedup over TDMA. These higher gains in multiapplication jobsets can be attributed to the more complex nature of packet dependencies between threads, which BounceNet can exploit to further speed up execution time as illustrated in Section 3.2.

C. Lossy Networks: To evaluate BounceNet's robustness to varying channel conditions, we conduct experiments in lossy network settings. We vary the packet loss rates in the wireless NoC from 0% up to 10%, and in the event of a loss, the packet is retransmitted. In Fig. 3.9, we compare the average application speedup achieved over the baselines as the loss rate increases. We observe that BounceNet

Speedups	CSMA	TDMA	Switch-thresh	Q-CSMA
Max	69.18%	274.56%	37.09%	55.94%
Min	1.26%	4.88%	0.63%	1.12%
Mean	18.21%	46.90%	9.73%	11.94%

Table 3.4: Summary of Execution Time Speedups by BounceNet. The per-application speedups are shown in Fig. 3.7.

Speedups	CSMA	TDMA	Switch-thresh	Q-CSMA
Max	93.18%	515.04%	48.16%	26.78%
Min	13.3%	24.72%	4.41%	5.82%
Mean	33.93%	166.32%	19.97%	17.48%

Table 3.5: Summary of Execution Time Speedups by BounceNet for Multiapplication runs

is able to generalize very well to varying channel conditions and loss rates, and can maintain the same gains over the baselines throughout. Note that BounceNet was never trained explicitly for lossy network settings. Despite this, it is able to generalize since it can implicitly infer the channel conditions from the channel activity like increased number of collisions.

We also test BounceNet's sensitivity to errors in the observed state caused by packet losses at the BounceNet agent's transceiver during the *"Listening Interval"*. We conduct experiments where we vary the packet loss rate from 0% to 2% in order to introduce noise in the observed state. We find that even under 2% loss rate, BounceNet's suffers a median performance degradation of only 0.85% across all benchmarks compared to its performance with perfect state information.

D. Scaling Trends: We believe that a learning based approach like BounceNet can greatly benefit the wireless NoC performance as the number of cores scale to thousands of cores. To demonstrate this we show the gains that BounceNet achieves over baseline protocols for different metrics as the cores vary from 4 to 1024 in Fig. 3.8. Since multi2sim and other architectural simulators cannot scale beyond a hundred cores, we evaluate these results in our custom simulator by training a separate BounceNet model for each core count. From Fig. 3.8, we can see that BounceNet's gains over the baselines scale favorably with the number of cores. This is because BounceNet is able to generate fine-grained MAC protocols by controlling the actions of each core individually, and thus can generate highly optimized protocols that improve substantially upon the baselines at high core counts.

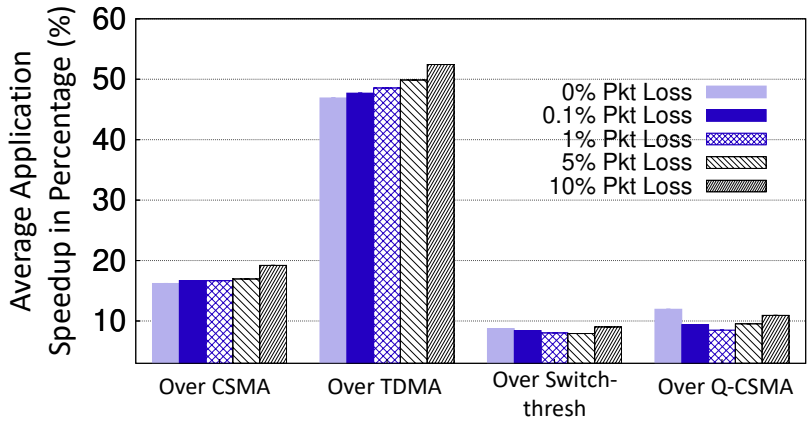


Figure 3.9: Effect of Packet losses on BounceNet's application speedup performance compared to Baselines.

3.7 Related Work

A. Wireless Network-on-Chip Protocols: The majority of past networking research on wireless NoC does not leverage the broadcast nature of wireless to enable instantaneous cache synchronization and instead focuses on using wireless only between far apart cores to reduce the latency. These complementary works focus on problems related to optimizing network topology [92, 154, 155], packet routing [156, 93, 157], flow control [158, 159] and improving the reliability of the PHY layer for far apart cores [160, 140, 142]. However, such designs have limited gains over wired NoCs [90]. More recent work in architecture research exploits the broadcast nature of wireless to boost the performance of wireless enable NoCs [153, 97, 81, 80]. These systems either use *contention-free* mechanisms such as token passing [153] or *contention-based* mechanisms such as carrier sense with exponential backoff [161, 97]. The closest to our work are [81, 80] which attempt to adapt to traffic patterns by switching between a CSMA or a token passing protocol based on a preset threshold. However, hand tuning the threshold values is a challenging task and does not provide the flexibility and expressibility of BounceNet to support complex and highly variable traffic patterns.

B. Network-on-Chip Technologies: Past work on wired NoCs proposes the use of deep learning and RL to learn efficient packet routing protocols [162], learn memory access patterns to reduce cache misses [163], and reduce static and dynamic power consumption on an NoC [164]. To the best of our knowledge, ours is the first work that attempts to exploit deep reinforcement learning techniques to generate medium access protocols for Wireless NoCs.

C. Deep Learning in Wireless Networks: Deep RL has recently been applied in wireless networks to optimize duty cycling in sensor networks [165], resource allocation in cellular networks [166, 167], dynamic spectrum access [168, 169], rate adaptation in CSMA networks [170],and control policies at the PHY layer [145]. [171] provides an extensive survey of deep learning in wireless networks. The closest to our work are [105, 172, 173, 174] which use reinforcement learning to modify the backoff parameters in CSMA or decide whether to transmit or not for every packet at every time step. However, such designs are not applicable in the context of wireless NoCs owing to the unique set of constraints imposed by the NoC, such as the much smaller time-scale of operation rendering neural network inference per transmission slot infeasible, the limited SRAM memory to store model parameters and the enormous action space to explore. These constraints require significant redesign to BounceNet's deep RL framework where it has to now generate high-level, versatile and adaptable protocols that can be deployed for thousands of clock cycles, and generating such protocols cannot be reduced to a simple classification task per transmission-slot (e.g. transmit or not).

3.8 Limitations and Discussion

Some points are worth noting: First, given the enormous costs and engineering efforts involved in prototyping a full chip with integrated processors, memory, and NoC, it is outside the scope of this work to implement BounceNet in hardware. As a result, we evaluate BounceNet on a full-system cycle-accurate architectural simulator, as is the norm among computer architecture researchers. These full-system simulators exhaustively model all components of a CPU and also ensure that all timing dependencies are simulated accurately [113]. As a result, the trends and insights obtained from such architectural simulations often carry over to full fledged prototypes. Moreover, the wireless channel in this WNoC application domain is in fact very stable as opposed to WLAN channels which are extremely dynamic. This is because the multicore is isolated in a chip package, and the wireless channel can be precisely measured and characterized, thus allowing compensation for multipath fading and other artifacts. As a result, the wireless BER in these environments can be as low as 10^{-16} [91], making such a simulation based evaluation representative.

Second, in parallel programming for multicore processors, programmers today try hard to avoid broadcast transmissions as the overhead of running the cache coherency protocol is high. With wireless NoC, the overhead of broadcast traffic is now limited which opens the door to rewriting applications in a manner that embraces broadcast, and can in turn benefit even more from an adaptive protocol like BounceNet.

Lastly, in this chapter we focus on the MAC layer since it is considered a roadblock to realize the full potential of wireless NoCs. However, studying the challenges and opportunities at the other layers such as PHY remains exciting and promising avenue which we leave for future work.

Chapter 4

ENABLING IOT SELF-LOCALIZATION USING AMBIENT 5G SIGNALS

4.1 Introduction

Recent years have witnessed a tremendous growth in the number of connected IoT devices, with surveys projecting up to 31 billion deployed IoT nodes by 2030 [175]. With such ubiquitous deployment of IoT nodes, the ability to localize and track these nodes with high accuracy is essential for many applications. For example, in data-driven agriculture, it can enable real time micro-climate monitoring and livestock tracking [176]. In smart cities, IoT sensors are deployed throughout the city for tasks such as air quality monitoring, tracking buses, trains, and cars, and monitoring the structural health of infrastructure [177]. In the era of Industry 4.0, it can also enable wide area inventory tracking and facilitate factory automation [178].

Today, the most prevalent outdoors localization technology is GPS which is mainly used in cars and mobile phones. However, off-the-self GPS chips can consume about the same power as the entire IoT device, thus reducing the battery life to half, in addition to the extra hardware costs [179]. Due to this, past work has proposed the use of cellular networks or dedicated IoT base stations for localization [180, 181]. These solutions, however, either achieve very low resolution of 100s of meters [180, 182] or require active participation of the base stations to jointly compute the location or tightly synchronize the base stations [181, 183, 184]. Realizing such solutions in practice requires the cooperation of cellular providers to bear the additional cost of modifying the base stations and a back end server to support the localization feature.

In this chapter, *we ask whether an IoT device can accurately localize itself simply by listening to ambient 5G cellular signals, without any coordination with the 5G base stations?* Doing so would allow us to easily deploy self-localizing IoT nodes is wide areas without the need to modify the cellular base stations or deploy new base stations for localization.

5G cellular networks present unique opportunities for enabling accurate localization. First, the small

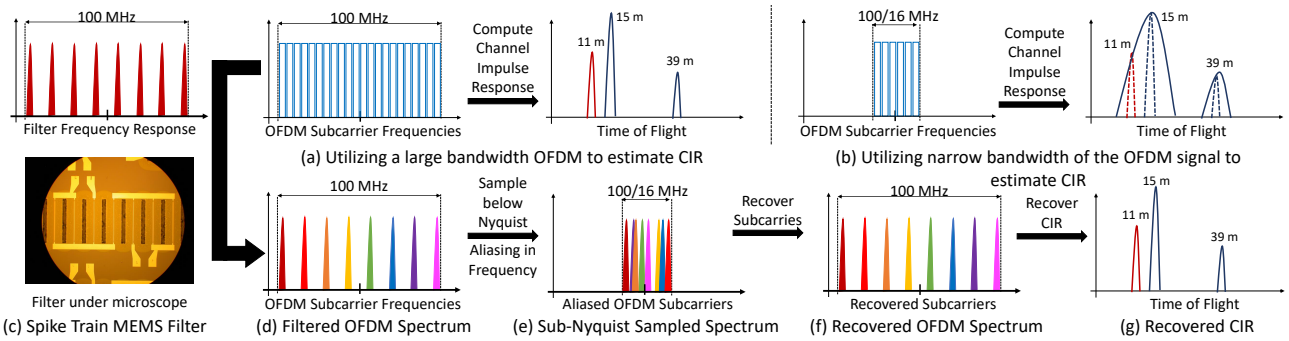


Figure 4.1: ISLA's pipeline. (a) wideband OFDM signal and its corresponding CIR. (b) narrowband OFDM signal and its corresponding lower resolution CIR. (c) ISLA's spike train MEMS filter that sparsifies the wideband signal. (d-f) follow the signal journey through ISLA's pipeline that recovers the original CIR.

cell architecture in 5G networks will lead to a very high density of 5G base stations, with up to 40 to 50 base stations deployed per square km [185], thereby allowing us to leverage more anchor points in the network for increased localization accuracy. Second, the 5G standard is designed to support very high data rates and can have OFDM signals spanning up to 100 MHz in bandwidth in the sub-6 GHz frequency range, and up to 400 MHz bandwidth in the mmWave frequency range [186]. Such large bandwidth can be used for accurate localization. To see how, consider the 5G OFDM signal shown in Fig. 4.1(a) where data bits are encoded in N frequency subcarriers. We can use the preamble which contains known bits to compute the channel impulse response (CIR) by taking an inverse FFT. The CIR in Fig. 4.1(a) shows the Time-of-Flight (ToF) of different signal paths. Estimating the ToF from few base stations allows us to localize the device. The larger the bandwidth of the signal, the higher the resolution. In fact, we can achieve a resolution of 3 meters for 100 MHz and 0.75 meters for 400 MHz signals.[1]

Leveraging these opportunities, however, is challenging since power-constrained and low-cost IoT nodes cannot capture the large bandwidth of the 5G signals. They are equipped with low-power and low-speed Analog-to-Digital Converters (ADCs) that can only capture a narrow bandwidth. In fact, while IoT has been one of the cornerstone applications in the design of 5G, it is only supported in narrowband chunks for low data rate applications [187, 188]. Therefore, while the 5G standard does allocate higher bandwidth (up to 400 MHz) for mobile broadband and high data rate applications, IoT nodes can capture only a very small fraction of this bandwidth ($\sim$ 20$\times$ smaller [186]). As a result, they significantly lose out on the ToF resolution that was made possible by the high bandwidth 5G signals as shown in Fig. 4.1(b). Moreover, it is infeasible to measure the absolute time-of-flight without any coordination or synchronization with the base stations.

[1]The resolution is computed as c/B where c is the speed of light and B is the bandwidth of the signal.

In this chapter, we present ISLA, a system that enables **I**oT **S**elf-**L**ocalization using **A**mbient 5G signals. ISLA does not require any coordination with or modifications to the base stations. The key enabler of ISLA is the use of MEMS (micro-electro-mechanical-system) acoustic resonators. Past work [189, 190] has demonstrated that we can use such MEMS resonators to design new kinds of RF filters that look like a spike-train in the frequency domain, as shown in Fig. 4.1(c). To understand how we can leverage such MEMS spike-train filters, consider the 5G OFDM signal shown in Fig. 4.1(a). Passing this signal through the filter allows us to keep a few subcarriers of the wideband OFDM symbol while suppressing all other subcarriers as shown in Fig. 4.1(d). There are two important features of the resulting signal: (1) Since the remaining subcarriers that are passed by the filter span the entire wideband, we should, in principle, be able to recover the channel impulse response at the same high resolution of the original signal. (2) Since the remaining subcarriers create a sparse signal in the frequency domain, it should be possible to recover these subcarriers by sampling the signal below the Nyquist sampling rate using the same low-power low-speed ADCs on the IoT nodes.[2]

However, recovering the channel impulse response from a signal sampled with the low-speed ADCs is non-trivial. First, sampling the signal below the Nyquist rate leads to aliasing in the frequency domain as shown in Fig. 4.1(e). Some subcarriers might collide by aliasing on top of each other making it hard to recover these subcarriers. Past work in sparse recovery addresses this problem by using two co-prime subsampling rates [191]. Unfortunately, we do not have the flexibility to choose co-prime subsampling factors. In fact, since the number of OFDM subcarriers in the 5G standard is a power of 2 (e.g. 1024, 2048, 4096), we can only subsample the signal by powers of 2 otherwise the values of the subcarriers will be corrupted as we prove in section 4.5.[3] To address this, we carefully co-design the MEMS hardware with the recovery algorithm. In particular, we jointly optimize the filter shape (spacing between peaks, width of each peak, frequency span) with the subsampling rate to minimize the number of colliding OFDM subcarriers as we describe in detail in section 4.5.

Second, the recovered OFDM subcarriers are not uniformly distributed across the wideband bandwidth. This is because non-idealities in the MEMS filter make it hard to design a uniform spike train like the one shown in Fig. 4.1(c). As a result, we can no longer recover the CIR using standard super-resolution algorithms like MUSIC with spatial smoothing [192, 193] as they require uniform measurements. Instead, we formulate an inverse optimization problem that accounts for non-idealities and optimizes the CIR in the continuous time domain to achieve super resolution as described in Sec. 4.5.

[2]Note that the MEMS filter is passive and does not consume any power.

[3]For example, for a 100 MHz OFDM signal, we can only sample at 50 MS/s (2×), 25 MS/s (4×), 12.5 MS/s (8×), 6.25 MS/s (16×), ...

Finally, while the above can provide very precise ToF measurements, these ToF estimates are not going to capture the true time taken by the signal to travel between the base station and the IoT device. This is because the 5G base stations are not time-synchronized with each other or the IoT device. To localize the device without any synchronization with the base station, ISLA leverages a second antenna on the receiver to compute the differential ToF of the propagation paths. While the absolute ToF measurements are corrupted by synchronization offsets, these offsets are constant across the 2 antennas on the IoT node, and hence can be eliminated by subtracting the measurements from the 2 antennas. Using this differential ToF at the IoT receiver, we show in section 4.7 that with measurements from four or more base stations, the IoT device can localize itself regardless of its orientation. We integrate our approach into a full system that addresses additional system challenges such as figuring the base station ID and accounting for carrier frequency offsets.

Evaluation: We implemented and evaluated ISLA indoors for microbenchmarks and outdoors for overall localization performance. We ran experiments in three outdoor settings:(1) Between campus buildings (52 m$\times$85 m), (2) a large parking lot (240 m$\times$400 m), and (3) an agricultural farm (480 m$\times$860 m). We use USRP X310 radios as base stations that can transmit high-bandwidth packets of 100 MHz. Our custom IoT nodes are equipped with 2 antennas and subsample the 5G signals at 6.25 MS/s which is $16\times$ below the Nyquist rate. We fabricated a MEMS spike-train filter operating at a center frequency of 400 MHz and used it to demonstrate accurate reconstruction of the channel impulse response. However, due to significant interference at the 400 MHz band outdoors in our city, we ran experiments at 1 GHz and applied the filter response in digital. Our results reveal that with 5 base stations in range, ISLA can achieve a median accuracy of 1.58 m on campus, 17.6 m in the parking lot, and 37.8 m in the farm where the IoT node can be as much as 500 meters away from most base stations. For the parking lot testbed, the accuracy improves to 9.27 m with 15 base stations and 4.26 m with 25 base stations in range. We compare ISLA's localization approach with several baselines [192, 194, 180] and show up to $4-11\times$ higher localization accuracy. Finally, we show that ISLA achieves a comparable performance to having a full 100 MHz receiver while using a $16\times$ lower sampling rate.

Contributions: We make the following contributions:

- We present, to the best of our knowledge, the first system that allows IoT nodes to localize themselves using ambient 5G signals without any coordination with the base stations.
- We demonstrate the ability to reduce the sampling rate by $16\times$ while retaining the benefits of high bandwidth 5G signals by leveraging recent advances in MEMS RF filters.
- We implement and evaluate ISLA to demonstrate accurate localization in 3 outdoor settings.

4.2 Related Work

Localization has been extensively studied in cellular, WiFi, and IoT networks. Our work differs from past research in that it is the first to enable self-localization using ambient 5G signals without requiring coordination with the base stations.

A. Cellular Based Localization: Several studies [195, 182, 180, 196, 197] have proposed to use nearby cell tower information and statistics in order to localize a mobile device. These methods, however, have a median accuracy of around 100 to 500 meters, and are mostly useful for very coarse localization. To improve localization accuracy, [198, 199] propose to combine WiFi APs with cellular base stations. Despite their relatively higher accuracy, these methods require fingerprinting the surroundings and as such require extensive training and do not generalize to new locations. More recent work exploits massive MIMO and millimeter wave for localization in 5G [200, 31, 201]. However, all of this work requires coordination with base stations and assumes the devices can capture the entire bandwidth of the 5G signals which does not work for IoT devices.

B. IoT Based Localization: [179] leverages TV whitespaces to achieve high localization accuracy for LoRA IoT devices. However, it requires all base stations to be tightly synchronized at the physical layer (time and phase) in order to measure TDoA (Time Difference of Arrival). Recent work [181] designs low power backscatter devices that leverage LoRa for localization to achieve high accuracy. However, the system mainly targets indoor applications where software radios can be deployed as base stations to sample the I/Q of the signal and localize the IoT node. Moreover, its current system design [181] supports only a single node. The authors of [202] propose an outdoors localization technique for SigFox IoT devices based on fingerprinting. However, as mentioned earlier, fingerprinting requires constant training and cannot scale to new environments. Finally, there is a lot of work on using UWB or RFID nodes for localization [203, 204, 205]. However, these works focus on indoors and short range as the range of UWB and RFIDs is limited to 10-30 meters [206, 207].

