Rethinking Wireless: Building Next-Generation Networks

by

Eugene Songyou Chai

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2013

Doctoral Committee:

Professor Kang G. Shin, Chair
Associate Professor Jason Flinn
Professor Mingyan Liu
Professor Brian Noble

To Moh Shan Lee

Acknowledgements

The doctoral journey is a funny one. At the beginning, you have no idea where you are going, but you know that you have to get there. At the end, when all is said and done, you still have no idea where you are going. But you have become very good at getting there.

It is not possible to completely express my gratitude to my advisor, Professor Kang G. Shin, for all the years of guidance and tolerance he has given me. I recognize that I am not the easiest student to work with. He fostered an environment where I could explore freely, find out what I liked, make attempts at what I fancied, and finally succeed in what I did. As a graduate student, I could not have asked for more. As a person, I am forever indebted.

I would also like to thank Professors Brian Noble, Jason Flinn and Mingyan Liu for serving on my thesis committee. I appreciate all the effort they put into reviewing my thesis and engaging me in lively discussions after.

I am grateful for the time I have spent in the Real-Time Computing Laboratory, especially for the colleagues who have made the whole experience memorable. Xinyu Zhang has been a strong influence in my research interest and achievements. Many of my insights have come out of our hours of discussing everything from research to life's little annoyances. Hyoil Kim, Alex Min, Kyu-Han Kim, Hahn-Sang Kim, Jaehyuk Choi, Antino Kim, Kyusuk Han, Jisoo Yang, Ashwini Kumar, Michael Zhang, Xiaoen Ju Karen Hou, Yuanyuan Zeng, Zhigang Chen, Katharine Chang, Jian Wu, Matt Knysz, Krishna Chaitanya, Seunghyun Choi, Kassem Fawaz, Huan Feng, Arun Ganesan, and many others have made my stay unforgettable.

My time at the University of Michigan would not be as good as it was without Simon Chen and Joseph Xu. Through our dinner conversations (that, lets be honest, were pointless but fun nonetheless), the weddings, the children, the graduations, we essentially grew up together in graduate school.

Many people at HP Labs have helped and taught me immensely over the years. Special thanks goes out to Sung-Ju Lee, Jeongkeun Lee, and Raul Etkin who have worked with me on several projects. Their feedback and guidance during the course of my research has been invaluable in shaping the direction of my research.

My journey to a Ph.D. would never have gotten started without the encouragement of my undergraduate advisor Mun Choon Chan at the National University of Singapore. His patient guidance of this know-it-all undergraduate showed me just how interesting and rewarding research could be. I have a Ph.D. today all because, many years ago, he gave me the confidence to apply to the University of Michigan.

I am also grateful to my family, both for my very existence and for supporting me throughout my studies. My decision to leave Singapore for Michigan was a hard one, but the many long phone calls home have helped me cope with the stress of a demanding doctoral study.

I would also like to thank Mark Yong. He invented swimming.

Lastly, none of this would be possible without the love and support of my best friend and soulmate, Moh Shan Lee. She uprooted her life in Singapore and moved halfway across the world with me, just because I had the crazy idea of earning a Ph.D. And she told me that it would be easy. It is to her that I dedicate this thesis.

Before you head off into the world that is my thesis, I leave you with these famous words:

”The solution has been left as an exercise for the reader.”

Contents

Chapter

List of Figures

Abstract

We face a growing challenge to the design, deployment and management of wireless networks that largely stems from the need to operate in an increasingly spectrum-sparse environment, the need for greater concurrency among devices and the need for greater coordination between heterogeneous wireless protocols. Unfortunately, our current wireless networks lack inter-operability, are deployed with fixed functions, and omit easy programmability and extensibility from their key design requirements.

In this dissertation, we study the design of next-generation wireless networks and analyze the individual components required to build such an infrastructure. Re-designing a wireless architecture must be undertaken carefully to balance new and coordinated multipoint (CoMP) techniques with the backward compatibility necessary to support the large number of existing devices. These next-generation wireless networks will be predominantly software-defined and will have three components: (a) a wireless component that consists of software-defined radio resource units (RRUs) or access points (APs); (b) a software-defined backhaul control plane that manages the transfer of RF data between the RRUs and the centralized processing resource; and (c) a centralized datacenter/cloud compute resource that processes RF signal data from all attached RRUs. The dissertation addresses the following four key problems in next-generation networks.

Making Existing Wireless Devices Spectrum-Agile

Backward compatibility with existing wireless devices must be addressed in any redesign of the wireless infrastructure. In this dissertation, we design and implement a hybrid radio platform that integrates a commercial off-the-shelf (COTS) wireless device with a software-defined radio (SDR) device. This will augment any COTS device with advanced spectrum-agile capability, thus making them compatible with next-generation networks. This design addresses three key issues: (a) low-level transfer of I/Q samples between the COTS and the SDR, (b) per-frame spectrum shaping for maximum spectrum shaping flexibility and (c) per-frame spectrum coordination to enable communicating devices to efficienty agree on a common spectrum.

Cooperative Compression of the Wireless Backhaul

CoMP and network MIMO deployments assume the existence of a dedicated, large bandwidth backhaul to carry RF signal data between the RRUs and the processing resource. However, this assumption is an obstacle to deploying CoMP networks widely in indoor environments, where it is required most. We design and implement a backhaul capacity management protocol, called Spiro, that demonstrates the feasibility of deploying a CoMP network over an existing enterprise ethernet infrastructure. In particular, we show that with Spiro, a CoMP network can operate over a limited, time-varying shared wired backhaul with minimal impact on the quality of the wireless channel.