C. IoT Self-Localization: LivingIoT [208] enables self-localization on IoT nodes. It designs a miniaturized device that can be carried by a bumblebee and uses backscatter for communication. The node localizes itself by extracting the angle to the Access Point from the amplitude measurements using an envelop detector. The technique, however, requires the APs to switch the phase across two antennas to change the received amplitude at the IoT node, and hence, cannot be applied to 5G without modifying the base stations. [209] enables self-localization by placing a camera on a WISP RFID but only operates within a range of 3.6 m from the RFID reader.

D. WiFi Based Localization: There has been a lot of work on indoor localization using WiFi [183, 192,

210, 211, 212, 171, 213, 193, 194]. The closest to our work are [192, 183, 194], which estimate the channel impulse response (CIR) and time of flight (ToF) from the WiFi access point (AP). Chronos [183] hops between WiFi channels to compute the CIR at high resolution. However, it requires tight timing coordination with the AP to compensate for carrier frequency offset (CFO) and ensure phase coherence across the measurements. ISLA, on the other hand, captures measurements from many frequencies across a wideband without hopping by using the MEMS filter, and hence, does not require any coordination with the base stations. SpotFi [192] combines measurements across antennas with large WiFi bandwidth to separate Line of Sight (LoS) path from multipath reflections in the CIR using MUSIC along two dimensions: ToF and Angle of Arrival (AoA). mD-Track [194] also incorporates Doppler shifts and Angle of Departure (AoD) in addition to ToF and AoA and iteratively refines the CIR to achieve a better estimate of the LoS path. In section 4.10, we adapt SpotFi's and mD-Track's CIR estimation algorithms to our setting and demonstrate that ISLA's algorithm achieves $4-11\times$ higher accuracy. It is worth noting, however, that for our application, these past works cannot benefit from the doppler or AoA/AoD dimensions.

E. MEMS Filter: Recent work has used MEMS spike-train filters for the application of wideband spectrum sensing [190]. However, [190] can only detect signal power at different frequencies and cannot recover complex I and Q samples needed for estimating the CIR. Furthermore, [190] deals with collisions resulting from aliasing by using co-prime sub-sampling rates. Such approach does not apply in the context of 5G OFDM signals, since, as we show in section 4.5 the sub-sampling factor can only be a power of 2. ISLA instead co-designs the hardware filter together with sampling rate to avoid collisions.

4.3 Background

A. Spike-Train MEMS Filters: Our work builds on recent advances in MEMS RF filters. MEMS filters can work between a few MHz and 30 GHz and can be integrated with ICs to form a chip-scale RF front-end solution for IoT devices. Past work on MEMS RF filters optimize for filters with a single passband [214, 215], however, the MEMS filter used by ISLA leverages MEMS resonators that have an assortment of equally spaced resonance frequencies to create a spike train in the frequency domain as shown in Fig. 4.1(c).

A MEMS filter works by leveraging the inverse piezoelectric effect to convert RF signals into acoustic vibrations for filtering and processing. It then converts acoustic waves in the device back to the RF signals through piezoelectric effect. In this process, the frequency filtering is achieved because not all

frequencies can be efficiently converted between RF and acoustic domains. Frequencies that match the resonance frequencies of the piezoelectric structure can go through the conversions with little loss, while other frequencies are filtered out. Hence, the spike train frequencies can be designed by changing the dimension of the piezoelectric material in the MEMS device as well as the placement of electrodes shown under the microscope in Fig. 4.1(c).

B. Wireless Channel Impulse Response (CIR): The wireless channel can be modeled as the superposition of the signal along all the different paths it takes to travel from the transmitter to the receiver. The channel at frequency f_i can be written as: $h_i = \sum_{l=1}^{L} a_l \exp^{-j2\pi f_i d_l/c}$, where L is the number of propagation paths between the transceivers, d_l is the distance traversed by path l, a_l is the complex path attenuation of path l, and c is the speed of light.

In OFDM systems, data is transmitted over multiple frequency subcarriers $\{f_0, \dots f_{N-1}\}$. If the frequency spacing between these subcarriers is Δf, then the bandwidth spanned by the signal is $B = \Delta f \times (N-1)$. Now, given the channel measurements $\{h_0, \dots h_{N-1}\}$ across these frequencies, the Channel Impulse Response (CIR) can be computed as the inverse FFT of the channel measurements.

$$CIR(\tau) = \sum_{n=0}^{N-1} \Big(\sum_{l=1}^{L} a_l \exp^{-j2\pi \frac{d_l}{c} f_n} \Big) \exp^{j2\pi\tau f_n} \tag{4.1}$$

where $\tau = \{\frac{0}{B}, \dots \frac{(N-1)}{B}\}$ seconds. There are two important things to note here. First, the resolution in Time-of-Flight in the CIR is $1/B$ seconds, that is inversely proportional to the bandwidth B. Hence, larger bandwidth results in higher ToF resolution and more accurate ranging. Second, the maximum unambiguous ToF that can be measured from the CIR is $\frac{(N-1)}{B} = 1/\Delta f$ seconds. This means, if some physical propagation path in the environment has ToF $> 1/\Delta f$ then it would alias and appear at a different tap value in the estimated CIR in Eq. 4.1. For 5G OFDM signal with $B = 100$ MHz bandwidth and $\Delta f = 60$ kHz , we have a resolution of 10 ns (3 meters) and a range of $16.6\ \mu s$ (5 km).

4.4 System Overview

ISLA enables self-localization on narrowband IoT devices by leveraging the MEMS spike-train filter to capture ambient wideband 5G signals. ISLAconsists of 3 main components:

(1) Capturing the wideband 5G OFDM signal using the MEMS filter: The received 5G signal is passed through the MEMS filter which samples the OFDM symbol in the frequency domain. Specifically, the MEMS filter passes the OFDM frequency bins that align with the filter passbands while sup-

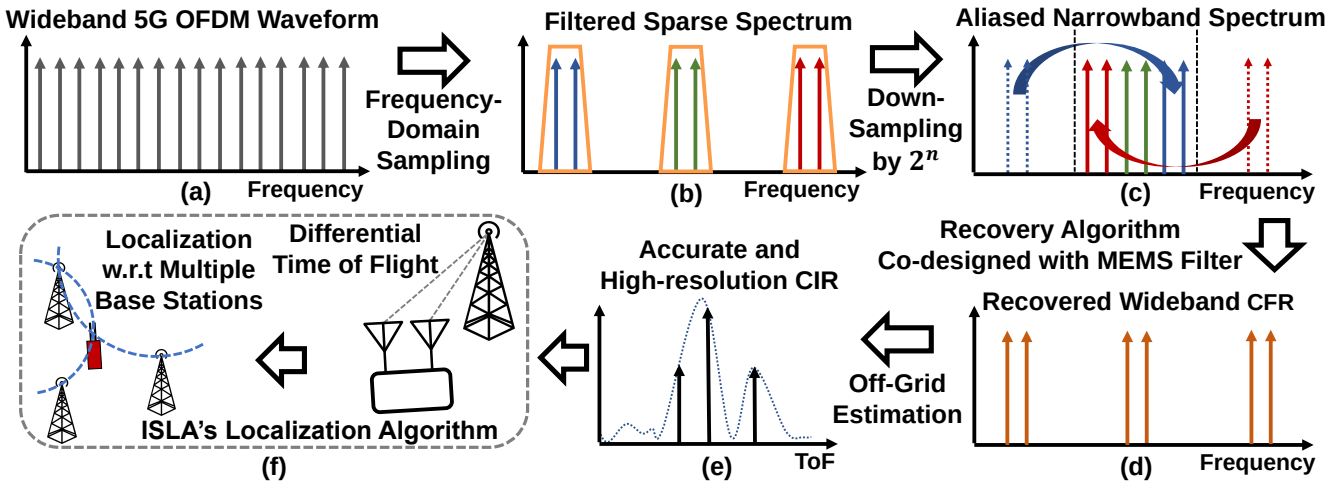


Figure 4.2: Overview showing the flow of ISLA's system

pressing all other frequency bins. The resulting output from the filter is a sparse spectrum as shown in Fig. 4.2(b). This sparse signal is then subsampled by the narrowband IoT device significantly below the Nyquist rate ($16\times$ lower) which results in aliasing the remaining subcarriers into the narrowband as shown in Fig. 4.2(c). We co-design the filter hardware with the recovery algorithm to easily reconstruct the wideband OFDM subcarriers as we describe in section 4.5.

(2) Super-Resolution CIR Estimation: Using the recovered wideband channel measurements, ISLA then reconstructs a high resolution Channel Impulse Response (CIR) by leveraging its super-resolution algorithm which estimates the off-grid positions of the propagation paths as described in Section 4.6. This high-resolution CIR allows ISLA to filter out the LoS path from the multipath in the channel for high resolution time-of-flight estimation as shown in Fig. 4.2(e).

(3) Localization Algorithm: Since the IoT node is not synchronized with the base station, the measured ToF will be corrupted by a timing offset. To address this, ISLA leverages two antennas on the IoT device and computes the differential CIR across the antennas to eliminate the synchronization offsets. This results in the locus of the IoT device to lie on a circle that is defined by the locations of the base stations and the angle subtended by the base stations at the IoT device's location, as we explain in Section 4.7. Thus, by looking at the intersection of such circles, we can accurately infer the position of the IoT device as shown in Fig. 4.2(f). Finally, we show how to integrate ISLA with the 5G-NR standard by addressing additional system challenges in section 4.8.

4.5 Capturing 5G Signals Using MEMS Filter

ISLA leverages the MEMS spike-train filters to capture the wideband channel measurements on a narrowband receiver. We explain this sensing process through Fig. 4.2. Consider a preamble OFDM symbol transmitted from the base station with N subcarrier frequencies at $\{f_0, \ldots, f_{N-1}\}$, shown in Fig. 4.2(a). Let the received time domain symbol be $x(t)$ and its frequency domain representation be $X(f)$. We have $X(f) = \sum_{n=0}^{N-1} c_n h_n \delta(f - f_n)$, where c_n are the data bits modulated onto the subcarriers and h_n are the channel values at f_n. We want to extract this channel information to compute the Channel Impulse Response $CIR(\tau)$. Since the preamble bits c_n are known, we can compensate for c_n and compute the $CIR(\tau)$ by taking an IFFT of the channel values h_n. However, this requires capturing the entire bandwidth of the 5G OFDM signal. Our goal is to recover the CIR using a narrowbandwith. To do so, we leverage the MEMS spike-train filter.

The spike-train filter response is made up of uniformly spaced passbands as shown in Fig. 4.2(b). The spike-train filter serves to sparsify the OFDM symbol by selectively passing subcarriers that fall inside the MEMS passbands, while suppressing all other frequencies. Let the set of frequencies passed by the spike-train be indexed by M. Then, the frequency domain of the signal $\tilde{X}(f)$ ($\tilde{x}(t)$ in the time domain) after passing through the spike-train filter will be $\tilde{X}(f) = \sum_{i \in M} c_i h_i \delta(f - f_i)$.

This sparse spectrum is shown in Fig. 4.2(b). Next, the IoT receiver subsamples the signal $\tilde{x}(t)$ using a low-speed ADC that samples at a rate $R = B/P$, where B is the bandwidth of the transmitted symbol and P is an integer corresponding to the subsampling factor. Let $y(t)$ be the subsampled signal, that is, $y(t) = \tilde{x}(P \times t)$, and let $Y(f)$ be its frequency domain representation. Then $Y(f)$ is an aliased version of $\tilde{X}(f)$:

$$Y(f) = \sum_{i=0}^{P-1} \tilde{X}(f + iR) \tag{4.2}$$

$Y(f)$ will cover a narrow bandwidth equal to R MHz as depicted in Fig. 4.2(c). The process of aliasing is as follows. Any frequency f_j, $j \in M$, that falls outside the narrowband of the IoT device, will alias onto the frequency bin $\tilde{f}_j$ inside the narrowband after subsampling, such that $f_j - \tilde{f}_j = z \times R$, where z is some integer. Note that for every f_j, we have a unique $\tilde{f}_j$. So given the measurement at the aliased frequency $\tilde{f}_j$, we can potentially recover the channel value h_j at the corresponding unaliased frequency f_j.

However, recovering these channel values from the aliased spectrum is non-trivial because multiple of the frequency subcarriers passed by the spike-train filter may collide by aliasing on top of each other and summing up. This is unfavorable since now we are unable to extract the channel values for any of

the colliding frequencies. Past work addresses this by leveraging multiple co-prime subsampling factors, which ensures that the same frequencies do not collide repeatedly.

Unfortunately, we do not have such flexibility to choose any sub-sampling factor here. This is because in order to recover the channel value h_j from the aliased frequency $\tilde{f}_j$, we need to ensure that the complex scaling factor $c_j \times h_j$ encoded on subcarrier f_j remains preserved upon aliasing. This is crucial because the wireless channel information is contained inside this scaling factor. The following lemma states the condition that ensures this:

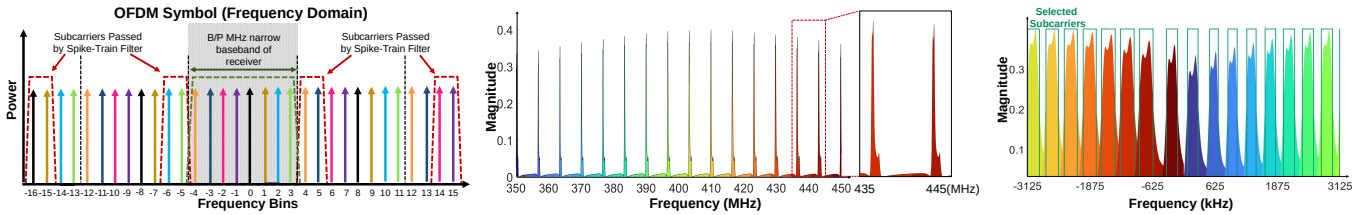
Lemma 4.1 *For a sub-sampling factor P and N OFDM subcarriers, the complex valued scaling factors for each subcarrier will be preserved upon aliasing if $N = z \times P$, for some integer z, given the aliasing results in no collisions.*

The proof for the above lemma is in Appendix D. Thus, to be able to recover channel values, we are restricted to subsample the signal by an integer factor of N. Further, since the OFDM subcarriers in the 5G standard are set to powers of 2, we can only subsample the wideband signal by powers of 2.

Due to this lack of choice in subsampling factors, we instead shift our focus on designing the spike-train filter such that the frequencies passed by the filter do not collide upon aliasing. We achieve this by leveraging the structured periodic sparsity of the spike-train, and design a filter that ensures no collisions for the given subsampling factor P.

Doing so significantly simplifies our recovery algorithm. In particular, given that (1) the frequency response of the spike-train filter and its collision-free aliasing patterns are known, and that (2) the scaling factors at the frequency subcarriers remain preserved upon aliasing, we can now simply rearrange the frequencies in Y(f) to their corresponding unaliased frequency positions as shown in Fig. 4.2(d). Further, we can extract the channel values at these unaliased frequencies by dividing the complex scaling factor $c_j \times h_j$ by the known preamble bit c_j. Thus, by leveraging the spike-train filter, ISLA is able to extract wideband channel values on a narrow band IoT device. Next, we discuss the design parameters of the spike-train filter that ensures no collisions.

Spike-Train Filter Design: We explain the spike-train filter design with a specific example, shown in Fig. 4.3(a). Let the wideband transmitted OFDM signal (B MHz bandwidth) be comprised of 32 frequency subcarriers, indexed from -16 to 15, with 0 denoting the carrier frequency bin. From Lemma 4.1, we want the subsampling factor P to divide $N = 32$. So, we choose $P = 4$, that is, the IoT receiver subsamples the signal by $4\times$. This implies that the IoT receiver is only able to capture $\frac{N}{P} = 8$ frequency bins centered around the carrier frequency as shown by the shaded region in Fig. 4.3(a). Let this narrow band set of frequencies be denoted as f_{NB}.



(a) Spike-Train Filter Parameters (b) Spike-Train Filter Frequency Response (c) Aliased Frequency Response

Figure 4.3: (a) MEMS Filter Parameters that ensure zero collisions while recovering maximum channel information. (b) Frequency response of MEMS spike-train filter. (c) Aliasing pattern of spike-train filter frequency response.

Recall that when you subsample a B MHz signal by $P\times$, then all frequency subcarriers spaced by $R = \frac{B}{P}$ MHz will alias onto the same frequency bin in the narrow band spectrum. Here, this translates into all frequencies spaced by 8 subcarriers aliasing onto the same narrowband bin. This is depicted in Fig. 4.3(a) through the color coding scheme. For instance, the subcarriers at $\{-9, -1, 7, 15\}$ (represented as purple colored) would all appear at frequency bin -1 in the narrow band spectrum upon aliasing. For a given subcarrier k in the narrow band spectrum, that is, $k \in \{-4, \ldots, 3\}$, let us denote the set of subcarriers that would alias into k as I_k. So we have $I_{-1} = \{-9, -1, 7, 15\}$.

The spike-train filter will selectively pass frequency subcarriers in the wideband OFDM signal, which after aliasing can be recovered from the narrow band signal at the receiver. Let the set of frequency subcarriers passed by the spike-train filter be denoted by f_M, where $M \in [-15, \ldots, 16]$. We want the following conditions to hold:

1. *No Collisions:* To ensure that we can successfully recover the wideband channels, no two subcarriers in f_M should alias and collide in the same narrowband frequency bin upon subsampling. To achieve this, the spike-train filter must satisfy: *For any set I_k where $k \in \{-4, \ldots, 3\}$, f_M must contain at most one subcarrier from I_k.*
2. *Extract Maximum Possible Channel Values:* Given that the narrowband spectrum spans 8 frequency subcarriers, this means that the receiver can successfully recover at most 8 channel values after subsampling. In the presence of noise, we want to recover as many channel measurements as possible for robustness. Hence, every narrowband subcarrier in f_{NB} should yield one channel measurement from the wideband signal. This translates to: *For any set I_k where $k \in \{-4, \ldots, 3\}$, f_M must contain at least one frequency subcarrier from I_k.*

   1 and 2 put together, dictates that the spike-train filter should pass *exactly one* frequency subcarrier from each I_k.
3. *Span the Wideband OFDM symbol:* To retain the high ToF resolution, we want the set of frequencies

in f_M to span the entire wideband signal.

The above conditions can be met leveraging the structured sparsity in the spike-train filter response. Specifically, we can design three key parameters of the spike-train filter: (1) spacing between consecutive spikes ΔF, (2) width of the spikes ΔS, and (3) the starting frequency subcarrier f_M^0 in the spike-train, to follow Lemma 4.2. We prove in Appendix D that such a filter response satisfies the above conditions.

Lemma 4.2 *Consider an OFDM symbol with N frequency subcarriers, indexed as $\{f_{\frac{-N}{2}}, \ldots, 0, \ldots, f_{\frac{N}{2}-1}\}$ with inter-frequency spacing of Δf, and a narrowband receiver that subsamples by $P\times$. If P^2 divides N, then the ideal filter parameters that meet all three requirements are: (1) $f_M^0 = f_{\frac{-N}{2}}$, (2) $\left(\frac{N}{P^2} - 1\right) \times \Delta f < \Delta S < \frac{N}{P^2} \times \Delta f$, and (3) $\Delta F = \frac{N}{P}(1 + \frac{1}{P}) \times \Delta f$.*

Furthermore, we can achieve the required filter response by designing the topology of the MEMS resonators, which we explain in more details in Appendix E.

In Fig. 4.3(a), we show the ideal frequency response of the spike-train filter designed with the above parameters as the red dotted line. In theory, such a filter should allow us to leverage all f_{NB} subcarriers to recover the wideband channel measurements from the aliased signal. However, in practice, MEMS spike-train filters are non-ideal i.e., the roll-off of the passband boundaries are not as sharp as perfect rectangular functions, the spikes are not perfectly equally spaced, and the passband widths are not identical. These imperfections can be observed in the frequency response shown in Fig. 4.3(b). As a result of these non-idealities, there will still be collisions at the boundary regions of the spikes after aliasing, as shown in Fig. 4.3(c). To avoid collisions from polluting our CIR estimates, we only consider the subcarriers that do not collide as shown in Fig. 4.3(c). However, this results in non-uniform sampling of the OFDM subcarriers across the wideband channel. In sec. 4.6, we show how to leverage ISLA's super-resolution algorithm to recover high resolution CIR estimates from these non-uniform channel measurements.

Tradeoff Between Range and Resolution: Recall from section 4.3 that the resolution in ToF depends on bandwidth, whereas the maximum unambiguous ToF (range) depends on the inter-frequency spacing between channel measurements. In the 5G OFDM signal with bandwidth $B = 100$ MHz and subcarrier spacing $\Delta f = 60kHz$, ISLA is able to retain the high ToF resolution of 10 ns (3 m) by collecting wideband channel measurements that span the entire 100 MHz. However, in doing so, the frequency spacing between the channel measurements in ISLA increases, thus reducing the maximum ToF range. Specifically, the frequency spacing increases by $P = 16\times$ in ISLA, thus reducing the maximum range from 5 km to 312 meters. This is an issue since now it becomes difficult to identify the LoS path from the CIR for localization. You could have the case where the LoS path is at 200 meters, but a reflected

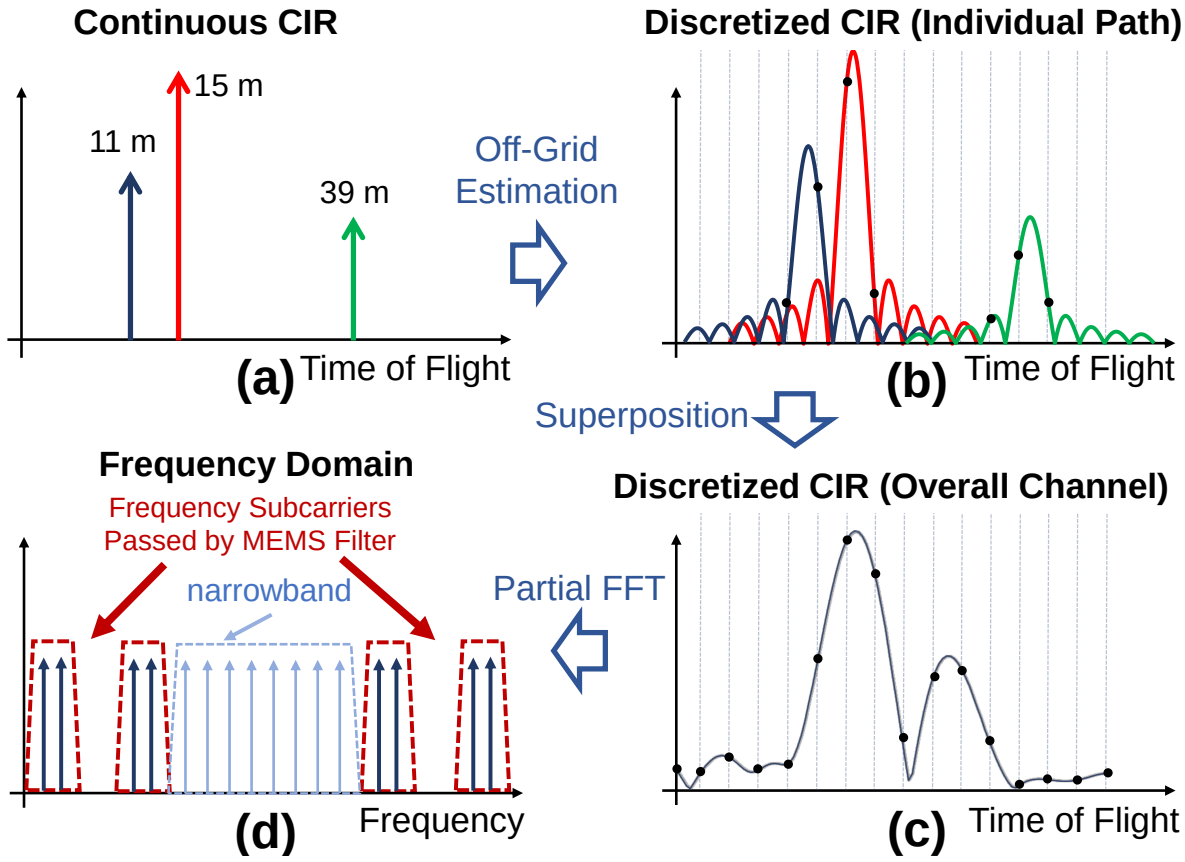


Figure 4.4: Signal paths to measured channel forward function

path at 400 meters aliases and appears at the bin corresponding to 88 meters in the CIR. Thus, you cannot simply pick the first peak as LoS.

To address this, ISLA combines the wideband channel measurements from the spike-train filter, h_M, with the narrowband channel measurements h_{NB} collected at the subcarriers f_{NB}, and formulates a joint optimization with both these channels to estimate the CIR. Since the narrowband channel measurements h_{NB} retain the same subcarrier spacing of $\Delta f = 60kHz$, it increases the effective maximum ToF range back to 5 km, thus resolving the LoS ambiguity in the CIR.

4.6 Super-Resolution CIR Estimation

Here, we describe our super-resolution algorithm that can retrieve high resolution ToF estimates τ_l's along with the associated complex attenuations a_l for the L multipath components in the channel. As discussed in Sec. 4.5, the IoT device can recover channel measurements $h_{tot} = h_M \cup h_{NB}$ at the sub-

carriers $f_{tot} = f_M \cup f_{NB}$ where f_M are recovered from the spike-train filter and f_{NB} without the filter. Since these channel values are sampled at non-uniformly spaced frequencies, we cannot apply standard super-resolution algorithms like MUSIC with spatial smoothing [192, 193] as they require uniform measurements. Instead, we optimize for the channel impulse response in the continuous time domain by leveraging an off-grid estimation technique that can estimate high resolution ToF values from the channel information.

We begin by framing this as an inverse problem. We start by modeling the forward operator $\mathcal{F}$: $h_{tot} = \mathcal{F}(\tau_1, \ldots, \tau_L, a_1, \ldots, a_L)$, which maps physical path parameters to the wireless channel. $\mathcal{F}$ comprises of the following distinct transformations, as illustrated in Fig. 4.4:

(1) CIR in Continuous Domain: (Fig. 4.4(a)) Given path parameters $\{\tau_1, \ldots, \tau_L, a_1, \ldots, a_L\}$, the continuous domain CIR can be written as: $CIR_{cont} = \sum_{l=1}^{L} a_l \delta(\tau - \tau_l)$, with each path represented as an impulse positioned at its respective ToF τ_l, and scaled by its complex attenuation a_l.

(2) Off-Grid Estimation: (Fig. 4.4(b)) The OFDM symbol spans a bandwidth B MHz and comprises of N subcarriers. Due to this discretization and truncation in the frequency domain, the observed CIR at the receiver will also be discretized, and computed on the grid defined by τ_g, where $\tau_g = \{\frac{0}{B}, \ldots, \frac{(N-1)}{B}\}$. However, as with most natural signals, the ToFs of the physical propagation paths τ_l will rarely align with this discretized τ_g grid, that is, the τ_l's will lie at an off-grid position. As a result, the leakage from the continuous off-grid CIR component from path l to the discrete CIR grid positions at τ_g can be computed as $CIR^l(\tau_g) = a_l \psi_N(\tau_g - \tau_l)$, where ψ_N is the discretized sinc function defined as:

$$\psi_N(\tau) = \frac{sin\,(\pi\tau)}{sin\left(\frac{\pi\tau}{N}\right)} \exp\left(-\pi j \left(\frac{N-1}{N}\right)\tau\right) \tag{4.3}$$

(3) Superposition: (Fig. 4.4(c)) With multiple propagation paths in the channel, the net observed CIR at the receiver is the sum of the CIR profiles contributed by each propagation path: $CIR^{net}(\tau_g) = \sum_{l=1}^{L} a_l \psi_N(\tau_g - \tau_l)$.

(4) Discrete Fourier Transform: (Fig. 4.4(d)) Finally, the channel h_{tot} can be computed by sampling the corresponding frequencies f_{tot} from the DFT of the superposed CIR. Let us denote the $N \times N$ Fourier matrix as $\mathbf{F}_N$, and let $\mathbf{V}$ be the matrix that chooses the rows corresponding to f_{tot} from $\mathbf{F}_N$. Then we have: $h_{tot} = \mathbf{V}\,\mathbf{F}_N\ CIR^{net}$ where CIR^{net} is a $N \times 1$ dimension vector.

Putting the above four transformations together, the forward operator $\mathcal{F}$ can be expressed as:

$$h_{tot} = \mathcal{F}(\{\tau_l, a_l\}_{l=1}^{L}) = \mathbf{V}\,\mathbf{F}_N \Psi \vec{a} \tag{4.4}$$

where Ψ is a $N \times L$ matrix with $\Psi_{i,j} = \psi_N(\tau_i - \tau_j)$, and $\vec{a}$ is a $L \times 1$ vector comprising the complex

attenuations a_l for each path. Now that we have the forward operator, the inverse problem to retrieve the path parameters from observed channel vector $h^{'}_{tot}$ can be formulated as a L-2 minimization:

$$\{\tau_l^*, a_l^*\}_{l=1}^{L} = \underset{\tau_1,\ldots,\tau_L,a_1,\ldots,a_L}{\arg\min} \|h^{'}_{tot} - \mathbf{VF}_N \Psi \vec{a}\|^2 \tag{4.5}$$

Solving the Optimization: Note that if we are given Ψ, then Eq. 4.5 becomes a linear optimization problem in $\vec{a}$. Thus, given Ψ, the closed form solution for $\vec{a}$ that minimizes Eq. 4.5 is $\vec{a} = (\mathbf{VF}_N \Psi)^{\dagger} h^{'}_{tot}$, where † represents the pseudo-inverse. Thus, the objective function in Eq. 4.5 can be rewritten as:

$$\begin{aligned} &\{\tau_l^*\}_{l=1}^{L} = \underset{\tau_1,\ldots,\tau_L}{\arg\min} \|h^{'}_{tot} - \mathbf{VF}_N \Psi (\mathbf{VF}_N \Psi)^{\dagger} h^{'}_{tot}\|^2 \\ &\text{s.t.} \quad \tau_l \geq 0 \quad \forall\, l \in \{1, 2, \ldots, L\} \end{aligned} \tag{4.6}$$

The objective function is now reduced to just the ToF variables τ_l's. This optimization problem is non-convex and constrained, and we use the well-known interior-point method to solve this [216]. For the initialization point to the optimization algorithm, we use approximate ToF values from the CIR computed by taking the inverse FFT of the observed channel $h^{'}_{tot}$. While these ToF estimates are distorted by the discretization and superpositioning artifacts described previously, it gives a good starting point for the optimization.

Also, note that the number of paths N in the wireless channel is not known a priori. As we keep increasing the number of paths N that the algorithm is initialized with, it keeps finding a better and better fit to the channel data, and after a point, starts overfitting to the noise. In order to avoid overfitting and yet yield accurate estimates for the path parameters, we run the optimization problem multiple times, each time increasing the number of paths it is initialized with by 1. We terminate the algorithm when the decrease in the value of the objective function falls below some threshold ϵ, and set the current value of N to be the number of paths in the channel.

4.7 ISLA's Localization Algorithm

The above off-grid estimation algorithm gives us highly precise ToF estimates for the propagation paths. However, since the 5G base stations are not time synchronized with the IoT device, there is going to be an offset between the sampling clocks in their RF chains. As a result, the measured ToF at the IoT node also includes delays from the sampling time offset (STO) between the different base stations and the IoT node, and hence cannot provide accurate distance estimates.

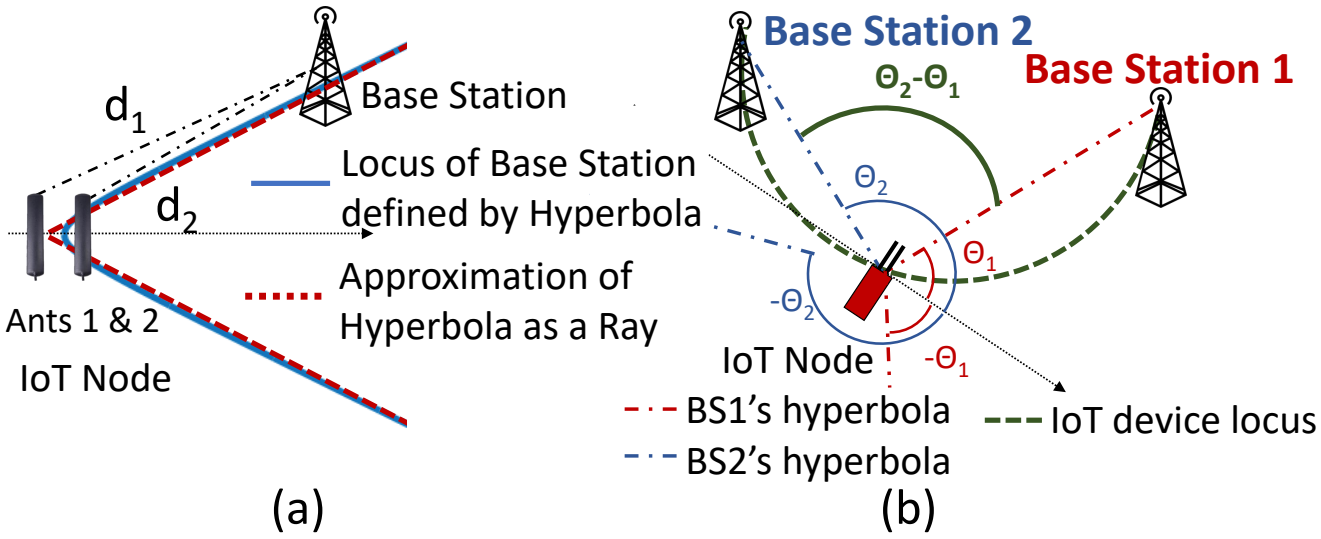


Figure 4.5: ISLA's Localization Algorithm

To address this, ISLA leverages two antennas on the IoT node to compute the differential ToF rather than the absolute. The key idea here is that while the absolute ToF measurements are corrupted by synchronization offsets, these offsets are constant across the two antennas on the IoT node. Hence, the offsets can be eliminated by differencing the two measurements. Let the ToF values to the two antennas be τ_1 and τ_2, and their corresponding distances be d_1 and d_2, as denoted in Fig. 4.5(a). Then the locus of the base station from the IoT device's frame of reference is a hyperbola with the two antennas being the foci, and the difference in distances to the two foci equaling $d_2 - d_1$. At large distances, this hyperbola can be approximated as two rays along the asymptotes of the hyperbola, depicted by the red dashed lines in Fig. 4.5(a).

By overhearing packets from different base stations, the IoT device can infer the locus of each base station to lie on approximated rays originating from the IoT device's location. This is shown in Fig. 4.5(b), where base station 1 can lie on the rays at angles θ_1 or $-\theta_1$, and similarly the base station 2 can lie on the rays at angles θ_2 or $-\theta_2$. Both θ and $-\theta$ are possible, since there is the ambiguity that the signal might have arrived from the front or the back of the device. Given this, we can see that the angle subtended by the two base stations at the location of the IoT device will be $\|\theta_2 - \theta_1\|$, and this is going to be constant irrespective of the orientation of the IoT node. There is ambiguity in that the angle subtended can also be $\|\theta_2 + \theta_1\|$, and we will address this shortly.

Given the angle subtended by the base stations and the known locations of the base stations, according to the Inscribed Angle Theorem, we can determine the locus of the IoT device to lie on the arc of a circle, where the line segment connecting the two base stations is the chord and the corresponding inscribed angle is equal to the angle subtended by the base stations. This is illustrated in Fig. 4.5(b) as the

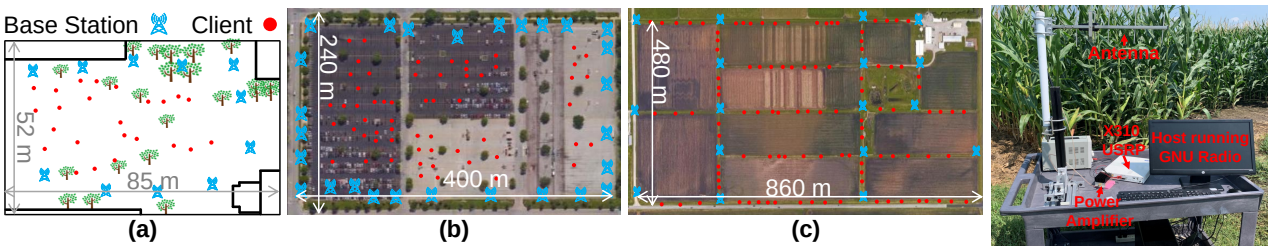


Figure 4.6: **Outdoor Experiment Testbeds:** (a) Campus testbed surrounded by buildings. (b) Parking lot testbed. (c) Agricultural farm testbed. (d) Prototype base station in the agricultural farm testbed.

green dashed arc. Leveraging different pairs of base stations, ISLA can draw multiple such arcs and the intersection points of these arcs will give us the IoT device's location.

Sources of Ambiguity: There are some sources of ambiguity that need to be resolved. First, the angle subtended by the two base stations in Fig. 4.5(b) could also be $\|\theta_2 + \theta_1\|$, and second, the arc drawn with the base stations at the end points could also be pointing towards the north rather than south, as depicted in Fig. 4.5(b). These ambiguities can be resolved easily by leveraging 4 base stations as anchor points. Keeping one base station common, we have three base station pairs which yields three unique arcs. Only the right configurations of angles subtended and arcs drawn will give us a common intersection point for all three arcs. ISLA's localization algorithm tries all configurations and picks the one where all arcs coincide at the same point.

4.8 Integrating ISLA with 5G-NR Standard

Similar to the LTE standard, the 5G-NR packet consists of 10 subframes, each of duration 1 ms [217]. To allow for coherent packet demodulation, the 5G frame appends known preamble bits on each subframe which enables channel estimation and correction across the entire bandwidth of the 5G channel. Additionally, in the first subframe of the packet, the base station also includes all information required by devices to associate with the network, which comprises of the synchronization signals (PSS and SSS frames) for CFO correction and frame timing, and the Base Station ID. To allow every device in the network to receive this critical information, it is always encoded in the narrowest supported bandwidth of the wideband packet, which is 4.32 MHz in the 5G standard [217].

ISLA's hardware circuit, discussed in Section 4.9, is designed such that it can switch between capturing the 6.25 MHz narrowband spectrum, or the wideband spectrum via the spike-train filter. ISLA begins by capturing the first subframe of the 5G packet through its narrowband RF path, and extracts the synchronization frames and base station ID encoded in the narrowband subcarriers of the wideband packet.

Using publicly available databases like [218], ISLA can retrieve the location of the Base Station given its ID. The synchronization frames help eliminate coarse CFO and SFO. From the subsequent subframes, ISLA first estimates the narrowband channel, and then switches to the RF path with the spike-train filter to sense wideband channel. Note that ISLA does not need to meet tight timing constraints to switch since each subframe lasts 1 ms and there are multiple such subframes in each packet that can be leveraged for channel estimation. Thus, ISLA can simply skip a subframe while switching.

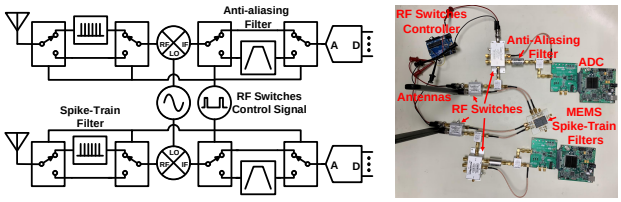
However, because ISLA captures the narrowband channel and wideband channel from different subframes, there is going to be an additional phase accumulation between the two measurements due to residual CFO. To address this, we slightly modify Eq.4.6, and the detailed description for this modification is presented in Appendix F.