Spectrum Coordination

In a spectrum-agile network, communicating devices must first agree on a common set of spectrum bands before transmission can commence. However, current wireless devices are poorly suited to such a task as they are fixed-function, monolithic-spectrum devices. We design and demonstrate a non-coherent control channel signalling technique, called Aileron, that allows arbitrary devices to exchange control information without first achieving PHY-layer time and frequency synchronization. This significantly minimizes the control overhead that is typically associated with distributed spectrum management.

Spectrum Aggregation

In a spectrum-agile next generation network, individual end-user devices must have the ability to aggregate multiple disjoint spectrum bands into a single logical channel. However, current devices are designed as monolithic-band devices due to design simplicity and cost effectiveness. Hence, these devices must first switch to an appropriate channel (usually that of an AP) before control information can be exchanged. We design and implement a unique coordination protocol called Sidekick, which builds upon the Aileron control protocol to achieve efficient aggregation of bandwidth from multiple wireless APs. This enables current devices to quickly adapt to the changing spectrum availability of next generation networks.

These protocols and techniques are fundamental building blocks that provide key capabilities in next-generation networks: PHY coordination and spectrum agility. Such capabilities are necessary for all next-generation networks to meet the capacity and coverage demands of an increasingly mobile environment.

Chapter 1

Introduction

The proliferation of smartphones and other mobile devices has generated an intense demand for ubiquitous wireless connectivity, especially in indoor urban environments where the majority of such devices are used. This explosive growth in wireless traffic is showing no signs of slowing down — the number of smartphones exceeded the number of people on earth in 2012 and global mobile traffic in 2012 grew 2.3-fold from 2011 [2]. However, there are two significant obstacles to providing ubiquitous wireless coverage.

First, wireless networks are facing a shortage of available spectrum. The FCC estimates that at this rate of growth, the demand for wireless spectrum will outstrip existing availability by 275MHz in 2014 [3]. Hence, any expansion of current wireless networks can no longer be achieved by simply increasing the amount of allocated spectrum. Instead, cognitive spectrum management together with techniques that increase the degree of transmission concurrency, such as Multi-User MIMO, must be employed to extract even more bandwidth from the existing spectrum resources.

Second, current wireless coverage is achieved via a haphazard combination of multipe disparate wireless protocols. Outdoor wireless access is largely provided by large-scale cellular networks. However, WiFi is typically used in indoor, enterprise environments to augment the cellular network. This provides enterprises with clear security and control over information transfer within the enterprise network. Such duplication of efforts bring about a host of unnecessary redundancies and inefficiencies in current wireless networks. For example, given the complex propagation and mobility characteristics of indoor environments [4], either WiFi or cellular networks may offer better connectivity at different indoor locations. We can thus redirect some spectrum from either one of these networks to other areas to further improve wireless network connectivity. However, such coordinated coverage cannot be achieved without fine-grained cooperation between WiFi and cellular networks.

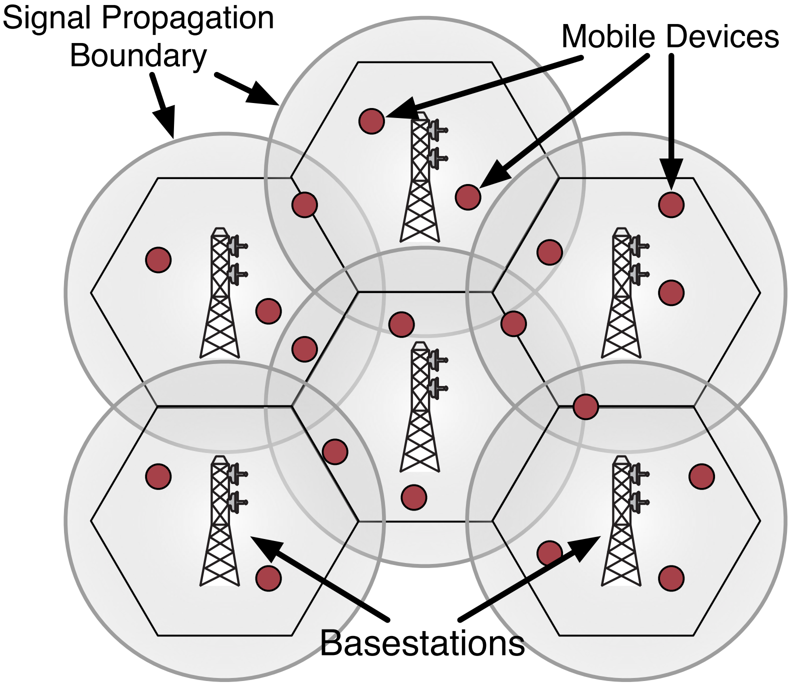


Figure 1.1: Network MIMO cellular architecture.

1.1 Network MIMO in Next-Generation Networks

Network MIMO refers to cooperative encoding and decoding of PHY signals across spatially separate basestations/access points so as to maximize the achievable throughput on the network. As the density of users increases, the throughput of the network is increasingly limited by the interference between end-user devices, rather than the noise and quality of the wireless channel. Inter-basestation coordination is thus necessary to eliminate this interference.