4.9 System Implementation

System Design: We have built a prototype ISLA device by combining our MEMS spike-train filter with commodity, off-the-shelf, low-power components. Figure 4.7(a) shows the circuit diagram, and Fig. 4.7(b) shows the actually prototype. It receives ambient 5G transmissions with two antennas followed by identical RF chains. Depending on whether the IoT devices wants to receive the full 100 MHz spectrum using the spike-train filter or the narrowband spectrum, the RF chains can switch between two paths: (1) the received wideband spectrum first be filtered by the MEMS spike-train filter, and then down-converted and sampled without using the anti-aliasing filter. (2) the MEMS spike-train filter is bypassed but the down-converted signal will first go through an anti-aliasing filter before sampling. We select between the two paths using RF switches controlled by a single microcontroller.

Implementation: We fabricated a MEMS spike-train filter at 400 MHz center frequency. However, due to the strong interference from the amateur radios in this band, we were not able to run experiments outdoor using this filter. Hence, the above prototype was only used indoors. In the outdoor experiments, we transmitted in a vacant 100 MHz wide spectrum between 950 and 1050 MHz, and we emulate the IoT radio front-end described above with the MEMS spike-train filters in digital using an X310 USRP software-defined radio (SDR). We would like to note that in practical deployments we do not expect interference to play a major issue since ISLA will be deployed in the proprietary frequency bands licensed by cellular companies, which in turn will have limited interference.

The X310 SDR has two identical RF chains, and can sample the full 100 MHz bandwidth with UBX160 daughterboards. To emulate the MEMS spike-train in digital, we first measure the spike-train filter fre-



(a) Circuit Diagram (b) Prototype Circuit

Figure 4.7: ISLA Prototype Circuit

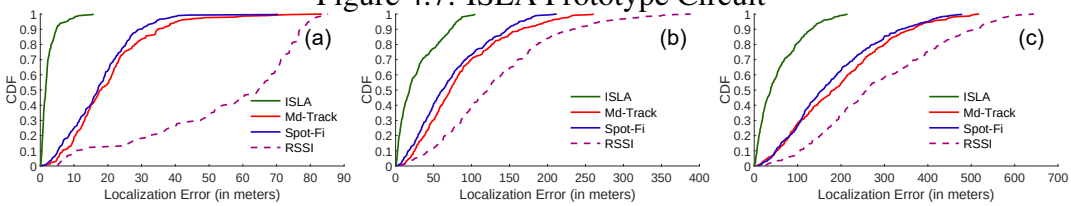


Figure 4.8: ISLA's localization accuracy compared against baselines across different testbeds: (a) Campus (b) Parking lot (c) Farm.

quency response once using a vector network analyzer (VNA), and we apply this filter frequency response to the received signals sampled at 100 MHz. Then, we downsample the filtered signal by simply keeping every 16th sample. This is equivalent to filtering the RF signal in analog and sample it below the Nyquist sampling rate. We also used a bandpass filters between the antenna and SDRs to remove out-of-band interferences and synchronized the two RF chains in time and phase through the GNU Radio Python API. In section 4.10.3, we present mircobenchmarks demonstrating the equivalence between applying the filter in digital and the above hardware prototype.

Testbed: Additionally, we also built 5G base station TX prototypes to transmit ambient 5G communication signals. As shown in Fig 4.6(d), the base station prototype consists an X310 USRP SDR with a UBX160 daughterboard, a 9 dBi Yagi directional antenna, and an RF Bay MPA-22-30 30 dB power amplifier. The base stations transmit 100 MHz OFDM packets. Using five base station prototypes, we created three testbeds with different dimensions and at different locations to conduct our experiments. Figure 4.6 shows the satellite images of our testbeds with the base stations and clients locations marked. The first testbed is 85 m long and 52 m wide on a university campus, surrounded by buildings on all sides. We designated 11 basestation locations in this testbed and chose five of them for each experiment. The second testbed is a 400 m by 240 m parking lot with 27 base station locations. The third testbed is at a 102 acre farmland with 860 m length 480 m width. We selected five out of the 17 potential locations to place the base stations in each experiment. For ground truth locations, we used differential GPS RTK with real-time RTCM correction data, which provides centimeter-level positioning accuracy.

4.10 Experimental Evaluation

4.10.1 Baselines

(1) *Spot-Fi:* [192] proposes a 2D MUSIC algorithm with spatial smoothing, which can localize clients by separating the multipath components jointly along the ToF and AoA domains.
(2) *mD-Track:* [194] separates propagation paths by leveraging multiple dimensions of the wireless signal (ToF, AoA, AoD and Doppler), and proposes an iterative algorithm that goes through multiple rounds of error computation and path re-estimation. In our experimental setup, leveraging the AoD and Doppler dimensions provides little benefit since the base station is equipped with a single antenna and the IoT device does not have high mobility relative to the base station.
Note that, systems like Spot-Fi and mD-Track were not designed for ambient localization, and thus need to be adapted here. Specifically, we leverage the ToF estimates provided by these baselines for the LoS path, and, in turn, self-localize the client by computing the relative ToF, as described in Section 4.7.
(3) *RSSI:* Past work leverages RSSI measurements to localize clients in outdoor cellular networks, by either using approximate path loss models for trilateration, or by using the known locations of nearby cells as coarse estimates. We implemented one recent RSSI baseline [180].
(4) *Spike-train filter-adapted baselines:* To provide a fair comparison against ISLA, we modify Spot-Fi and mD-Track to leverage the spike-train filter and utilize the wideband channel measurements for localization. It is non-trivial to adapt Spot-Fi for the spike-train filter since the spatial smoothing technique used in Spot-Fi requires uniformly spaced channel measurements across frequency, whereas the spike-train filter samples the OFDM frequency bins non-uniformly. To address this, we restructure the spatial smoothing subarray from [192] that allows Spot-Fi to be applied across the non-uniform frequencies sampled by the spike-train filter.

4.10.2 Results

Unless otherwise specified, for all results, we utilize 5 randomly chosen base stations as the anchor points.

A. Localization Accuracy Comparison against Baselines: We compare ISLA's localization against the baselines in Fig. 4.8. Note that, while ISLA is designed specifically to leverage the wideband channel sensed by the MEMS filter, the baselines are implemented without modification and thus utilize only the narrowband channel for localization.

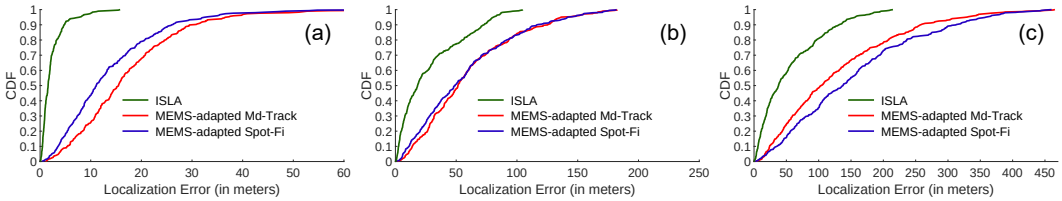
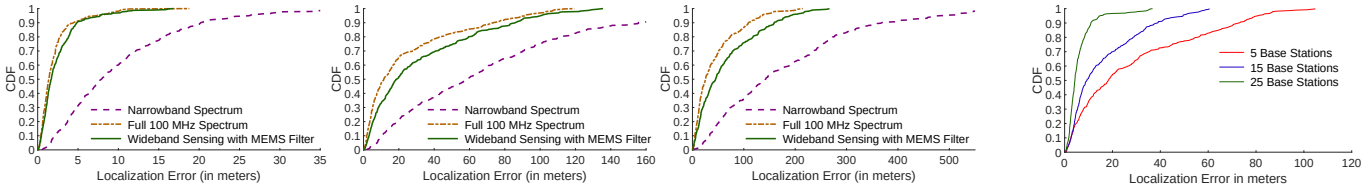


Figure 4.9: ISLA's localization accuracy compared against MEMS filter adapted baselines at: (a) Campus (b) Parking lot (c) Farm.



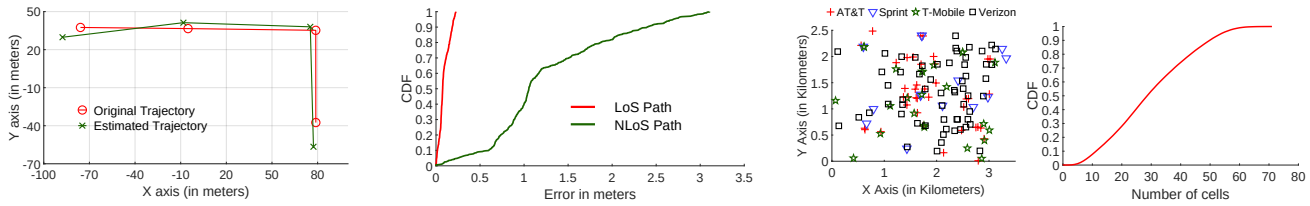
(a) Campus (b) Parking Lot (c) Agricultural Farm (d) ISLA's localization error vs #BSs

Figure 4.10: (a-c) Comparison of ISLA's localization accuracy when leveraging different amounts of spectrum across all three testbeds. (d) ISLA's localization error with different number of visible base stations.

From Fig. 4.8, ISLA achieves a median localization accuracy of 1.58 meters in the campus testbed, 17.6 meters in the parking lot testbed, and 37.8 meters in the farm testbed. Across the same three testbeds, Spot-Fi achieves median accuracies of 17.05 meters, 61.2 meters and 156.6 meters, whereas mD-Track achieves 18.11 meters, 71.8 meters, and 183.1 meters respectively. Thus, ISLA improves the localization accuracy over Spot-Fi and mD-track by $\sim 11\times$ in the campus testbed, and by $\sim 4\times$ in the parking lot and farm. ISLA is able to achieve such high gains since it leverages the spike-train filter to sense wideband channel on the narrowband device, which allows for much higher resolution compared to the baselines operating solely in the narrowband. Further, the localization improvement over the narrowband baselines is most significant in the campus testbed, since it has the most multipath from surrounding buildings, and thus ToF resolution is critical to separate out the LoS path from reflections.

Lastly, the RSSI baseline achieves median accuracies of 64.54 meters, 120.7 meters, and 260.8 meters respectively across the three testbeds. RSSI based methods generally have poor performance, as they tend to oversimplify path loss models that map RSSI values to distance, which does not hold for real world multipath channels.

B. Comparison against Spike-train-adapted Baselines: Next, we evaluate how leveraging the spike-train filter would benefit the performance of our narrowband baselines. Fig. 4.9 shows the CDF of localization accuracy comparing ISLA against the modified baselines that utilize the wideband channel from the spike-train filter. The RSSI baseline is not included here since its localization performance does not depend on bandwidth. Compared to its narrowband implementation, Spot-Fi's median accuracy



(a) Tracking Object Trajectory (b) Fab. MEMS filter vs ISLA's Imp. (c) 4G BS Density (d) # of Visible BSs

Figure 4.11: (a) Using ISLA to track object trajectory. (b) ToF difference between ISLA's prototype with fabricated MEMS filter and digitally implemented MEMS filter. (c) Deployment of 4G base stations in the downtown area of a major US city. (d) Number of visible 4G base stations at various downtown locations.

improves to 11.08 meters in the Campus testbed, 49.07 meters in the Parking Lot, and 137.76 meters in the farm. Similarly, mD-Track's median performance improves to 15.48 meters, 51.45 meters and 103.78 meters in the three testbeds respectively. Thus, Spot-Fi and mD-Track see improvements in localization accuracy by up to 54% and 76% respectively. This shows that other localization techniques can also benefit from the wide-band channel sensing capabilities enabled by the spike-train filter.

Additionally, Fig. 4.9 shows that given the same channel information, ISLA's off-grid CIR estimation algorithm is able to better resolve and estimate the relative ToF compared to Spot-Fi and mD-Track. This is because these baselines were designed to leverage multiple information dimensions to separate out the multipath components, with both baselines leveraging 3 or more antennas for separation in the AoA domain, and mD-Track further using the additional dimensions of Doppler and AoD as well. In contrast, here the IoT device has to separate out multipath in the ToF domain alone, and ISLA is able to achieve very accurate localization owing to its off-grid estimation algorithm.

C. ISLA Leveraging Different Amounts of Spectrum: In this experiment, we compare ISLA's localization algorithm applied across three different amounts of spectrum utilization — (1) ISLA applied only to the wideband sparse channel sensed by the spike-train filter (without combining with narrowband channel), (2) ISLA applied only to the narrowband channel of IoT device, and (3) ISLA applied across the entire 100 MHz bandwidth of the received 5G signal. Fig. 4.10 plots the CDF of localization accuracy achieved across the three testbeds.

ISLA applied on the narrowband channel performs the poorest, achieving median accuracies of 7.9 meters, 58.9 meters and 142.52 meters in the campus, parking lot and farm testbeds. In contrast, ISLA along with the spike-train filter can achieve corresponding median accuracies of 1.68 meters, 18.8 meters and 45.04 meters. Thus, ISLA along with spike-train, achieves an improvement in localization accuracy of $3.16\times - 4.7\times$ compared to ISLA applied in the narrowband spectrum, despite both baselines capturing the same amount of channel measurements. The advantage of spike-train stems from the fact that it

enables the narrowband receiver to capture channel measurements that span a much larger bandwidth, which results in much higher ToF resolution.

On the other hand, ISLA's localization algorithm applied on the full 100 MHz spectrum achieves median accuracies of 1.38 meters, 11.44 meters and 25.8 meters respectively on the three testbeds. Thus, ISLA with the spike-train filter reduces the localization accuracy by only $1.21\times$, $1.64\times$, and $1.74\times$ respectively compared to this upper bound. This demonstrates that the spike-train filter can enable a narrowband device to achieve localization accuracy within a factor of $2\times$ compared to a broadband receiver, despite the fact that it subsamples the signal by $16\times$ below Nyquist.

D. Localization with Number of Anchor Base Stations:
In Fig. 4.10(d), we compare ISLA's localization performance with 5, 15 and 25 base stations used as anchor points respectively, in the parking lot testbed. With 5 base stations, ISLA achieves a median accuracy of 17.6 meters, which improves to 9.27 meters with 15 base stations, and 4.26 meters with 25 base stations. This improvement becomes even more significant at the tail, with ISLA achieving 90^{th} percentile accuracy of 73.16 meters with 5 base stations, which improves to 10.9 meters accuracy with 25 base stations at 90^{th} percentile. Thus, leveraging more base stations can significantly improve the localization accuracy achieved by ISLA.

E. Tracking Objects: We move the IoT device across an L-shaped trajectory (160 meters in length and 85 meters in width) in the parking lot testbed, and collect packet transmissions from the base stations at different points along this trajectory. In this experiment, we pick 7 fixed base stations to utilize as anchor points, and we show the ground truth trajectory and corresponding estimated trajectory by ISLA in Fig. 4.11(a). As can be observed, ISLA's high localization accuracy allows to faithfully capture the shape of the ground truth trajectory.

4.10.3 Microbenchmarks

A. CIR Estimation using Fabricated MEMS Spike-train Filter: To verify the equivalence between our outdoor implementation and using the prototype with the fabricated MEMS spike-train filter at 400 MHz, we conduct indoor experiments at 400 MHz. Specifically, we evaluate the error in reconstructed CIR and estimated ToF values between the prototype with the fabricated filter and ISLA with the digital filter implementation. In Fig. 4.11(b), we show the CDF of the errors in ToF values (converted to distance (meters)) recovered by the two approaches, for both LoS and NLoS paths. We can see that the position of the LoS path in the CIR estimated from both approaches are very close, with the median error between

Direction	NW	NE	SE	SW
Median	1.3535 m	1.3544 m	1.3267 m	1.3681 m
Std Dev	0.4948 m	0.6026 m	0.4908 m	0.512 m

Table 4.1: Invariance of Localization Error to Orientation

their estimates being 0.075 meters. The error in the NLoS paths is higher, with a median error of 1.05 meters. However, this will not affect the localization performance between the two since localization only uses the LoS path. This microbenchmark demonstrates that ISLA's approach of applying the filter and subsampling in digital is equivalent to using the fabricated filter from a localization perspective, and that the results shown here are representative of a fully implemented system.

B. Density of Deployed Base Stations: In Section 4.10.2D, we have shown that ISLA's localization accuracy increases substantially as we use more anchor base stations. Here, we study the distribution of how many base stations can the client overhear at a given location. Using publicly available databases [218], we retrieved the locations of 4G LTE base stations belonging to 4 major carriers in the United States. We chose 4G LTE for this analysis since 5G deployment is still in its nascent stage in the USA, but we expect the target coverage for 5G networks to exceed the 4G deployment.

In Fig. 4.11(c), we show the scatter plot of the 4G base stations located in the downtown area of a major metropolitan city in the USA. Using the cell coverage information provided in [218] for the different base stations, in Fig. 4.11(d), we plot the CDF of the number of base stations that the client can overhear at different locations on the map. We can see that at the 10^{th} percentile, the number of visible base stations is 11, thus implying that less than 10% of client locations see less than 11 base stations. Further, the median number of base stations visible to the client is 29. This demonstrates that the cellular deployment is dense enough to allow many anchor points, which in turn can achieves high localization accuracy.

C. Invariance to Orientation: Here, we demonstrate that the localization performance is independent of the orientation of the IoT device. This is because the arcs that define the locus of the IoT node, depend only on the angle subtended by the base stations at the IoT device's location, which is invariant to device rotation. At a given location in our campus testbed, we orient the IoT device along 4 different directions and perform 100 localization experiments at each orientation. From Table 4.1, we can see that the median and standard deviation in localization error is almost the same across the 4 orientations, thus demonstrating invariance to orientation.

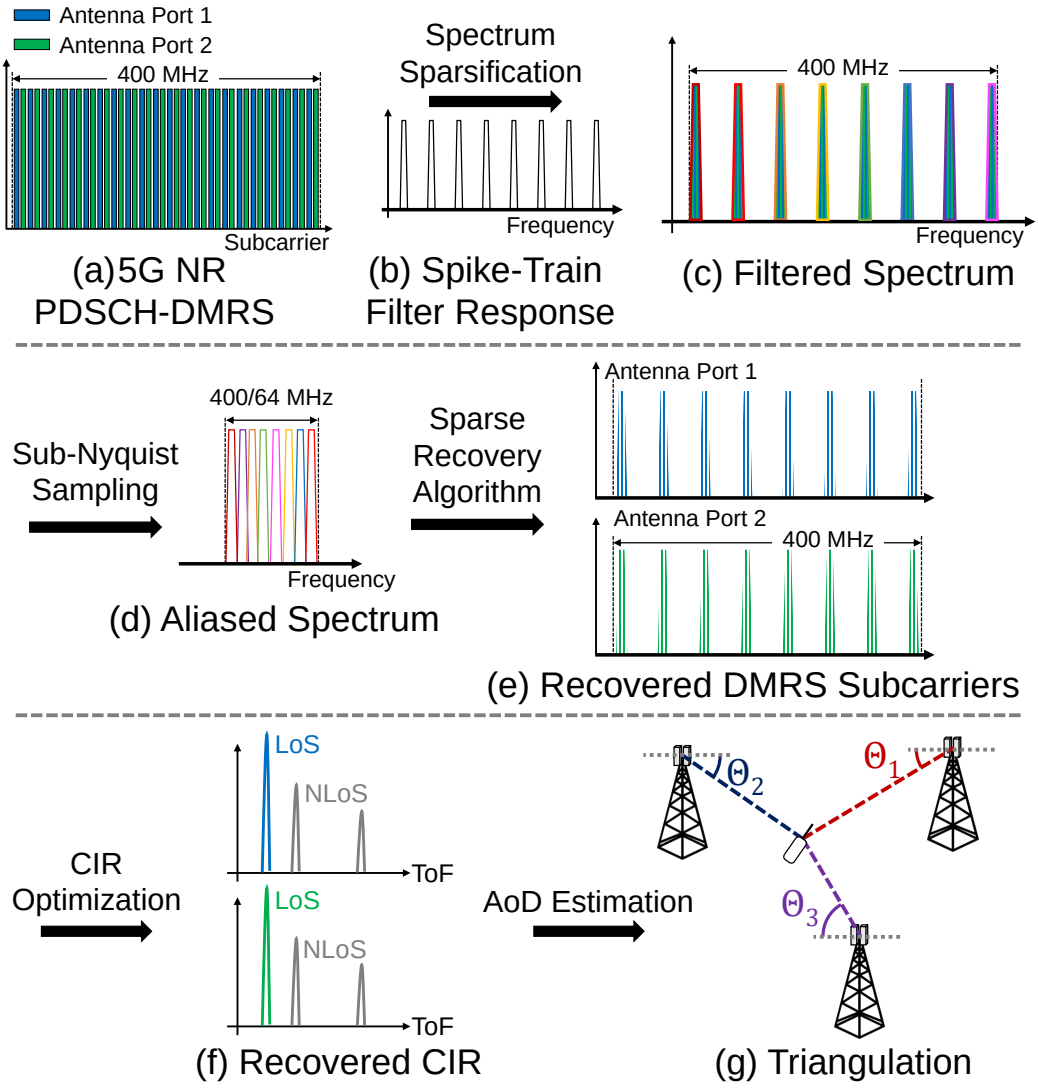


Figure 4.12: *mm-ISLA* **pipeline.** (a) Wideband 5G PDSCH-DMRS Spectrum Allocated to 2 Antenna Ports. (b) MEMS Spike-Train Filter Frequency Response. (c) Filtered Sparse Spectrum. (d) Sub-Nyquist Sampled Spectrum Aliased to the Narrow ADC Bandwidth. (e) Recovered DMRS Subcarriers. (f) Recover Channel Impulse Response. (g) AoD-Based Triangulation Localization.