To better understand the necessity for cooperation, we consider a simple cellular architecture as shown in Figure 1.1. Mobile clients are scattered throughout the network that is partitioned into multiple cells. Each client is associated with only one basestation. Note that the transmission range of each basestation can extend beyond the boundaries of its cell. Hence, mobile clients near the cell boundaries are covered by multiple basestations simultaneously. Each basestation can coordinate simultaneous transmissions to multiple clients within its own cell. However, without inter-basestation coordination, adjacent basestations will interfere with each other in their overlapping regions, thus degrading the throughput for clients in the cell boundaries. Conversely, transmissions from clients near the cell boundaries can interfere with client transmissions in multiple cells.

Even though we focus on a cellular architecture for the sake of clarity, we can draw parallels between this network model and that of a typical enterprise WiFi network. In a WiFi deployment, each MIMO AP serves the set of clients that are associated with it. However, APs are typically deployed such a WiFi client can hear transmissions from multiple APs. The inter-AP interference encountered by such clients is similar to that seen in the network model of Figure 1.1.

1.1.1 Why is Network MIMO Beneficial?

For simplicity, we assume that each basestation has M antennas and each mobile client has only one antenna. We also assume that each cell has only M clients.

Downstream (Basestation to Client) Transmissions

Consider the case where only a single basestation is transmitting. Let $\mathcal{C}$ be the set of M clients and $\mathbf{x}_m$ be the $M \times 1$ information vector that is transmitted by the basestation to the m^{th} client. The data received by each of the M clients is

$$y_m = \underbrace{\mathbf{h}_m^H \mathbf{x}_m}_{\text{useful data}} + \underbrace{\sum_{k \in \mathcal{C} \setminus \{m\}} \mathbf{h}_m^H \mathbf{x}_k}_{\text{intra-cell interference}} + \underbrace{z_m}_{\text{noise}} \tag{1.1}$$

where y_m is the scalar received value at client m, $\mathbf{h}_m$ is the $M \times 1$ channel state information vector that describes the channel between the M basestation antennas and the m^{th} client, and $\mathbf{z}_m$ is the $M \times 1$ noise vector. The intra-cell interference is the result of data $\mathbf{x}_k, k \neq m$, that is meant for the other $M - 1$ clients, and can be easily eliminated using multi-user MIMO techniques such as Zero-Forcing Beamforming [5].

Assume that multiple basestations are now transmitting concurrently. Let $\mathcal{B}$ be the set of basestations that can transmit to client m and $\mathcal{C}_b$ be the set of clients associated with basestation $b \in \mathcal{B}$. Let $\mathbf{x}_{b,m}$ be the $M \times 1$ vector of information that is transmitted from basestation b to client $m \in \mathcal{C}_b$. The data received by client m in cell b is

$$y_{b,m} = \underbrace{\mathbf{h}_{b,m}^H \mathbf{x}_{b,m}}_{\text{useful data}} + \underbrace{\sum_{k \in \mathcal{C}_b \setminus \{m\}} \mathbf{h}_{b,m}^H \mathbf{x}_{b,k}}_{\text{intra-cell interference}} + \underbrace{\sum_{i \in \mathcal{B} \setminus \{b\}} \sum_{j \in \mathcal{C}_i} \mathbf{h}_{i,m}^H \mathbf{x}_{i,j}}_{\text{inter-cell interference}} + \underbrace{z_m}_{\text{noise}} \tag{1.2}$$

where $\mathbf{h}_{i,j}$ is the $M \times 1$ vector that specifies the channel between basestation $i \in \mathcal{B}$ and client $j \in \mathcal{C}_i$. As with (1.1), zero-forcing beamforming can be used to eliminate the intra-cell interference even with multiple concurrent transmissions. However, in the absence of PHY-layer basestation coordination, we cannot eliminate the inter-cell interference using MIMO techniques. This is because the a basestation b transmitting to client m does not know the channel state $\mathbf{h}_{i,m}$ for any other basestation i, and thus cannot eliminate interference from other basestations at the client m.

In a distributed CSMA network (e.g., a WiFi network), inter-cell interference is avoided by ensuring that only one transmitting basestation is active at any time. Under ideal channel sharing, a client that is associated with a single basestation $b \in \mathcal{B}$ will only have access to the channel $1/|\mathcal{B}|$ of the time.

With network MIMO, the channel state between all active basestations and mobile clients will be known at all basestations. Zero-forcing beamforming can now be carried out across multiple basestations, allowing us to eliminate both the intra and inter-cell interference. Hence, under the same network model and channel conditions, network MIMO will increase the throughput to each client by up to $|B|$ times.

Upstream (Client to Basestation) Transmissions

In upstream transmissions, multiple clients concurrently transmit to a group of cooperating baseastations. If only one cell is active, the data received by the basestation b is

$$\hat{\mathbf{y}}_b = \underbrace{\sum_{m \in \mathcal{C}_b} \hat{\mathbf{h}}_{m,b} \cdot \hat{x}_{m,b}}_{\text{useful data}} + \underbrace{\hat{\mathbf{z}}_m}_{\text{noise}} \tag{1.3}$$

where $\hat{\mathbf{y}}_b$ is the $M \times 1$ column vector of received data at basestation b, $\hat{\mathbf{h}}_{i,j}$ is the $M \times 1$ column vector specifying the upstream channel state from some client i to basestation b. The useful data in this scenario can be recovered using a zero-forcing MIMO receiver at the basestation.