4.11 Extending *ISLA* to mmWave

4.11.1 Motivation and Challenges

Our original implementation of *ISLA* is limited to sub-6GHz bands. However, leveraging the mmWave bands 5G signals for localizing IoT nodes is even more appealing, because of two characteristics of 5G

mmWave networks: 1) The small cell sizes lead to very dense deployments of base stations, up to 40 to 50 BS per square km [185], resulting in more potential anchor points for accurate localization. 2) The unprecedentedly wide signal bandwidth, up to 400 MHz in mmWave eMBB channels, provides high-resolution Time of Flight (ToF) estimation and, hence, high localization accuracy.

However, adapting *ISLA*'s coordination-free localization protocol to mmWave bands would be impractical, because *ISLA* avoids coordination with the gNBs by measuring the Time Difference of Arrival (TDoA) between two antennas on the IoT node. Such IoT design requires two antenna front-ends with tightly synchronized RX chains, which is infeasible in mmWave frequencies because of the expensive and power-consuming mmWave front-ends. Therefore, *mm-ISLA* abandons the dual front-end IoT design and the TDoA-based localization algorithm of *ISLA*. Instead, *mm-ISLA* overcomes the coordination-free challenge by leveraging the additional degree of freedom provided by the MIMO antenna arrays at the 5G gNBs. *mm-ISLA* first resolves channels from multiple TX antennas at the gNBs leveraging a unique 5G-NR waveform – DeModulation Reference Signal (DMRS) in the Physical Downlink Shared Channel (PDSCH). The unique resource allocation pattern in the DMRS waveforms allows *mm-ISLA* to distinguish the OFDM subcarriers allocated to each antenna in the gNB MIMO antenna array. Therefore, *mm-ISLA* can then leverage the channel differences across the antennas to estimate the Angle of Departure (AoD) of the Line-of-Sight (LoS) path from the gNB to the IoT node. Finally, with the AoD measurements of three gNBs, an *mm-ISLA* node can localize itself using the standard triangulation localization algorithm.

4.11.2 LoS AoD Estimation with PDSCH-DMRS Waveform

Figure 4.12 illustrates *mm-ISLA*'s system pipeline. *mm-ISLA* adopts the same super-resolution CIR reconstruction method as *mm-ISLA* by formulating an inverse optimization problem. Towards solving the coordination-free challenge, *mm-ISLA* however, takes a completely different approach than *ISLA*. The TDoA-based localization algorithm of *ISLA* is abandoned, because it requires two tightly synchronized RF front-ends, RF chains, and ADCs. The additional RF circuitry and ADC doubles the cost and power-consumption of the IoT nodes, which is even more infeasible in the mmWave frequencies than in the sub-6GHz bands. Restricted to a single antenna front-end, *mm-ISLA* enabled IoT nodes still manage to localize themselves without any coordination with the gNBs. To do so, *mm-ISLA* leverages another unique opportunity in 5G networks – the spatial diversity of the MIMO antenna arrays at the 5G gNBs. *mm-ISLA* tries to measure the ToF differences across antennas in the gNB MIMO antenna array, from which *mm-ISLA* can infer the AoD of the LoS path from the gNB to the IoT node. With AoD estimates

of three or more gNBs along with the gNB locations and antenna array orientations, *mm-ISLA* enabled IoT nodes will be able to apply the standard triangulation algorithm to localize themselves. However, to do so, *mm-ISLA* has to first be able to estimate the CIR from each gNB MIMO antenna separately.

The question becomes how can *mm-ISLA* isolate concurrent transmissions from TX MIMO antennas at the 5G gNB and estimate them corresponding CIR separately? Note that signals from different TX antennas have to be transmitted at the same time; otherwise, the transmitting time offset will corrupt the AoD estimation. To overcome this challenge, *mm-ISLA* leverages another unique opportunities in the 5G-NR standards, that is the resource allocation pattern in the 5G-NR PDSCH-DMRS waveforms. PDSCH-DMRS is a specific type of 5G-NR waveform used for decoding the PDSCH data, so it's a preamble-like waveform one can leverage to estimate the channel. When MIMO is enabled at the gNB, to decode the channels from the MIMO antennas, different antenna ports are allocated with a different set of interleaved subcarriers in the resource block [219], as shown in Fig. 4.12(a). Therefore, we can identify the DMRS subcarriers corresponding to each TX antenna and estimate their channels separately. Since the interleaved subcarrier allocation pattern ensures that the DMRS waveform from all TX antennas covers the entire bandwidth of the resource block, we can still achieve wideband CFR estimations for all TX antennas. Therefore, we can estimate the super-resolution CIRs corresponding to each TX antenna with a small modification to the inversion optimization problem to incorporate the subcarrier allocation in the PDSCH-DMRS waveform. Finally, we compare the ToF differences across the TX antennas to estimate the AoD of the LoS path.

4.12 Limitations and Discussion

- *Power Footprint:* To enable ambient localization, ISLA leverages a second antenna and RF chain, which increases the power footprint of the IoT device. However, we would like to note that the power overhead of an additional RF chain is going to be lower than that of a GPS module, which is the likely alternative for localization. This is because the additional RF chain on the IoT device is going to operate in the narrowband with very low sampling rates, whereas GPS incurs high operational power since it needs to receive and correlate long sequences to get the signal power above the noise floor for GPS lock acquisition. Hence, while ISLA's design does lead to an increased power footprint, it is still a better alternative compared to GPS.
- *Loss of SNR:* Since the MEMS spike-train filter is a passive device, the signal suffers from insertion loss when passed through the filter, thus resulting in loss of SNR. This is further exacerbated by the fact

that in practice, the out-of-band rejection of the spike train filter is finite, which results in further loss of SNR. It is possible to reduce the impact of this SNR loss at the circuit level by improving impedance matching and the isolation between input and output ports. We can also compensate for the SNR loss by averaging the channel measurements across multiple OFDM symbols.

- *Line-of-sight*: Similar to many localization systems, ISLA assumes the availability of line-of-sight (LoS) paths to the base stations which might not hold under occlusion. This, however, can be addressed by potentially selecting a subset of base stations with LoS paths using similar techniques demonstrated in [192]. With the dense deployment of 5G base stations, we expect a significant subset of base stations to have LoS path to the node.
- *Fast Mobility*: The current design of ISLA is not suitable for highly dynamic applications with fast mobility such as tracking cars. This is because the localization algorithm must receive wideband 5G packets from 4 or more base stations before it can self-localize.
- *Multiple Providers*: ISLA can benefit from capturing signals from multiple different providers since the IoT node does not need to associate with the base stations. However, different providers operate in different frequency bands which would require different spike-train filters. This could potentially be addressed by having multiple filters and switching between them similar to our design in sec. 4.9.

Chapter 5

CONCLUSION

In this dissertation, I presented end-to-end systems and techniques that allow mmWave networks to scale and enable higher user density and data rates in communication systems, as well as higher sensing resolution and the ability to scale to ubiquitously deployed heterogenous devices in sensing and localization systems. A common theme in this dissertation is developing new algorithms and techniques that expand and enhance the capabilities of mmWave networks, by leveraging intelligence and learning techniques across the entire computing and network stack. Our work aims to pave the way for next-generation mmWave wireless networks by developing hardware-software systems that pushes the boundaries of technology and applications in terms of scale and function. Specifically, we designed and built the following systems as part of this dissertation:

- **Many-to-Many Beam Alignment for Dense Spatial Reuse in mmWave WLANs :** We introduced BounceNet, the first many-to-many millimeter wave beam alignment system that can efficiently align the beams of many APs and clients in a manner that allows them to simultaneously communicate without interfering. We demonstrate the opportunity of routing physical signals along different paths that bounce off the environment to improve the spatial reuse of the network. We harness this opportunity to design new algorithms that maximize network throughput while maintaining a lower bound of fairness for each client. We evaluated BounceNet using three experimental testbeds and demonstrated that it can enable dense spatial reuse and scale the total network throughput with the number of APs and clients.

- **Scaling mmWave Wireless Network-on-Chip using Deep Reinforcement Learning:** We present NeuMAC, the first MAC protocol that can learn and adapt to the highly dynamic traffic patterns at very fine granularity in a mmWave wireless NoC processor. We design a lightweight deep reinforcement learning framework that introduces little overhead to the multi-core processor and can operate within the tight timing, power and area constraints of chip multicore processors. The protocol also accounts for non-trivial dependencies between packet delivery and computation speedups by optimizing for end-

to-end execution time. We extensively evaluate our design and demonstrate significant improvement in network performance and reduction in the overall execution time on the multicore processor.

- **Enabling High Resolution Self-Localization for Massive IoT Deployments using Ambient 5G Signals:** We present ISLA, the first system that allows IoT nodes to localize themselves at scale by only using ambient 5G signals without any coordination with the base stations. We demonstrate the ability to reduce the sampling rate by $16\times$ while retaining the benefits of high bandwidth 5G signals by leveraging recent advances in MEMS RF filters. We implement and evaluate ISLA to demonstrate accurate localization in 3 outdoor settings.

5.1 Future Directions

There are multiple research directions that I want to explore going ahead:

- **Wireless for Autonomous Driving and V2X connectivity:** Unlike cameras and LiDARs which fail in low visibility conditions, radar signals can penetrate through fog, snow and dust, and are, therefore, more favorable for such scenarios. Additionally, past work also demonstrates that radar sensors can be used for much more than simple unidirectional ranging, and can in fact be used to generate high resolution and perceptual radar images of cars even through fog and rain. As a result of these advances, we are witnessing a rapid proliferation of radar sensors in modern cars for sensing and autonomous driving functionalities.

  However, with this rapid deployment where every car on the street is soon going to be equipped with a radar sensor, we will face severe interference between the radar sensors on different cars since radar is an active sensor (unlike cameras) that transmits FMCW signals into the environment and processes the reflected signals for perception. Hence, a very important future direction of research is designing and building interference avoidance and mitigation schemes for ubiquitous radar sensors that will allow wireless perception for self driving cars to truly scale.

  Additionally, in order to achieve the vision of *fully autonomous traffic networks*, I want to conduct research of building and designing reliable and high throughput V2I and V2V networking infrastructure at scale. There is a need for a new networking paradigm where cars can share information with each other and with traffic infrastructure to cover blind spots and receive advance notification for safety critical information. My research will explore the design trade-offs and networking protocols in such

V2X networks that can provide the required reliability and latency guarantees for the self-driving application.

- **Scaling IoT Deployments by Expanding into mmWave regime:** IoT deployments are becoming an increasingly critical component of the workflow in a number of diverse industries such as manufacturing, agriculture, retail and transportation. However, today's IoT devices operate in narrow and highly crowded portions of the spectrum, offering extremely low communication data rates (Sigfox offers up to 600 bits per second and LoRaWAN offers up to 27 kilo bits per second). As the deployment scale of these devices runs into the billions, we will face a huge strain on network performance and reliability. I want to explore the possibility of low power millimeter-wave (mmWave) based IoT radios, since the huge bandwidth at mmWave frequencies would allow us to accommodate billions of IoT devices. However, the current state of mmWave technology prohibits us from realizing this goal since mmWave devices are expensive and power hungry. I want to work on minimizing the hardware complexity by eliminating the high-power and expensive radio front-end and RF circuitry, in lieu of cheaper and power efficient components like non-uniformly spaced phased arrays, low resolution ADCs (Analog-to-Digital Converters), and 2-bit phase shifters. I believe that the performance loss resulting from these non-ideal hardware components can be compensated for by leveraging intelligent algorithmic solutions at the upper layers of the network stack.

- **AI for Next-G Wireless Networks:** Next-G wireless networks (5G, 6G and beyond) are positioned to enable unprecedented communication and sensing capabilities for a diverse set of devices, ranging from resource constrained IoT devices to power and throughput hungry smartphones and virtual reality headsets. To achieve this goal, 5G and future cellular networks will leverage many new hardware and software capabilities such as massive antenna arrays, multiple frequency bands and flexible bandwidth and channel allocation. While such a diverse feature set brings flexibility to service a wide variety of communication scenarios, it also significantly increases the complexity of the radio access network (RAN), offering a combinatorially large number of choices for the various control knobs of the wireless links (modulation order, coding rate, OFDM parameters, etc.). Manually configuring these networks for each of the different use cases is going to be challenging and sub-optimal. Towards this end, I want to explore the possibility of leveraging *AI for Self-Organizing Wireless Networks* that can learn from experience and automatically configure itself to achieve the optimal user experience for the specific task at hand.

  Given that cellular systems are becoming increasingly complex, I believe this is a natural step in the evolution of Next-G wireless networks. A data-driven approach that learns directly from experience

without requiring hand-tuned protocols will allow next-G cellular networks to scale seamlessly to the extremely diverse set of end user devices and application domains envisioned in networks of the future.

Appendix A

BOUNCENET – PROOF OF LEMMA 2.6.1

Suppose we are given a graph $G(V, E)$ where $|V| = N$ and $d(u)$ denotes the degree of u. Consider the following process which iteratively assigns weights (in the range $\{0 \dots M\}$) to the vertices. The initial assignment is F_0 such that $F_0(v) = M$ for all $v \in V$. We compute F_t as follows:

- Compute a Weighted Max Independent Set W_{t+1} in the weighted graph induced by G and F_t.
- If $u \in W_{t+1}$, then $F_{t+1}(u) = F_t(u) - (d(u) + 1)$ if $F_t(u) > 2(d(u) + 1)$ and $F_{t+1}(u) = 0$ otherwise.
- If $u \notin W_{t+1}$, then $F_{t+1}(u) = F_t(u)$.

Lemma A.1 *If* $t = O(M \log(NM))$, *then* $F_t(u) = 0 \; \forall u \in V$

Proof Consider the potential function $T_t = \sum_u F_t(u)$.

Claim A.1 $T_{t+1} \leq T_t(1 - 1/M)$.

Proof Consider the set of vertices S_t containing u's such that $F_t(u) > 0$. Since the maximum value of $F_t(u)$ is M, it follows that

$$|S_t| \geq T_t/M \tag{A.1}$$

Consider now the set W_{t+1}, and w.l.o.g. assume that $W_{t+1} \subset S_t$. Observe that W_{t+1} must be a *maximal* independent set, i.e., we cannot add any $u \in S_t - W_{t+1}$ to W_{t+1} without violating the independence property. Since the total number of nodes with an edge to a node in W_{t+1} (including self-loops) is at most $\sum_{w \in W_{t+1}} d(w) + 1$, it follows that

$$\sum_{w \in W_{t+1}} d(w) + 1 \geq |S_t| \tag{A.2}$$

However, the left-hand side in the above expression is upper bounded by the amount by which we reduce the potential, i.e., by the difference $T_t - T_{t+1}$ (the reduction in potential could be higher, because we round all weights smaller than $d + 1$ to 0). From Equations A.1 and A.2 we have

$$T_t - T_{t+1} \geq \sum_{w \in W_{t+1}} d(w) + 1 \geq |S_t| \geq T_t/M$$

and the lemma follows.

Since T_t has integral values, it follows that after $O(M \log(T_0))$ steps we have $T_t = 0$, and therefore $F_t(u) = 0$ for all u.

Appendix B

BOUNCENET – DATA RATE GAINS FOR 5 APS

In Fig. B.1, we present results for the case when there are 5 APs in the network. This allows us to evaluate BounceNet's performance in scenarios where the number of clients is greater than the number of APs. In such scenarios where the clients outnumber the APs, two or more clients could be assigned to the same AP, following the algorithm presented in Section 2.6.1. Since clients that share an AP can essentially be considered as interfering links, the corresponding nodes in the conflict graph will have edges between them. We can then apply BounceNet's signal routing algorithm (Section 2.6.2 and 2.6.3) to this modified conflict graph.

Fig. B.1(a) shows the total network data rate, and Fig. B.1(b) shows the average network data rate per client, as a function of the number of clients in the network. BounceNet is able to deliver a total of 21.33 Gbps, 20.81 Gbps and 15.78 Gbps data rates for 10 clients in the 3° beam, 12° beam and the phased array testbeds respectively. The baseline performs almost as well as BounceNet for the 3° beam since the interference in this case is very limited, and, as a result, the baseline is able to exploit spatial reuse. However, as the amount of interference increases, the performance of the baseline deteriorates, with BounceNet achieving $2.2\times$ and $3.2\times$ gain in network throughput over the baseline for the case of 10 clients in the 12° beam, and the phased array testbeds respectively. Since the baseline does not account for interfering links, it leads to frequent packet collisions, and as a result, inefficient use of the channel.

Compared to 802.11ad, BounceNet achieves $3.26\times$, $3.35\times$, and $2.78\times$ gain in network throughput for the case of 10 clients in the 3° beam, 12° beam, and the phased array testbed respectively. One should note that for 802.11ad, the gains with 5 APs are smaller as compared to the gains observed in Section 2.9.C, where there were 10 APs in the network. This is because BounceNet's strength over 802.11ad comes primarily from its ability to exploit spatial reuse efficiently, and with only 5 APs in the network, the potential for spatial reuse is reduced, and therefore the gains that BounceNet can provide over the standard will be smaller. Hence, to achieve significant gains in throughput, BounceNet advocates for dense AP deployments with narrow directional antenna beams in mmWave networks.

Finally, the following points are worth noting.

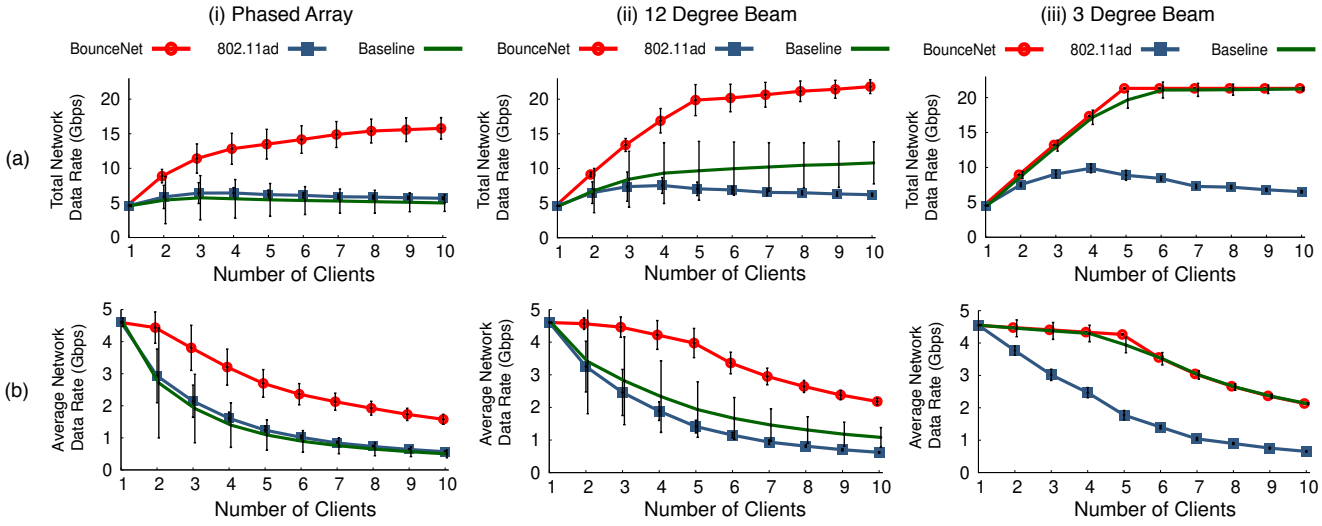


Figure B.1: Data rates in BounceNet, 802.11ad and baseline for the case of 5 APs in network (a) Total Network Data Rates (b) Average Client Data Rates.

- With the 3° beam in the 60 GHz testbed, we see that the total network data rate for BounceNet saturates after 5 clients as can be seen in Fig. B.1(a)(iii), achieving 21.33 Gbps and 21.29 Gbps for 10 clients and 5 clients respectively. This is expected, since at any given time at most 5 clients can be communicating simultaneously in the network. Such saturation can also be observed in the other two testbeds.
- It may seem counter-intuitive that the total network data rate for BounceNet in the 12° and the phased array testbeds continues to grow even when there are more than 5 clients in the network. This happens because as the number of clients increases in the network, the total number of propagation paths (direct and reflected) between APs and clients increases as well. Therefore, now it is more likely that BounceNet can find a set of five propagation paths that can coexist in the network, and consequently, BounceNet can schedule more clients in every time slot. However, one should note that the rate of growth of the network data rate reduces as the number of clients increases beyond five, and correspondingly, the average per-client data rates start to drop more sharply beyond five clients as can be seen in Fig. B.1(b).

Appendix C

NEUMAC – ENERGY AND LATENCY OVERHEAD CHARACTERIZATION

It is widely acknowledged that deep learning inference has high latency and energy overheads. However, since NeuMAC needs to optimize the performance of a multicore CPU, it needs to operate at very small time scales. As a result, it is imperative that NeuMAC's inference step be efficient in time and energy. In this appendix, we characterize the overheads of running inference on NeuMAC's Deep RL agent.

Towards this end, we design an illustrative hardware macro for NeuMAC's neural accelerator (shown in Fig. C.1). The trained quantized weights of NeuMAC's network are stored in the 32 KB on-chip SRAM. The primary compute elements in the macro are the (i) 128 element 8-bit multiplier, that can perform 128 parallel multiplications of 8-bit numbers, (ii) followed by a 7-layer carry save adder tree, which can add up to 128 8-bit numbers. Thus, the multiplier block and adder tree block together can implement either one 128 dimensional dot product, or two 64 dimensional dot products in a one iteration. The ReLU non-linear activation is implemented using comparators, which finally writes the result into an output buffer. It is important to note that this hardware macro is significantly simpler than a full scale neural network accelerator, such as [147].

Next, we elaborate on the pipeline for computing one inference step on NeuMAC's RL agent. Note that computing the value of one element in the first hidden layer of NeuMAC's neural network requires one 64 dimensional dot product[1]. Therefore, computing the values of all elements in the first hidden layer requires a total of 128 counts of 64 dimensional dot products. Similarly, computing the values at the second hidden layer requires 128 counts of 128 dimensional dot products, and computing the final layer requires 64 counts of 128 dimensional dot products. Hence, to compute one inference step in NeuMAC's deep network, we need to perform a total of 192 counts of 128-element dot products, and 128 counts of 64-element dot products. Further, since we can implement two 64-element dot products in parallel, one inference step requires an equivalent of 256 counts of 128 dimensional dot products to compute the output. Using this above macro design along with conservative and widely accepted hardware estimates, we next show that the design of NeuMAC's neural network architecture adds only marginal overheads,

[1] Although NeuMAC's input has 65 elements, for simplicity sake we perform calculations with 64 element input.

allowing it to operate under the resource constrained setting of a wireless NoC.

Latency Overhead: Here we estimate the latency of computing one inference step on NeuMAC's RL agent. The memory array is organized as 16 blocks of 64 by 256 memory elements, making a total of 32 KB storage. For 45nm technology, read access time from such memory sizes can be conservatively estimated to be around 2 ns [220]. Similarly, a 32-dimensional dot product can be computed within 2 ns [221]. Hence, we pipeline the data flow in three stages, first after the memory read, second after adding the outputs of 32 multipliers, and third at the output of the comparator bank. Hence, each stage has a maximum latency of 2 ns. As a result of such pipelining, one 128 element dot product is computed every 2 ns, that is, every 2 clock cycles[2]. As noted previously, one inference step requires 256 counts of 128 dimensional dot products. Hence, the total latency for one inference step is $256 \times 2 = 512\ ns$ (512 clock cycles). This inference latency of 512 cycles results in a small overhead of less than 6% per time step in our RL formulation. One point to note is that, the final deep network output is quantized to 8 bits. Hence, the sigmoid filter after the last layer can be implemented via a 256 element look-up table at a negligible latency overhead.

Energy Overhead: Next, we estimate energy consumption of the hardware macro. We use the energy values from the widely-cited paper [222], which approximately characterizes energy consumption of various compute elements and memory accesses. The dominant energy consumption steps are the reads from the memory array and the computations on the MAC (Multiply-ACcumulate) unit. From [222], 8 bit multiplies consume 0.2 pJ, and 8-bit additions consume 0.03 pJ. One 128 dimensional dot product on the MAC unit involves 128 multiplications and 127 additions. Thus the total energy comes to 29.41 pJ. Memory reads of 64 bits from 2 KB memory blocks requires 5 pJ. Thus, the 128 bit memory reads for each dot product requires 10 pJ. As a result, one 128 element dot product on the hardware accelerator requires 39.41 pJ, and with 256 counts, the energy consumed for a single inference step is 10088.96 pJ. Given that we require one inference every 10,000 ns, the neural accelerator consumes approximately only 1 mW of power on average. In comparison, a single transceiver on the multicore consumes 16 mW [81]. Lastly, note that the numbers in [222] are at 45 nm technology, so 1 mW is a conservative estimate.

Area Overhead: Lastly, the area overhead of the hardware macro is small. Since area is dominated by memory, the 32 KB of SRAM and few registers in the hardware accelerator impose a small overhead in comparison to the 512 KB of cache memory at each of the 64 cores. Thus we envision that such a hardware macro can reside on the same die and share the same clock as the multicore processor.

Thus, even a simple accelerator like the one demonstrated in Fig. C.1 can enable NeuMAC's agent to

[2]Our CPU clock is 1 GHz.

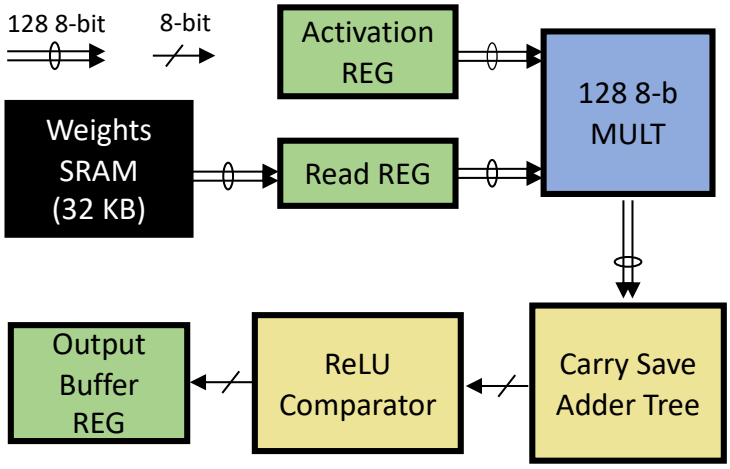


Figure C.1: Illustrative Block Diagram of hardware macro employed for overhead characterization of NeuMAC's deep network

operate under the resource constrained setting of a wireless NoC. Note that we do not employ any other advanced hardware optimization techniques and rely on reported hardware numbers that are widely accepted rather than the state-of-the-art today.

Appendix D

ISLA – PROOF OF LEMMAS 4.5.1 & 4.5.2

Here we re-state the lemmas and provide proofs.

Lemma 4.1 *For a sub-sampling factor P and N OFDM subcarriers, the complex valued scaling factors for each subcarrier will be preserved upon aliasing if $N = z \times P$, for some integer z, given the aliasing results in no collisions.*

Proof of lemma 4.1: Assume that $x[n]$ is a discrete signal from 0 to $N-1$, and we are sub-sampling (or *decimating*) it by a factor of P, meaning $y[n] = X[n \times P]$ for some integer P. Then the Discrete Fourier Transform of $y[n]$, denoted by $\hat{Y}[k]$ is

$$\begin{aligned}\hat{Y}[k] &= \sum_{n=0}^{N/P-1} x[nP]e^{-j2\frac{2\pi}{N/P}kn} \\ &= \frac{1}{P}\sum_{n=0}^{N-1} x[n]\sum_{m=0}^{P-1} e^{j\frac{2\pi}{P}mn}e^{-j2\frac{2\pi}{N/P}\frac{kn}{P}} \\ &= \frac{1}{P}\sum_{m=0}^{P-1}\big(\sum_{n=0}^{N-1} x[n]e^{-j(\frac{2\pi}{N}n)(k\frac{N/P}{N/P}-\frac{N}{P}m)}\big).\end{aligned}$$

Now if P divides N, in other words $N = Pz$ for some integer z, the above simplifies to

$$\begin{aligned}\hat{Y}[k] &= \frac{1}{P}\sum_{m=0}^{P-1}\big(\sum_{n=0}^{N-1} x[n]e^{-j(\frac{2\pi}{N}n)(k-zm)}\big) \\ &= \frac{1}{P}\sum_{m=0}^{P-1}\hat{X}[k-zm],\end{aligned}$$

where $\hat{X}$ is the DFT of $x[n]$. This proves that, as long as there is no collision, meaning that there is at most one index m in the above equation for which $\hat{X}[k - zm] \neq 0$, then the complex values of $\hat{X}[k]$ will be fully preserved upon sub-sampling. This proves the lemma.

We also point out that if P does not divide N, then the complex values are *not* preserved. Specifically, if N/P is not a proper integer, $\hat{Y}[k]$ will be in terms of $\hat{X}[k\frac{N/P}{N/P} - \frac{N}{P}m]$ where inside the argument, $k\frac{N/P}{N/P} - \frac{N}{P}m$, is not necessarily an integer. As a result, the original information of $\hat{X}[k]$ is never repeated in any of the $\hat{Y}$ indices. In fact, $\hat{Y}$ would closely relate to an interpolated version of $\hat{X}$ with the Dirichlet kernel.

Lemma 4.2 *Consider an OFDM symbol with N frequency subcarriers, indexed as $\{f_{\frac{-N}{2}}, \ldots, 0, \ldots, f_{\frac{N}{2}-1}\}$ with inter-frequency spacing of Δf, and a narrowband receiver that subsamples by $P\times$. If P^2 divides N, then the ideal filter parameters that meet all three requirements are: (1) $f_M^0 = f_{\frac{-N}{2}}$, (2) $\left(\frac{N}{P^2} - 1\right) \times \Delta f < \Delta S < \frac{N}{P^2} \times \Delta f$, and (3) $\Delta F = \frac{N}{P}(1 + \frac{1}{P}) \times \Delta f$.*

Proof of Lemma 4.2: First, we show that no two frequencies collide after aliasing. Let $q = \frac{N}{P}$, and assume that two frequencies f_α and f_β collide. Let f_α be k-th subcarrier (for $0 \leq k < P$) covered at the i-th passband ($0 \leq i < *\frac{\Delta S}{\Delta f}$), and let f_β have k' and i' as corresponding indices. To collide after aliasing, $f_\alpha - f_\beta = (k - k')\Delta F + (i - i')\Delta f$ must be an integer multiple of $q\Delta f$. However, $|k - k'| \leq P - 1$ and $|i - i'| < \frac{N}{P^2}$. Thus $\frac{|f_\alpha - f_\beta|}{\Delta f} < (\frac{P-1}{P} + \frac{1}{P})q = q$, meaning we must have $f_\alpha - f_\beta = 0$, proving the first design requirement. Second, we note that P passbands that do not overlap (since $\Delta S < \Delta F$), and each passband covers exactly $\frac{N}{P^2}$ subcarriers. We therefore have a total of $P \times \frac{N}{P^2} = q$ subcarriers that, as we just showed, do not overlap after aliasing. Therefore, after aliasing, each of the q subcarriers is covered exactly once, ensuring the second design requirement. Finally, we note that the smallest bin index is covered by the filter is $\min f_M = \frac{-N}{2}$, and the largest bin index is the last bin of the last passband, whose index can be computed as follows:

$$\begin{aligned}
\max f_M &= \frac{-N}{2} + (P-1) \times \Delta F + *\frac{\Delta S}{\Delta f} - 1 \\
&= \frac{-N}{2} + (P-1) \times \frac{N}{P}(1 + \frac{1}{P}) + (\frac{N}{P^2}) - 1 \\
&= -\frac{N}{2} + N - 1 = \frac{N}{2} - 1.
\end{aligned}$$

Thus, the entire bandwidth (including $f_{\frac{-N}{2}}$ and $f_{\frac{N}{2}-1}$) is covered, ensuring the last design requirement.

Appendix E

ISLA – MEMS SPIKE-TRAIN FILTER

Spike-Train Filter Implementation: Following Lemma 4.2, we can derive the desired frequency response of the spike-train filter, and design MEMS resonators topology accordingly. For example, in our experiment, we used a 100 MHz 5G-like OFDM waveform with N=2048 subcarriers and a subcarrier spacing $\Delta f = 49\ kHz$, and we down-sample the filtered waveform by a factor of P=16. According to Lemma 4.2, the desired filter should 16 spikes with a spike spacing of 6.64 MHz spanning the 100 MHz bandwidth, and each spike should have a width around 400 kHz.

We can design a spike-train filter leveraging the periodic resonance frequencies of a type of MEMS acoustic resonators that is commonly referred to as a LOBAR (Lateral Overtone Bulk Acoustic Resonator). As shown in Fig. E.1, the LOBAR resonator consists of 12 electrodes on the top of a thin film made of the piezoelectric material $LiNbO_3$. And we combine seven resonators in a ladder filter topology [223] to build a filter circuit. As a result, the LOBAR resonator architecture determines the spike frequencies, whereas the slight difference between different resonators determines the width of the spikes. For simplicity, here we only focus on these two key parameters of the spike-train filter response, since they are restricted by our channel recovery algorithm as described in Sec. 4.5. More details on the MEMS spike-train filter design can be found in [224].

(1) The width of the film: the spacing between spikes Δf is determined by the width of the thin film W as $\Delta f = v/W$, where v is the acoustic velocity in the piezoelectric material, which is $\sim 4\ km/s$ in our design. Therefore, to achieve the 6.6 MHz spike spacing, we design the film width W to be $\sim 660\ \mu m$.

(2) The film width difference between different shunt and series resonators: the spike width ΔF of the spike-train filter equals to the resonant frequency difference between shunt and series resonators in the ladder filter, which is determined by the difference ΔW between shunt and series resonators: $\Delta F = fc\frac{\Delta W}{W}$. We design with piezoelectric film width to be $660\ \mu m$ for series resonators and $660.26\ \mu m$ for shunt resonators, which leads to $\Delta W = 0.26\mu m$, so that the widths of the spikes are around 400 kHz.

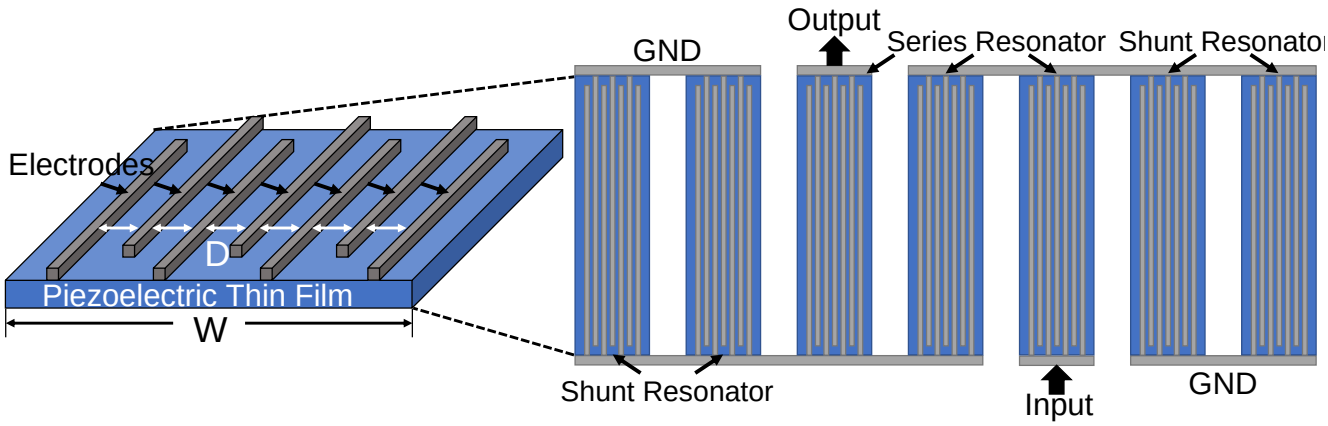


Figure E.1: MEMS Spike-Train Filter Architecture

Appendix F

ISLA – UPDATED OBJECTIVE FUNCTION TO ACCOUNT FOR RESIDUAL CFO

ISLA captures the narrowband channel and wideband channel from different subframes. Thus, there is going to be an additional phase accumulation between the two measurements due to residual CFO. To address this, we slightly modify Eq.4.6 where we split the objective function into two separate L-2 norm minimizations, with the first term containing only the wideband channel $h_M^{'}$, and the second term containing only the narrowband channel $h_{NB}^{'}$. This objective function is given below:

$$\begin{aligned}\{\tau_l^*\}_{l=1}^{L} = \underset{\tau_1,\ldots,\tau_L}{\arg\min}\Big(& \|h_M^{'} - V_M F_N \Psi (V_M F_N \Psi)^{\dagger} h_M^{'}\|^2 \\ & + \|h_{NB}^{'} - V_{NB} F_N \Psi (V_{NB} F_N \Psi)^{\dagger} h_{NB}^{'}\|^2 \Big)\end{aligned} \tag{F.1}$$

$$s.t. \quad \tau_l \geq 0 \quad \forall\, l \in \{1, 2, \ldots, L\}$$

The modified objective function is now invariant to phase offsets between the two channels, and ISLA can solve this updated optimization using the same technique described in Sec. 4.6.

BIBLIOGRAPHY

[1] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11 ad: directional 60 GHz communication for multi-gigabit-per-second Wi-Fi," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 132–141, 2014.

[2] Y. Ghasempour, C. R. Da Silva, C. Cordeiro, and E. W. Knightly, "Ieee 802.11 ay: Next-generation 60 ghz communication for 100 gb/s wi-fi," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 186–192, 2017.