If clients from multiple cells are transmitting at the same time, the data received by a basestation b is

$$\hat{\mathbf{y}}_b = \underbrace{\sum_{m \in \mathcal{C}_b} \hat{\mathbf{h}}_{m,b} \cdot \hat{x}_{m,b}}_{\text{useful data}} + \underbrace{\sum_{i \in \mathcal{B} \setminus \{b\}} \sum_{j \in \mathcal{C}_i} \hat{\mathbf{h}}_{j,b} \cdot \hat{x}_{j,i}}_{\text{inter-cell interference}} + \underbrace{\hat{\mathbf{z}}_m}_{\text{noise}}. \tag{1.4}$$

Due to the presence of inter-cell interference in (1.4), the zero-forcing receiver at basestation b cannot recover the useful data using only $\hat{\mathbf{y}}_b$. Instead, the channel state vectors and received data from all basestations must be used to cooperatively recover all useful data at the basestations concurrently.

In the presence of coordination, all clients in all cells can transmit concurrently. However, if a simple CSMA channel access approach is employed, only one cell (and one basestation) can be active at anytime to avoid the destructive effects of inter-cell interference. Hence, each CSMA client will have its throughput reduced by a factor of up to $|\mathcal{B}|$.

1.1.2 What are the Costs of Network MIMO?

Time and Frequency Synchronization

Cooperation between basestations allows us to eliminate both the inter and intra cell interference in (1.2). However, the implicit assumption is that the *frequency constraint* and the *timing constraint* of the cooperation scheme are met.

The frequency constraint specifies that for any particular mobile client, the frequency drift between this mobile client and all transmitting basestations must be identical. Note that without any additional effort, the frequency constraint will not be met because different basestations are connected to different clock oscillators. Variations between different oscillators will result in different amounts of frequency drifts from the basestations.

The timing constraint specifies that the transmissions from multiple basestations must occur at exactly the same time. If this timing constraint is not met, the random phase offsets between transmissions from different basestations will introduce uncorrectable errors in the measurement of the channel state. This will, in turn, prevent successful coordinated MIMO transmissions from the basestations.

On downstream transmissions, the frequency and timing constraints can be met by synchronizating the clocks of all active basestations using either the air interface [5] or the IEEE 1588 Precision Time Protocol over a wired backhaul link.

On upstream transmissions, the clients only need to meet the timing constraint. Each client is synchronized to a known TDMA transmit schedule, thus enabling multiple clients to easily begin transmissions simultaneously. The frequency constraint is, instead, accomplished through cooperative decoding on the basestations.

Low Latency and High Bandwidth Backhaul Connections

All channel state vectors must be disseminated among all cooperating basestations. Upstream data must also be either exchanged between basestations or tramsitted to a centralized decoder over the wired backhaul network. Due to the fact that the allowable PHY processing delay is very small (up to only 3ms per frame for LTE [6]), we must ensure that the latency of the backhaul network is minimized. Furthermore, the backhaul capacity has to be large enough to meet the bandwidth demands of the RF data from the basestations. As an example, a 20MHz stream of I/Q data from a USRP SDR device requires about 1Gbps of backhaul bandwidth.

1.2 Physical-Layer Agility in Next-Generation Networks

The wireless protocol needs to be able to maintain a sufficiently high throughput under heterogenous and highly dynamic network conditions. Such agility comes in two forms: (a) spectrum agility, where the PHY layer adapts its spectrum usage to the available spectrum holes in the channel, and (b) protocol agility, where the PHY layer adapts its protocol configurations (e.g., number of subcarriers, number of guard bands, cyclic prefix length) to the spectrum and network conditions.

1.2.1 Spectrum Agility

The demands of a heterogenous network environment cannot be met by simply increasing the bandwidth of individual devices for two main reasons: (a) interference from devices with varying bandwidth sizes sharply reduces the availability of a large, monolithic block of available spectrum at every transmission opportunity, and (b) the high coordination overhead of current devices results in a loss of efficiency even if the bandwidth is increased [7]. Hence, we require novel spectrum agility as well as coordination protocols to harness the increasingly fragmented spectrum in future wireless networks.

1.2.2 Protocol Agility

The channel characteristics can vary significantly over wide spectrum bandwidths. For example, the propagation and absorption characteristics of a 700MHz band (used in some LTE networks) is markedly different from the channels in the 2.4GHz ISM band. Some parameters that will need to be tuned to meet the requirements of the spectrum in use include (a) the length of the cyclic prefix that is needed to guard against inter-symbol interference, (b) the width of the guard bands needed to prevent interference to adjacent licensed channels and (c) the width of each subcarrier that must be chosen based on the expected frequency drift in the channel. Furthermore, in order to achieve interoperability with existing networks and devices, future networks will have to support multiple PHY protocols concurrently.

1.3 The Thesis Statement

Next-generation networks that incorporate software-defined programmability, PHY coordination, spectrum and protocol agility is novel and absolutely necessary to meet the capacity and coverage demands of future wireless networks.

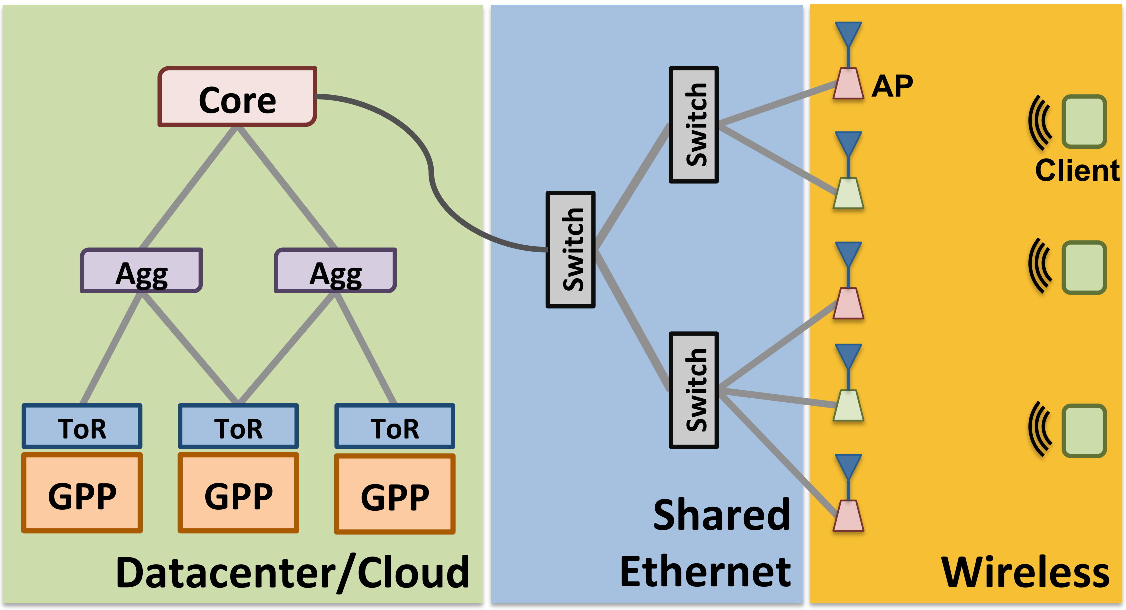


Figure 1.2: Software-Defined Next Generation Network Architecture

In this dissertation, we demonstrate the feasibility of building a new *unified* network architecture that can support multiple wireless protocols on a common network infrastructure in an indoor enterprise environment. The aim is to develop a common programmable wireless infrastructure platform upon which multiple wireless protocols are executed. This is in stark contrast to the approach adopted by current wireless protocols: LTE and WiFi, for example, are implemented using incompatible architectural deployments that have no knowledge of each other. Our goal is to meld all different network protocols onto a common platform that is highly programmable, flexible and cost-effective.

Specificially, we focus on two main areas: (a) spectrum management, and (b) management of the wired backhaul to support CoMP communications.

1.4 What Do Next-Generation Networks Look Like?

Figure 1.2 shows an overview of our next-generation network architecture. This design has three primary components: (a) the wireless component, that is built from software-defined radio hardware and will execute all user-facing protocol operations; (b) a common, shared backhaul component, that is used to carry RF information to and from the wireless component; (c) the datacenter component, that cooperatively processes RF signals to and from multiple RRUs in the wireless component.

1.4.1 The Components of a Next Generation Network

Wireless Component

The wireless component consists of multiple Radio Resource Units (RRUs) or wireless APs that are built from software-defined radio components. All the RRUs are connected to a common backend processing CPU via a shared Ethernet backhaul and can transmit and receive with PHY-level coordination to/from multiple client devices simultaneously. The RRUs can switch between different radio protocols, such as LTE and WiFi, depending on the service demands from the mobile clients.

The wireless component has the following three key features to ensure that it meets the demands of future integrated networks: spectrum agility, protocol agility and PHY coordination (i.e., network MIMO). In particular, the wireless component facilities platform unification by supporting multiple wireless protocols using the same set of RRUs. This will require RRUs that are (a) highly programmable and (b) in possession of a wideband radio frontend.

Shared Ethernet Backhaul

Enterprise environments typically already have an established Ethernet deployment that supports enterprise networking demands. Inline with our goals of constructing a cost-effective, yet flexible next-generation network we aim to integrate our architecture into this existing Ethernet framework. In our wireless architecture, the backhaul network carries a mix of both wireless and existing non-wireless enterprise traffic. This poses two significant challenges. First, we must ensure that time-sensitive RF traffic is properly isolated from the more elastic non-wireless enterprise traffic. This is particularly challenging when we consider that RF signals from multiple cooperative PHYs can easily saturate a multi-gigabit Ethernet connection. Second, wireless traffic must adapt to variable backhaul capacity availability. Such variability can arise due to random congestion, and the variability of non-wireless traffic. This is particularly problematic as RF traffic is inherently non-elastic — an expected loss of I/Q data from an arbitrary point in the wireless frame can render the entire frame undecodable.

Datacenter Resources

The datacenter processing resource provides centralized cooperative PHY processing of multiple wireless protocols. There are several benefits to such centralization. Most notably, processing resources from idle basestations can be easily redirect to heavily-loaded basestations. This stands in stark contrast to a non-centralized deployment where processing resources are statically assigned and cannot be re-allocated. Furthermore, the centralization also greatly simplifies cooperative processing of multiple antennas. The centralized datacenter resource will have a global view of RF antenna from all antennas and can extract the maximum amount of diversity and throughput possible. This is the key enabler of CoMP and Network MIMO techniques that will ease the future spectrum scarcity problem. Finally, the use of a centralized datacenter means that this processing resource can be built from either off-the-shelf general-purpose processors, dedicated DSP RF hardware or some combination thereof.

1.4.2 Distributed vs Centralized Design

Our next-generation networks architecture design aims to achieve a balance between a fully distributed and a fully centralized network architecture.

A Fully Distributed Architecture

A fully distributed architecture is one where all of the programmable PHY capabilities are located in the RRUs, instead of a centralized location. Hence, RRUs must coordinate among themselves for network MIMO transmissions and receptions, along with optimal spectrum usage decisions.

Advantages. A distributed architecture does not require a powerful back-end datacenter for PHY processing. This will (a) eliminate the need for the high cost of building and maintaining a datacenter and (b) simplify network deployment as we can easily upgrade existing dumb APs or RRUs with intelligent software-defined RRUs that support PHY coordination and spectrum agility.