[3] S. Jog, J. Wang, J. Guan, T. Moon, H. Hassanieh, and R. R. Choudhury, "Many-to-many beam alignment in millimeter wave networks," in *USENIX Conference on Networked Systems Design and Implementation, NSDI'19*, 2019.

[4] S. Jog, Z. Liu, A. Franques, V. Fernando, S. Abadal Cavallé, J. Torrellas, and H. Hassanieh, "One protocol to rule them all: Wireless network-on-chip using deep reinforcement learning," in *Proceedings of the 18th USENIX Symposium on Networked System Design and Implementation: April 12-14, 2021*. USENIX Association, 2021, pp. 973–989.

[5] S. Jog, J. Guan, S. Madani, R. Lu, S. Gong, D. Vasisht, and H. Hassanieh, "Enabling {IoT}{Self-Localization} using ambient 5g signals," in *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022, pp. 1011–1026.

[6] J. Guan, S. Jog, S. Madani, R. Lu, S. Gong, D. Vasisht, and H. Hassanieh, "Enabling iot self-localization using ambient 5g mmwave signals," in *Proceedings of the SIGCOMM'22 Poster and Demo Sessions*, 2022, pp. 49–51.

[7] Markets and Markets, "Millimeter Wave Technology Market worth 4,632.8 Million USD by 2022," Press Release, 2017.

[8] K. Hill, "A look at Verizon's fixed millimeter wave testing," RCR Wireless News, May 2017, http://www.rcrwireless.com/20170501/test-and-measurement/verizon-fixed-millimeter-wave-testing-tag6.

[9] TP-Link, "Talon AD7200 Multi-Band Wi-Fi Router."

[10] Acer, "TravelMate P6 TMP658-M-70S3 Laptop."

[11] NetGear, "NIGHTHAWK X10 R9000 Wi-Fi Router."

[12] Intel Inc., "Intel accelerates path to 5G," Press Release, 2016.

[13] W. Roh, J.-Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 106–113, 2014.

[14] M. Branda, "Qualcomm Research demonstrates robust mmWave design for 5G," Qualcomm Technologies Inc., November 2015.

[15] N. Choubey and A. Y. Panah, "Introducing Facebook's new terrestrial connectivity systems-Terragraph and Project ARIES," Facebook Research, 2016.

[16] Huawei, "Huawei to bring 73GHz mmWave Mu-MIMO live demo to Deutsche Telekom," Press Release, 2016.

[17] A. Connor-Simons, "Enabling wireless virtual reality," MIT News, 2016.

[18] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Enabling high-quality untethered virtual reality," in *NSDI*, 2017.

[19] S. Jog, J. Wang, H. Hassanieh, and R. R. Choudhury, "Enabling Dense Spatial Reuse in mmWave Networks," in *ACM SIGCOMM*, 2018.

[20] V. Turk, "These Supermarket Warehouse Robots Have Their Own Mobile Network," Vice Motherboard, 2016.

[21] B. Manz, "5G Cellular Networks Are the Future of Robotics," Mouser Electronics, 2016.

[22] H. Hassanieh, O. Abari, M. Rodriguez, M. Abdelghany, D. Katabi, and P. Indyk, "Fast millimeter wave beam alignment," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM, 2018, pp. 432–445.

[23] M. E. Rasekh, Z. Marzi, Y. Zhu, U. Madhow, and H. Zheng, "Noncoherent mmwave path tracking," in *Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*. ACM, 2017, pp. 13–18.

[24] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, "60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling," in *SIGMETRICS*, 2015.

[25] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with eyes closed: mm-wave beam steering without in-band measurement," in *Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 2416–2424.

[26] A. Zhou, X. Zhang, and H. Ma, "Beam-forecast: Facilitating Mobile 60 GHz Networks via Model-driven Beam Steering," in *INFOCOM*, 2017.

[27] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *IEEE*, 2014.

[28] C. R. Anderson and T. S. Rappaport, "In-Building Wideband Partition Loss Measurements at 2.5 and 60 GHz," *IEEE TWC*, 2004.

[29] M. K. Haider, Y. Ghasempour, D. Koutsonikolas, and E. W. Knightly, "Listeer: mmwave beam acquisition and steering by tracking indicator leds on wireless aps," 2018.

[30] M. K. Haider, Y. Ghasempour, and E. W. Knightly, "Search light: Tracking device mobility using indoor luminaries to adapt 60 GHz beams," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2018, pp. 181–190.

[31] J. Palacios, G. Bielsa, P. Casari, and J. Widmer, "Communication-driven localization and mapping for millimeter wave networks," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2402–2410.

[32] A. Zhou, L. Wu, S. Xu, H. Ma, T. Wei, and X. Zhang, "Following the shadow: Agile 3-d beam-steering for 60 GHz wireless networks," *IEEE INFOCOM*, 2018.

[33] A. Zhou, X. Zhang, and H. Ma, "Beam-forecast: Facilitating mobile 60 GHz networks via model-driven beam steering," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[34] T. Wei, A. Zhou, and X. Zhang, "Facilitating Robust 60 GHz Network Deployment By Sensing Ambient Reflectors," in *NSDI*, 2017.

[35] S. Kwon and J. Widmer, "Relay selection for mmwave communications," in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*. IEEE, 2017, pp. 1–6.

[36] S. Kwon and J. Widmer, "Multi-beam power allocation for mmwave communications under random blockage," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, 2018, pp. 1–5.

[37] S. Singh, F. Ziliotto, U. Madhow, E. Belding, and M. Rodwell, "Blockage and directivity in 60 GHz wireless personal area networks: From cross-layer model to multihop MAC design," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 8, pp. 1400–1413, 2009.

[38] S. Sur, X. Zhang, P. Ramanathan, and R. Chandra, "BeamSpy: Enabling Robust 60 GHz Links Under Blockage," in *NSDI*, 2016.

[39] D. Halperin, S. Kandula, J. Padhye, P. Bahl, and D. Wetherall, "Augmenting Data Center Networks with Multi-Gigabit Wireless Links," in *ACM SIGCOMM*, 2011.

[40] X. Zhou, Z. Zhang, Y. Zhu, Y. Li, S. Kumar, A. Vahdat, B. Y. Zhao, and H. Zheng, "Mirror Mirror on the Ceiling: Flexible Wireless Links for Data Centers," in *ACM SIGCOMM*, 2012.

[41] Y. Cui, S. Xiao, X. Wang, Z. Yang, C. Zhu, X. Li, L. Yang, and N. Ge, "Diamond: Nesting the Data Center Network with Wireless Rings in 3D Space," in *NSDI*, 2016.

[42] T. Wei and X. Zhang, "Pose Information Assisted 60 GHz Networks: Towards Seamless Coverage and Mobility Support," in *MobiCom'17*, 2017.

[43] D. Zhang, M. Garude, and P. H. Pathak, "mmChoir: Exploiting joint transmissions for reliable 60 GHz mmwave WLANs," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2018, pp. 251–260.

[44] H. S. Rahul, S. Kumar, and D. Katabi, "Jmb: scaling wireless capacity with user demands," in *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*. ACM, 2012, pp. 235–246.

[45] E. Hamed, H. Rahul, M. A. Abdelghany, and D. Katabi, "Real-time Distributed MIMO Systems," in *Proceedings of ACM SIGCOMM, 2016*, pp. 412–425.

[46] X. An, S. Zhang, and R. Hekmat, "Enhanced mac layer protocol for millimeter wave based WPAN," in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2008, pp. 1–5.

[47] K. Venugopal, M. C. Valenti, and R. W. Heath, "Interference in finite-sized highly dense millimeter wave networks," in *Information Theory and Applications Workshop (ITA), 2015*. IEEE, 2015, pp. 175–180.

[48] X. An and R. Hekmat, "Directional MAC protocol for millimeter wave based wireless personal area networks," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*. IEEE, 2008, pp. 1636–1640.

[49] M. X. Gong, R. Stacey, D. Akhmetov, and S. Mao, "A directional CSMA/CA protocol for mmWave wireless PANs," in *Wireless Communication and Networking Conference*. IEEE, 2010, pp. 1–6.

[50] M. X. Gong, D. Akhmetov, R. Want, and S. Mao, "Directional CSMA/CA protocol with spatial reuse for mmWave wireless networks," in *Global Telecommunications Conference (GLOBECOM)*. IEEE, 2010, pp. 1–5.

[51] J. Qiao, L. X. Cai, X. Shen, and J. W. Mark, "STDMA-based scheduling algorithm for concurrent transmissions in directional millimeter wave networks," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5221–5225.

[52] R. Mudumbai, S. Singh, and U. Madhow, "Medium access control for 60 GHz outdoor mesh networks with highly directional links," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 2871–2875.

[53] Y. Ghasempour, M. K. Haider, C. Cordeiro, D. Koutsonikolas, and E. W. Knightly, "Multi-stream beam-training for mmwave mimo networks," in *ACM MobiCom*, 2018.

[54] Y. Ghasempour, M. K. Haider, and E. W. Knightly, "Decoupling beam steering and user selection for MU-MIMO 60-GHz WLANs," *IEEE/ACM Transactions on Networking*, pp. 1–14, 2018.

[55] R. R. Choudhury, X. Yang, R. Ramanathan, and N. H. Vaidya, "On designing MAC protocols for wireless networks using directional antennas," *IEEE Transactions on Mobile Computing*, vol. 5, no. 5, pp. 477–491, 2006.

[56] R. R. Choudhury and N. H. Vaidya, "Deafness: A MAC problem in ad hoc networks when using directional antennas," in *Proceedings of the 12th IEEE International Conference on Network Protocols.*, 2004, pp. 283–292.

[57] G. Jakllari, W. Luo, and S. V. Krishnamurthy, "An integrated neighbor discovery and MAC protocol for ad hoc networks using directional antennas," *IEEE Transactions on Wireless Communications*, vol. 6, no. 3, pp. 1114–1024, 2007.

[58] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste, "DIRC: Increasing indoor wireless capacity using directional antennas," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 171–182, 2009.

[59] V. Shrivastava, N. Ahmed, S. Rayanchu, S. Banerjee, S. Keshav, K. Papagiannaki, and A. Mishra, "Centaur: realizing the full potential of centralized wlans through a hybrid data path," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 297–308.

[60] IEEE Standards Association, "IEEE Standards 802.11ad-2012: Enhancements for Very High Throughput in the 60 GHz Band," 2012.

[61] Y. Ghasempour, C. R. da Silva, C. Cordeiro, and E. W. Knightly, "IEEE 802.11ay: Next-generation 60 GHz communication for 100 Gb/s Wi-Fi," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 186–192, 2017.

[62] Qualcomm, "QCA9500 specifications," https://www.qualcomm.com/products/qca9500.

[63] Tensorcom, "Product specifications," http://tensorcom.com/products-1/.

[64] W. Wang, Y. Wang, X.-Y. Li, W.-Z. Song, and O. Frieder, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking*. ACM, 2006, pp. 262–273.

[65] S. Khanna and K. Kumaran, "On wireless spectrum estimation and generalized graph coloring," in *Proceedings of INFOCOM*, vol. 3. IEEE, 1998, pp. 1273–1283.

[66] K. N. Ramachandran, E. M. Belding-Royer, K. C. Almeroth, and M. M. Buddhikot, "Interference-Aware Channel Assignment in Multi-Radio Wireless Mesh Networks." in *Infocom*, vol. 6, 2006, pp. 1–12.

[67] A. Mishra, S. Banerjee, and W. Arbaugh, "Weighted coloring based channel assignment for WLANs," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 3, pp. 19–31, 2005.

[68] A. Frank, "Some polynomial algorithms for certain graphs and hypergraphs," in *Proceedings of the Fifth British Combinatorial Conference*, 1975, pp. 211–226.

[69] Pasternack Enterprises Inc., "60 GHz Transmitter/Receiver Development System," www.pasternack.com.

[70] T. Moon, J. Guan, and H. Hassanieh, "Online millimeter wave phased array calibration based on channel state information," in *IEEE VLSI Test Symposium, 2019. VTS'19.* IEEE, 2019.

[71] H. Assasa and J. Widmer, "Extending the ieee 802.11ad model: Scheduled access, spatial reuse, clustering, and relaying," in *Proceedings of the Workshop on Ns-3*, ser. WNS3 '17. New York, NY, USA: ACM, 2017. [Online]. Available: http://doi.acm.org/10.1145/3067665.3067667 pp. 39–46.

[72] Jack Clark, "Intel: Why a 1,000-core chip is feasible," Press Release, 2010.

[73] B. Bohnenstiehl, A. Stillmaker, J. Pimentel, T. Andreas, B. Liu, A. Tran, E. Adeagbo, and B. Baas, "A 5.8 pj/op 115 billion ops/sec, to 1.78 trillion ops/sec 32nm 1000-processor array," in *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*. IEEE, 2016, pp. 1–2.

[74] Y.-h. Chen, J. Emer, and V. Sze, "Eyeriss v2: A Flexible and High-Performance Accelerator for Emerging Deep Neural Networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 2, pp. 292–308, 2019.

[75] N. Abeyratne, R. Das, Q. Li, K. Sewell, B. Giridhar, R. G. Dreslinski, D. Blaauw, and T. Mudge, "Scaling towards kilo-core processors with asymmetric high-radix topologies," in *Proceedings of the HPCA-19*, 2013, pp. 496–507.

[76] S. Wang and T. Jin, "Wireless network-on-chip: A survey," *The Journal of Engineering*, vol. 2014, no. 3, pp. 98–104, 2014.

[77] A. Karkar, T. Mak, K.-F. Tong, and A. Yakovlev, "A survey of emerging interconnects for on-chip efficient multicast and broadcast in many-cores," *IEEE Circuits and Systems Magazine*, vol. 16, no. 1, pp. 58–72, 2016.

[78] S. Abadal, M. Nemirovsky, E. Alarcón, and A. Cabellos-Aparicio, "Networking challenges and prospective impact of broadcast-oriented wireless networks-on-chip," in *Proceedings of the 9th International Symposium on Networks-on-Chip*. ACM, 2015, p. 12.

[79] S. Abadal, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "Orthonoc: a broadcast-oriented dual-plane wireless network-on-chip architecture," *IEEE transactions on parallel and distributed systems*, vol. 29, no. 3, pp. 628–641, 2018.

[80] N. Mansoor and A. Ganguly, "Reconfigurable wireless network-on-chip with a dynamic medium access mechanism," in *Proceedings of the 9th International Symposium on Networks-on-Chip*. ACM, 2015, p. 13.

[81] V. Fernando, A. Franques, S. Abadal, S. Misailovic, and J. Torrellas, "Replica: A Wireless Many-core for Communication-Intensive and Approximate Data," in *ASPLOS*, 2019.

[82] S. Abadal, A. Cabellos-Aparicio, E. Alarcon, and J. Torrellas, "Wisync: An architecture for fast synchronization through on-chip wireless communication," *ACM SIGOPS Operating Systems Review*, vol. 50, no. 2, pp. 3–17, 2016.

[83] N. E. Jerger, L.-S. Peh, and M. Lipasti, "Virtual circuit tree multicasting: A case for on-chip hardware multicast support," in *2008 International Symposium on Computer Architecture*. IEEE, 2008, pp. 229–240.

[84] T. Krishna, L.-S. Peh, B. M. Beckmann, and S. K. Reinhardt, "Towards the ideal on-chip fabric for 1-to-many and many-to-1 communication," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2011, pp. 71–82.

[85] S. Abadal, B. Sheinman, O. Katz, O. Markish, D. Elad, Y. Fournier, D. Roca, M. Hanzich, G. Houzeaux, M. Nemirovsky et al., "Broadcast-enabled massive multicore architectures: A wireless rf approach," *IEEE micro*, vol. 35, no. 5, pp. 52–61, 2015.

[86] S. Deb, K. Chang, X. Yu, S. P. Sah, M. Cosic, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Design of an energy-efficient cmos-compatible noc architecture with millimeter-wave wireless interconnects," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2382–2396, 2012.

[87] N. Weissman and E. Socher, "9mw 6gbps bi-directional 85–90ghz transceiver in 65nm cmos," in *2014 9th European Microwave Integrated Circuit Conference*. IEEE, 2014, pp. 25–28.

[88] X. Yu, H. Rashtian, S. Mirabbasi, P. P. Pande, and D. Heo, "An 18.7-gb/s 60-ghz ook demodulator in 65-nm cmos for wireless network-on-chip," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 3, pp. 799–806, 2015.

[89] X. Yu, J. Baylon, P. Wettin, D. Heo, P. P. Pande, and S. Mirabbasi, "Architecture and design of multichannel millimeter-wave wireless noc," *IEEE Design & Test*, vol. 31, no. 6, pp. 19–28, 2014.

[90] S. Abadal, A. Mestres, R. Martínez, E. Alarcon, and A. Cabellos-Aparicio, "Multicast on-chip traffic analysis targeting manycore noc design," in *2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. IEEE, 2015, pp. 370–378.

[91] S. Deb, A. Ganguly, P. P. Pande, B. Belzer, and D. Heo, "Wireless noc as interconnection backbone for multicore chips: Promises and challenges," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 2, pp. 228–239, 2012.

[92] D. DiTomaso, A. Kodi, D. Matolak, S. Kaya, S. Laha, and W. Rayess, "A-winoc: Adaptive wireless network-on-chip architecture for chip multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3289–3302, 2014.

[93] S.-B. Lee, S.-W. Tam, I. Pefkianakis, S. Lu, M. F. Chang, C. Guo, G. Reinman, C. Peng, M. Naik, L. Zhang et al., "A scalable micro wireless interconnect structure for CMPs," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 217–228.

[94] D. W. Matolak, A. Kodi, S. Kaya, D. DiTomaso, S. Laha, and W. Rayess, "Wireless networks-on-chips: architecture, wireless channel, and devices," *IEEE Wireless Communications*, vol. 19, no. 5, pp. 58–65, 2012.

[95] V. Vijayakumaran, M. P. Yuvaraj, N. Mansoor, N. Nerurkar, A. Ganguly, and A. Kwasinski, "Cdma enabled wireless network-on-chip," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 10, no. 4, p. 28, 2014.

[96] M. Baharloo, A. Khonsari, P. Shiri, I. Namdari, and D. Rahmati, "High-average and guaranteed performance for wireless networks-on-chip architectures," in *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2018, pp. 226–231.

[97] A. Mestres, S. Abadal, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "A mac protocol for reliable broadcast communications in wireless network-on-chip," in *Proceedings of the 9th International Workshop on Network on Chip Architectures*. ACM, 2016, pp. 21–26.

[98] N. Mansoor, S. Shamim, and A. Ganguly, "A Demand-Aware Predictive Dynamic Bandwidth Allocation Mechanism for Wireless Network-on-Chip," in *Proceedings of the SLIP '16*, 2016.

[99] S. H. Gade, S. S. Rout, M. Sinha, H. K. Mondal, W. Singh, and S. Deb, "A utilization aware robust channel access mechanism for wireless nocs," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[100] A. B. Achballah, S. B. Othman, and S. B. Saoud, "Problems and challenges of emerging technology networks- on- chip: A review," *Microprocessors and Microsystems*, vol. 53, pp. 1–20, 2017.

[101] S. Abadal, A. Mestres, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "Medium access control in wireless network-on-chip: a context analysis," *IEEE Communications Magazine*, vol. 56, no. 6, pp. 172–178, 2018.

[102] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot et al., "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[103] H. Mao, M. Schwarzkopf, S. B. Venkatakrishnan, Z. Meng, and M. Alizadeh, "Learning scheduling algorithms for data processing clusters," *arXiv preprint arXiv:1810.01963*, 2018.

[104] S. Abadal, R. Martínez, J. Solé-Pareta, E. Alarcón, and A. Cabellos-Aparicio, "Characterization and modeling of multicast communication in cache-coherent manycore processors," in *Computers & Electrical Engineering*, 2016.

[105] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, 2019.

[106] A. Gomes, D. F. Macedo, and L. F. Vieira, "Automatic mac protocol selection in wireless networks based on reinforcement learning," *Computer Communications*, vol. 149, pp. 312–323, 2020.