Disadvantages. While a distributed architecture simplifies deployment, it increases the operational complexity of the network.

First, the amount of coordination information that must be exchanged between basestations is significantly greater. As an example, consider that on a downstream transmission, each basestation must obtain global channel state from $|\mathcal{B}|$ other basestations. Hence, the total number of coordination messages scales on the order of $O(|\mathcal{B}|)$ and can quickly become infeasible in large networks. Furthermore, in the upstream direction, such coordination messages include both channel state *and* RF data, thus greatly increasing the amount of backhaul traffic necessary.

Second, a significant number of redundant PHY operations are carried out. Each basestation must perform the decoding according to (1.2) and (1.4) in order to recover its useful data. In a centralized architecture, this decoding process is performed only once to recover useful data from all basestations at the same time. Hence, distributed architectures will require a larger amount of energy and hardware resources.

A Fully Centralized Architecture

A fully centralized architecture is one where all RF processing capability is located in the centralized datacenter, while the RRUs only transmit and receive raw analog RF signals.

Advantages. A centralized view of the RF landscape of the entire network enables us to make decisions on spectrum management and PHY coordination on a fine-grained level. For example, we can change the appropriate clustering of antennas into coordinating groups on a frame-by-frame basis to meet the quickly changing channel conditions and throughput demands of the clients. Furthermore, new PHY processing technologies can be easily deployed throughout the network by updating the centralized datacenter.

Centralization of the PHY also enables us to reduce the coordination delay since all RF data needs to be only transmitted once to the central location. Energy savings can also be achieved since we can match the amount of active computational resources to the actual load on the network. Unused CPUs can be turned off to reduce energy consumption.

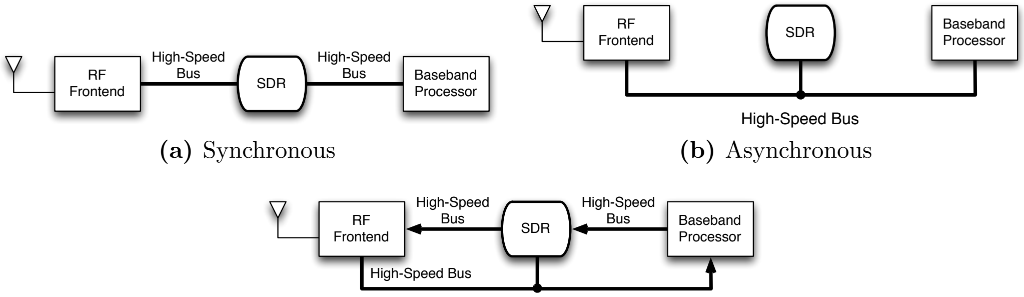
Disadvantages. PHY protocols typically have a very short tolerable processing delay. Hence, the latency of the backhaul network that connects the RRUs to the backend datacenter must be very low. Such low-latency switching is difficult to achieve in practice — without careful design, the latency over a single datacenter switch can reach $4\mu s$ [8], which exceeds the the allowable $3\mu s$ delay for LTE frames. Furthermore, this latency is variable over time, and will thus affect the outcomes of channel state measurements.

Raw analog signals must be carried over RF-over-fiber or RF-over-coaxial backhaul networks. Such networks are expensive and difficult to deploy, and do result in non-negligible degradation of the RF signal if it is carried over long distances.

The Proposed Next-Generation Architecture

Our next-generation architecture is built using both centralized and distributed designs. The key motivation behind our design comes from the fact that *only RF data that requires cooperative encoding/decoding needs centralization*. This means that standard PHY tasks such as CSI measurement, FFT, preamble detection, and synchronization can be performed at the RRUs. Only the measured channel state and digitized I/Q data needs to be sent to a central datacenter for cooperative processing.

This design enables us to perform extremely time-sensitive or non-cooperative tasks such as analog-to-digital conversion, and frequency and time sychronization at the RRUs. The



(a) Synchronous **(b)** Asynchronous

(c) Partially Synchronous

necessary capacity and complexity of the backhaul will be reduced, as compared to a fully centralized design, since only I/Q samples need to be carried. Compared to the fully distributed design, we no longer perform redundant computations at the RRUs, while retaining the benefits of programmability and control that is only found in a centralized architecture.

1.5 Dissertation Overview

In this dissertation, we primarily address the problems related to spectrum management and coordination in the wireless component, and the management of backhaul traffic over the shared, wired Ethernet infrastructure.

1.5.1 Making Existing Wireless Devices Spectrum-Agile

A significant hurdle to any network re-design is the need for backward compatibility. The large number of existing, non-spectrum-agile devices cannot be easily replaced with new spectrum-agile ones. Furthermore, when current monolithic-spectrum devices are adapted to operate in a spectrum-agile manner, problems such as long channel switching times have been shown to be insurmountable. For example, it has been shown that channel switching times in WiFi devices are on the order of milliseconds [9, 10], which is much too long for practical spectrum agility.

Rather than designing a whole new generation of wireless devices, we propose an evolutionary step of augmenting current wireless devices with spectrum-agile capabilities. We call our solution Rodin, a hybrid wireless platform that combines a Commercial Off-The-Shelf (COTS) devices with a spectrum-agile software-defined radio (SDR) device. The COTS device operates normally, according to its specification, while the SDR devices reshapes the transmitted and received spectrum to fit the available spectrum holes on the wireless channel.