[107] S. Galzarano, A. Liotta, and G. Fortino, "Ql-mac: A q-learning based mac for wireless sensor networks," in *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 2013, pp. 267–275.

[108] T. Lee, O. Jo, and K. Shin, "Corl: Collaborative reinforcement learning-based mac protocol for iot networks," *Electronics*, vol. 9, no. 1, p. 143, 2020.

[109] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-lm: Training multi-billion parameter language models using model parallelism," *arXiv preprint arXiv:1909.08053*, 2019.

[110] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[111] Y. Liu, Y. Wang, R. Yu, M. Li, V. Sharma, and Y. Wang, "Optimizing {CNN} model inference on cpus," in *2019 {USENIX} Annual Technical Conference ({USENIX}{ATC} 19)*, 2019, pp. 1025–1040.

[112] J. Hanhirova, T. Kämäräinen, S. Seppälä, M. Siekkinen, V. Hirvisalo, and A. Ylä-Jääski, "Latency and throughput characterization of convolutional neural networks for mobile computer vision," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 204–215.

[113] R. Ubal, B. Jang, P. Mistry, D. Schaa, and D. Kaeli, "Multi2sim: a simulation framework for cpu-gpu computing," in *2012 21st International Conference on Parallel Architectures and Compilation Techniques (PACT)*. IEEE, 2012, pp. 335–344.

[114] D. Shah, J. Shin et al., "Randomized scheduling algorithm for queueing networks," *The Annals of Applied Probability*, vol. 22, no. 1, pp. 128–171, 2012.

[115] V. Soteriou, H. Wang, and L. Peh, "A Statistical Traffic Model for On-Chip Interconnection Networks," in *Proceedings of MASCOTS '06*, 2006.

[116] S. Rajagopalan, D. Shah, and J. Shin, "Network adiabatic theorem: an efficient randomized protocol for contention resolution," in *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*, 2009, pp. 133–144.

[117] J. Ni, B. Tan, and R. Srikant, "Q-csma: Queue-length-based csma/ca algorithms for achieving maximum throughput and low delay in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 825–836, 2011.

[118] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, 2009.

[119] S.-Y. Yun, Y. Yi, J. Shin, and D. Y. Eun, "Optimal CSMA: A survey." in *ICCS*, 2012, pp. 199–204.

[120] R. Chandra, L. Dagum, D. Kohr, R. Menon, D. Maydan, and J. McDonald, *Parallel programming in OpenMP*. Morgan kaufmann, 2001.

[121] W. Gropp, W. D. Gropp, E. Lusk, A. D. F. E. E. Lusk, and A. Skjellum, *Using MPI: portable parallel programming with the message-passing interface*. MIT press, 1999, vol. 1.

[122] Y. Solihin, *Fundamentals of parallel multicore architecture*. CRC Press, 2015.

[123] D. Culler, J. P. Singh, and A. Gupta, *Parallel computer architecture: a hardware/software approach*. Gulf Professional Publishing, 1999.

[124] G. De Micheli and L. Benini, "Networks on chips: 15 years later," *Computer*, no. 5, pp. 10–11, 2017.

[125] S. Pasricha and N. Dutt, *On-chip communication architectures: system on chip interconnect*. Morgan Kaufmann, 2010.

[126] D. Wentzlaff, P. Griffin, H. Hoffmann, L. Bao, B. Edwards, C. Ramey, M. Mattina, C.-C. Miao, J. F. Brown III, and A. Agarwal, "On-chip interconnection architecture of the tile processor," *IEEE Micro*, vol. 27, no. 5, pp. 15–31, 2007.

[127] S. Vangal, J. Howard, G. Ruhl, S. Dighe, H. Wilson, J. Tschanz, D. Finan, A. Singh, T. Jacob, S. Jain, V. Erraguntla, C. Roberts, Y. Hoskote, N. Borkar, and S. Borkar, "An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 29–41, 2008.

[128] Altera, "An alternative to bus-based interconnects for large-scale design," in *White Paper*, 2008.

[129] J. Oh, M. Prvulovic, and A. Zajic, "Tlsync: support for multiple fast barriers using on-chip transmission lines," in *2011 38th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2011, pp. 105–115.

[130] R. Kumar, V. Zyuban, and D. M. Tullsen, "Interconnections in multi-core architectures: Understanding mechanisms, overheads and scaling," in *32nd International Symposium on Computer Architecture (ISCA'05)*. IEEE, 2005, pp. 408–419.

[131] D. Sánchez, *et al.*, "An Analysis of On-Chip Interconnection Networks for Large-Scale Chip Multiprocessors," *ACM T. Archit. Code Op.*, vol. 7, no. 1, 2010.

[132] T. Krishna, *et al.*, "Towards the ideal on-chip fabric for 1-to-many and many-to-1 communication," in *Proceedings of the MICRO-44*, 2011.

[133] X. Xiang, *et al.*, "A model for application slowdown estimation in on-chip networks and its use for improving system fairness and performance," in *ICCD '16*, 2016.

[134] S. Abadal, *et al.*, "OrthoNoC: A Broadcast-Oriented Dual-Plane Wireless Network-on-Chip Architecture," *IEEE Trans. Parallel Distrib. Syst.*, 2018.

[135] R. Kumar, T. Mattson, G. Pokam, and R. V. D. Wijngaart, "The case for message passing on many-core chips," in *Multiprocessor System-on-Chip*. Springer, 2011, pp. 115–123.

[136] S. Abadal, A. Mestres, M. Nemirovsky, H. Lee, A. González, E. Alarcón, and A. Cabellos-Aparicio, "Scalability of broadcast performance in wireless network-on-chip," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 12, pp. 3631–3645, 2016.

[137] S. Abadal, M. Iannazzo, M. Nemirovsky, A. Cabellos-Aparicio, H. Lee, and E. Alarcón, "On the area and energy scalability of wireless network-on-chip: A model-based benchmarked design space exploration," *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 5, pp. 1501–1513, 2015.

[138] R. K. Dokania and A. B. Apsel, "Analysis of challenges for on-chip optical interconnects," in *Proceedings of the 19th ACM Great Lakes symposium on VLSI*. ACM, 2009, pp. 275–280.

[139] D. Fritsche, *et al.*, "A Low-Power SiGe BiCMOS 190-GHz Transceiver Chipset With Demonstrated Data Rates up to 50 Gbit/s Using On-Chip Antennas," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 9, pp. 3312–3323, 2017.

[140] X. Timoneda, S. Abadal, A. Franques, D. Manessis, J. Zhou, J. Torrellas, E. Alarcón, and A. Cabellos-Aparicio, "Engineer the channel and adapt to it: Enabling wireless intra-chip communication," *arXiv preprint arXiv:1901.04291*, 2018.

[141] H. M. Cheema and A. Shamim, "The last barrier: On-chip antennas," *IEEE Microw. Mag.*, vol. 14, no. 1, pp. 79–91, 2013.

[142] X. Timoneda, S. Abadal, A. Cabellos-Aparicio, D. Manessis, J. Zhou, A. Franques, J. Torrellas, and E. Alarcón, "Millimeter-wave propagation within a computer chip package," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[143] S. Laha, *et al.*, "A New Frontier in Ultralow Power Wireless Links: Network-on-Chip and Chip-to-Chip Interconnects," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 2, pp. 186–198, 2015.

[144] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[145] S. Joseph, R. Misra, and S. Katti, "Towards Self-Driving Radios: Physical-Layer Control using Deep Reinforcement Learning," in *Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*. ACM, 2019, pp. 69–74.

[146] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. ACM, 2016, pp. 50–56.

[147] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers et al., "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 2017, pp. 1–12.

[148] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE journal of solid-state circuits*, vol. 52, no. 1, pp. 127–138, 2016.

[149] M. Ahmad, F. Hijaz, Q. Shi, and O. Khan, "Crono: A benchmark suite for multithreaded graph algorithms executing on futuristic multicores," in *2015 IEEE International Symposium on Workload Characterization*. IEEE, 2015, pp. 44–55.

[150] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*. ACM, 2008, pp. 72–81.

[151] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The splash-2 programs: Characterization and methodological considerations," *ACM SIGARCH computer architecture news*, vol. 23, no. 2, pp. 24–36, 1995.

[152] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[153] D. DiTomaso, A. Kodi, S. Kaya, and D. Matolak, "iWISE: Inter-router wireless scalable express channels for network-on-chips (NoCs) architecture," in *2011 IEEE 19th Annual Symposium on High Performance Interconnects*. IEEE, 2011, pp. 11–18.

[154] S. Deb, A. Ganguly, K. Chang, P. Pande, B. Beizer, and D. Heo, "Enhancing performance of network-on-chip architectures with millimeter-wave wireless interconnects," in *ASAP 2010-21st IEEE International Conference on Application-specific Systems, Architectures and Processors*. IEEE, 2010, pp. 73–80.

[155] D. Zhao and Y. Wang, "Sd-mac: Design and synthesis of a hardware-efficient collision-free qos-aware mac protocol for wireless network-on-chip," *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1230–1245, 2008.

[156] D. Zhao, Y. Wang, J. Li, and T. Kikkawa, "Design of multi-channel wireless noc to improve on-chip communication capacity," in *Proceedings of the Fifth ACM/IEEE International Symposium on Networks-on-Chip*. IEEE, 2011, pp. 177–184.

[157] C. Wang, W.-H. Hu, and N. Bagherzadeh, "A wireless network-on-chip design for multicore platforms," in *2011 19th International Euromicro conference on parallel, distributed and network-based processing*. IEEE, 2011, pp. 409–416.

[158] J. H. Bahn and N. Bagherzadeh, "Efficient parallel buffer structure and its management scheme for a robust network-on-chip (noc) architecture," in *Computer Society of Iran Computer Conference*. Springer, 2008, pp. 98–105.

[159] A. Ganguly, K. Chang, S. Deb, P. P. Pande, B. Belzer, and C. Teuscher, "Scalable hybrid wireless network-on-chip architectures for multicore systems," *IEEE Transactions on Computers*, vol. 60, no. 10, pp. 1485–1502, 2011.

[160] M. Rahaman and M. Chowdhury, "Improved bit error rate performance in intra-chip rf/wireless interconnect systems," in *Proc. ACM/IEEE Great Lake Symp. VLSI*, 2008.

[161] P. Dai, J. Chen, Y. Zhao, and Y.-H. Lai, "A study of a wire–wireless hybrid noc architecture with an energy-proportional multicast scheme for energy efficiency," *Computers and Electrical Engineering*, vol. 45, pp. 402–416, 2015.

[162] J. Yin, Y. Eckert, S. Che, M. Oskin, and G. H. Loh, "Toward more efficient noc arbitration: A deep reinforcement learning approach," in *Proc. IEEE 1st Int. Workshop AI-assisted Des. Architecture*, 2018.

[163] Y. Zeng and X. Guo, "Long short term memory based hardware prefetcher: A case study," in *Proceedings of the International Symposium on Memory Systems*. ACM, 2017, pp. 305–311.

[164] D. DiTomaso, A. Sikder, A. Kodi, and A. Louri, "Machine learning enabled power-aware network-on-chip design," in *Proceedings of the Conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2017, pp. 1354–1359.

[165] Z. Liu and I. Elhanany, "Rl-mac: a reinforcement learning based mac protocol for wireless sensor networks," *International Journal of Sensor Networks*, vol. 1, no. 3-4, pp. 117–124, 2006.

[166] S. Chinchali, P. Hu, T. Chu, M. Sharma, M. Bansal, R. Misra, M. Pavone, and S. Katti, "Cellular network traffic scheduling with deep reinforcement learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[167] U. Challita, L. Dong, and W. Saad, "Deep learning for proactive resource allocation in lte-u networks," in *European wireless technology conference*, 2017.

[168] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, 2018.

[169] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2018.

[170] N. Mastronarde, J. Modares, C. Wu, and J. Chakareski, "Reinforcement learning for energy-efficient delay-sensitive csma/ca scheduling," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–7.

[171] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[172] H. Bayat-Yeganeh, V. Shah-Mansouri, and H. Kebriaei, "A multi-state q-learning based csma mac protocol for wireless networks," *Wireless Networks*, vol. 24, no. 4, pp. 1251–1264, 2018.

[173] R. Ali, N. Shahin, Y. B. Zikria, B.-S. Kim, and S. W. Kim, "Deep reinforcement learning paradigm for performance optimization of channel observation–based mac protocols in dense wlans," *IEEE Access*, vol. 7, pp. 3500–3511, 2018.

[174] S. Amuru, Y. Xiao, M. van der Schaar, and R. M. Buehrer, "To send or not to send-learning mac contention," in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[175] A. Thierer and A. Castillo, "Projecting the growth and economic impact of the internet of things," *George Mason University, Mercatus Center, June*, vol. 15, 2015.

[176] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "Farmbeats: An iot platform for data-driven agriculture," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 2017, pp. 515–529.

[177] S. Kumar and A. Jasuja, "Air quality monitoring system based on IoT using raspberry pi," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2017, pp. 1341–1346.

[178] E. Manavalan and K. Jayakrishna, "A review of internet of things (IoT) embedded sustainable supply chain for industry 4.0 requirements," *Computers & Industrial Engineering*, vol. 127, pp. 925–953, 2019.

[179] A. Bansal, A. Gadre, V. Singh, A. Rowe, B. Iannucci, and S. Kumar, "Owll: Accurate lora localization using the tv whitespaces," in *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021)*, 2021, pp. 148–162.

[180] R. Elbakly and M. Youssef, "Crescendo: An infrastructure-free ubiquitous cellular network-based localization system," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–6.

[181] R. Nandakumar, V. Iyer, and S. Gollakota, "3d localization for sub-centimeter sized devices," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 108–119.

[182] M. Ibrahim and M. Youssef, "A hidden markov model for localization using low-end gsm cell phones," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–5.

[183] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single wifi access point," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, 2016, pp. 165–178.

[184] J. Xiong, K. Sundaresan, and K. Jamieson, "Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 537–549.

[185] Y. Hao, M. Chen, L. Hu, J. Song, M. Volk, and I. Humar, "Wireless fractal ultra-dense cellular networks," *Sensors*, vol. 17, no. 4, p. 841, 2017.

[186] P. Subrahmanya and A. Farajidana, "5g and beyond: Physical layer guiding principles and realization," *Journal of the Indian Institute of Science*, vol. 100, pp. 263–279, 2020.

[187] 3GPP, "Study on narrow-band internet of things (NB-IoT) / enhanced machine type communication (eMTC) support for non-terrestrial networks (NTN)," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 36.763, 06 2021.

[188] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5g networks for the internet of things: Communication technologies and challenges," *IEEE access*, vol. 6, pp. 3619–3647, 2017.

[189] S. Gong, Y.-H. Song, T. Manzaneque, R. Lu, Y. Yang, and A. Kourani, "Lithium niobate mems devices and subsystems for radio frequency signal processing," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2017, pp. 45–48.

[190] J. Guan, J. Zhang, R. Lu, H. Seo, J. Zhou, S. Gong, and H. Hassanieh, "Efficient wideband spectrum sensing using MEMS acoustic resonators," in *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*. USENIX Association, Apr. 2021, pp. 809–825.

[191] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi, "Ghz-wide sensing and decoding using the sparse fourier transform," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 2256–2264.

[192] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 269–282.

[193] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 71–84.

[194] Y. Xie, J. Xiong, M. Li, and K. Jamieson, "md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[195] M. Ibrahim and M. Youssef, "Cellsense: An accurate energy-efficient gsm positioning system," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 286–296, 2011.

[196] H. Rizk, A. Shokry, and M. Youssef, "Effectiveness of data augmentation in cellular-based localization using deep learning," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–6.

[197] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-id sequence matching," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011, pp. 293–306.

[198] H. Aly and M. Youssef, "Dejavu: an accurate energy-efficient outdoor localization system," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2013, pp. 154–163.

[199] A. Shokry, M. Torki, and M. Youssef, "Deeploc: a ubiquitous accurate and low-overhead outdoor cellular localization system," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2018, pp. 339–348.

[200] F. Wen, H. Wymeersch, B. Peng, W. P. Tay, H. C. So, and D. Yang, "A survey on 5g massive mimo localization," *Digital Signal Processing*, vol. 94, pp. 21–28, 2019.

[201] J. Palacios, P. Casari, and J. Widmer, "Jade: Zero-knowledge device localization and environment mapping for millimeter wave systems," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[202] H. Sallouha, A. Chiumento, and S. Pollin, "Localization in long-range ultra narrow band IoT networks using rssi," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[203] S. Gezici and Z. Sahinoglu, "Uwb geolocation techniques for ieee 802.15.4a personal area networks," *MERL Technical report*, 2004.

[204] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal processing magazine*, vol. 22, no. 4, pp. 41–53, 2005.

[205] J. Wang and D. Katabi, "Dude, where's my card? rfid positioning that works with multipath and non-line of sight," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013, pp. 51–62.

[206] I. Guvenc and C.-C. Chong, "A survey on toa based wireless localization and nlos mitigation techniques," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.

[207] M. Bouet and A. L. Dos Santos, "Rfid tags: Positioning principles and localization techniques," in *2008 1st IFIP Wireless Days*. IEEE, 2008, pp. 1–5.

[208] V. Iyer, R. Nandakumar, A. Wang, S. B. Fuller, and S. Gollakota, "Living iot: A flying wireless platform on live insects," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '19, 2019.

[209] S. Naderiparizi, Y. Zhao, J. Youngquist, A. P. Sample, and J. R. Smith, "Self-localizing battery-free cameras," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2015, pp. 445–449.

[210] C. Yang and H.-R. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.

[211] S. Boonsriwai and A. Apavatjrut, "Indoor wifi localization on mobile devices," in *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. IEEE, 2013, pp. 1–5.

[212] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, 2012, pp. 293–304.

[213] A. Marcaletti, M. Rea, D. Giustiniano, V. Lenders, and A. Fakhreddine, "Filtering noisy 802.11 time-of-flight ranging measurements," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 2014, pp. 13–20.

[214] Y. Song and S. Gong, "Wideband spurious-free lithium niobate rf-mems filters," *Journal of Microelectromechanical Systems*, vol. 26, no. 4, pp. 820–828, 2017.

[215] C. Zuo, N. Sinha, and G. Piazza, "Very high frequency channel-select mems filters based on self-coupled piezoelectric AlN contour-mode resonators," *Sensors and Actuators A: Physical*, vol. 160, no. 1, pp. 132 – 140, 2010.

[216] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[217] A. Omri, M. Shaqfeh, A. Ali, and H. Alnuweiri, "Synchronization procedure in 5g nr systems," *IEEE Access*, vol. 7, pp. 41 286–41 295, 2019.

[218] "Cell Mapper cell tower locations," https://www.cellmapper.net, accessed: Mon, Sep 13, 2021.

[219] P. Duplessis, "Hsopa: Exploiting ofdm and mimo to take umts beyond hsdpa/hsupa," *Nortel Technical Journal, Issue 2, August 2005*.

[220] H.-S. P. Wong and S. Salahuddin, "Memory leads the way to better computing," *Nature nanotechnology*, vol. 10, no. 3, p. 191, 2015.

[221] S. K. Gonugondla, B. Shim, and N. R. Shanbhag, "Perfect error compensation via algorithmic error cancellation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 966–970.

[222] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 10–14.

[223] M. Kadota, S. Tanaka, Y. Kuratani, and T. Kimura, "Ultrawide band ladder filter using SH0 plate wave in thin LiNbO3 plate and its application," in *2014 IEEE International Ultrasonics Symposium*, 2014, pp. 2031–2034.

[224] R. Lu, T. Manzaneque, Y. Yang, J. Zhou, H. Hassanieh, and S. Gong, "Rf filters with periodic passbands for sparse fourier transform-based spectrum sensing," *Journal of Microelectromechanical Systems*, vol. 27, no. 5, pp. 931–944, 2018.