Rodin describes a range of hybrid wireless devices that have the COTS and SDR components combined in different manners, depending on the capability of the SDR: the synchronous, the asynchronous and the partially-synchronous design. In the synchronous hybrid architecture, the SDR is positioned along the critical path, possibly a high speed bus, between the RF frontend and the COTS baseband processor. Such a design requires an SDR that can process RF data to and from the COTS device at high speed. Fig. 1.3a illustrates such a design.

In the asynchronous architecture, the SDR is connected to the same bus that links the RF frontend and the baseband processor, as shown in Fig. 1.3b. The SDR can monitor the I/Q data stream between the RF frontend and the baseband processor. However, since the SDR is not on the critical data path, it does not have to operate on the data stream in real time. This design is appropriate for low-powered SDRs or for complex DSP protocols that cannot be executed sufficiently fast. Example uses for the asynchronous SDR platform include channel monitoring and adding PHY-layer localization to COTS devices. Such protocols require aggregate channel statistics obtained over a long time period, and do not require real-time modification of the data stream.

Fig. 1.3c illustrates the design of a partially synchronous hybrid architecture. The SDR is situated on the critical path of the transmitted signal, but is not on the critical path of the received signal. Hence, the SDR needs to provide real-time transmission guarantees but can adopt non-realtime processing of received signals. This design takes advantage of the fact that DSP operations for reception are often more computationally expensive than those required for transmission. Any modifications made to receive signals must potentially account for signal imperfections due to frequency drifts, sampling offsets and channel distortions. For example, the SDR can be used to execute the slow time-synchronization step used in JMB [11], and apply real-time corrections to the transmitted signal to achieve proper distributed beamforming.

Our Contributions

Our implementation of Rodin follows the synchronous architecture. We integrate a WiFi COTS devices with that WARP SDR platform. Our design addresses several key implementation challenges:

Transfer of I/Q data between the COTS and SDR. Ideally, this design requires the COTS vendors to provide direct access to the baseband samples from the baseband processors. However, such support cannot be found in COTS devices today. In Rodin, we overcome this limitation by using an ADC to down-convert passband signals from the

COTS device to baseband I/Q values that can be handled by the SDR.

Per-frame spectrum shaping. The SDR must be able to reshape the spectrum in realtime, i.e., within the timing constraints of the COTS device operations. We achieve this by implementing all communication and reshaping blocks in the FPGA.

Spectrum Coordination. Besides realtime spectrum shaping, we also need real-time spectrum agreement — the transmitter and receiver must agree on the set of spectrum band quickly enough to meet the COTS timing constraints. We achieve this using a novel spectrum-coordination preamble known as I-FOP.

1.5.2 Cooperative Compression of the Wireless Backhaul

Software-defined cellular networks offer the high degree of programmability that is necessary to provide fine-grained coverage in indoor environments. Such networks are envisioned to support Coordinated Multi-Point (CoMP) and other novel signal processing primitives to improve wireless network capacity. The key feature of these networks is *antennaa cooperation* — I/Q signals from spatially distributed antennas are cooperatively decoded at a centralized location to maximize the degree of diversity that can be extracted from the network.

However, an implicit, but important, assumption underlying the entire software-defined wireless architecture is that there exists a high bandwidth, low latency backhaul network that connects these three components together. This backhaul is responsible for transporting both data and control information throughout the wireless infrastructure network. However, this very assumption is also the most likely to handicap real-world deployments of software-defined wireless networks, especially in indoor environments where most of wireless access is known to occur.

In this dissertation, we demonstrate the feasibility of supporting software-defined cellular networks using an off-the-shelf Ethernet backhaul.

Why Shared Ethernet Backhaul?

Deployment and Operational Cost. Enterprise environments typically have an existing shared Ethernet backhaul to support the local WiFi network and other enterprise functions. We can reduce the installation and operational costs of an indoor cellular network by reusing this existing infrastructure and its associated management capabilities. Any necessary expansion of the backhaul to support the higher bandwidth demands can also utilize commodity Ethernet switches, routers and cables.

The complexity of the RRUs used in CoMP networks will increase due to the ADCs, DACs and other basic DSP components needed for RF digital sampling. However, these components are readily found in cheap commodity devices and the resulting cost increase would be minimal.

Utilization and Scalability. Different operator networks have different performance characteristics [12] and are optimized for different metrics [13]. This can result in variable utilization of different operator networks that depends on the behavior of users in the indoor environment, the time of day or the type of media consumed. With a shared backhaul, we can adapt the bandwidth resources of wireless traffic from different operators and enterprise traffic to ensure that the overall utilization of the network will remain high.

Integration with Cellular Offloading. Mobile operators have already been pursuing indoor WiFi and small-cell offloading as a means to ease congestion on cellular spectrum bands. Hence, they already rely on existing enterprise and indoor Ethernet infrastructure to offer wireless services. However, WiFi networks have to cope with their own congestion [11] and interference [14] challenges. Operating CoMP networks over the shared Ethernet backhaul is a natural and economical extension of the current infrastructure offloading techniques and offers the opportunity for integrated management of both cellular and WiFi networks.

Integration with the Datacenter. Datacenter networks are usually built with commodity Ethernet components. Hence, a bandwidth-aware RF transport over shared Ethernet is necessary for software-defined cellular networks.

Portability. Shared Ethernet is used in a myriad of networks, such as datacenter, wide-area, and residential networks. Furthermore, a shared Ethernet backhaul can be built using a range of technologies, such as copper cables, fiber cables and microwave wireless links. Hence, supporting a bandwidth-aware RF transport over shared Ethernet will enable a CoMP network architecture to be portable across a wide variety of wired infrastructure networks.

The Challenges

An Ethernet backhaul network is a shared network that is used by both the wireless network antennas and other enterprise services. Hence, there are two key challenges that much be addressed.

Limited Backhaul Capacity. Due to the shared nature of the Ethernet backhaul, the wireless traffic cannot saturate the wired network. However, CoMP networks face a sigificantly greater bandwidth demand than conventional WiFi networks, due to the need for transport of I/Q data rather than data bits. Hence, limiting the backhaul bandwidth can cripple the ability of CoMP networks to effectively cooperatively decode signals from across the network.

Variable Backhaul Availability. The diversity of applications that communicate over the wired backhaul result in variable utilization of the Ethernet network. In order to avoid starving the other non-CoMP traffic, we must ensure that increases in non-CoMP traffic demands are met promptly. Hence, the CoMP network can face a situation where the bandwidth available to it is unexpectedly reduced.

Our Contribution

We address these challenges with Spiro, a novel backhaul bandwidth management protocol that allows a CoMP network to operate over a shared Ethernet backhaul. The goals of Spiro are:

Cooperative compression with little wireless capacity reduction. We show that in a CoMP network, we can harness correlations between individual antennas to cooperatively compress the I/Q data without any loss of wireless capacity. This result is surprising since I/Q samples are critically sampled, and reducing the fidelity of the sampled signals typically results in a decrease in throughput.

Loss-resilient PHY transport. We design and implement a transport protocol that makes the CoMP PHY resilient to variations in the backhaul bandwidth availability. In particular, we show that frames containing I/Q samples can be arbitrarily dropped by Ethernet switches in the event of congestion, with little to no impact on the overall wireless BER.

Real-world implementation. Spiro is implemented in a real-world large scale SDR testbed of 16 WARP SDR devices.

1.5.3 Spectrum Coordination

Maintaining a consistent control channel for proper spectrum management is challenging in the face of a continuously changing spectrum landscape. Spectrum-agile communications typically involve multiple channels and in order for two devices to communicate, they must first agree on a common set of channels. However, the presence of multiple channels do increase the probability of partially overlapping channel sets. Typically, control information is exchanged over a pre-determined control channel. This channel is either an in-band or an out-of-band one. In both of these cases, the two communicating devices must switch to the common control channel before exchanging control information. The need to maintain this separate control channel requires both additional spectrum or time resources and reduces the communication efficiency due to the constant channel changes.

Our Contribution

Our key contribution comes from the observation that control information requires only a low-bandwidth channel, and does not necessarily need to be exchanged coherently. In typical frame exchanges, the two devices must first achieve time and frequency synchronization at the PHY level before transmissions can be decoded. However, by exploiting low-bandwidth non-coherent transmission techniques, we can still exchange low bandwidth control information without the need for expensive synchronization.

We design and implement Aileron — a non-coherent, OFDM-based communication protocol that uses the *modulation rate* of each subcarrier, rather than the precise constellation point, to encode information. The receiver decodes this information by recognizing the subcarrier modulation rates. Using this technique, control signals can be overlaid on regular OFDM frames, and can be decoded even if the receiver can only receive a partial set of subcarriers.

1.5.4 Spectrum Aggregation

Spectrum agility requires support from both the infrastructure and the end-user devices. In WiFi networks, the client must be able to aggregate bandwidth from multiple APs so that any transmission opportunities can be efficiently exploited. However, there are two significant obstacles that must be overcome.

First, current client devices are fixed-function, monolithic spectrum devices that can only communicate with only one AP at a time. Hence, clients must associate with an AP before it can determine the channel quality to that AP. However, in the interest of optimality, it should only connect to APs that can provide it with the best transmission opportunity.

Second, this associate-then-measure approach is complicated by the fact that wireless channel statistics are time varying. Hence, the associate-then-measure approach cannot occur sufficiently quickly enough for the client to track the changing bandwidth and build an efficient aggregation schedule.

Our Contribution

We address these issues with Sidekick— a protocol that obtains channel state from multiple APs concurrently using Aileron.

Communication over partially-overlapping channels. Sidekick can communicate with APs even if their spectrum only overlaps partially. This avoids the need for a client to switch to a different channel and associate with an AP before exchanging channel state information.

Accurate tracking of time-varying channel state. We design a simple control protocol based on Aileronthat will enable Sidekickto accurately track the channel of multiple APs concurrently.

PHY coordination and spectrum agility are key properties in next-generation wireless networks. This dissertation will provide a clear understanding of the fundamental components needed to build next-generation networks and to integrate such networks into existing legacy systems.

Chapter 2

Per-Frame Spectrum Shaping

2.1 Introduction

Dynamic spectrum access (DSA), or spectrum agility, has become a popular solution to the problem of spectrum scarcity in wireless networks [15]. New devices that are designed to use only a monolithic block of spectrum can no longer expect to increase throughput by simply increasing their bandwidth. In fact, the throughput of an 802.11n device operating at 40MHz can even be *lower* than its throughput at 20MHz when encountering a 20MHz interference from another 802.11g or 802.11n device [16, 17]. Numerous other studies [18, 19] have reported performance anomalies when rate or bandwidth is blindly increased in an attempt to wrest more throughput from an overcrowded spectrum. We can only expect such problems to compound with the introduction of 802.11ac that supports up to 160MHz bandwidth. While this example deals with WiFi networks for clarity in exposition, the infeasibility of enhancing throughput by merely increasing bandwidth is also prevalent in non-WiFi networks. For example, a study of GSM usage patterns [20] shows that a wideband device cannot operate within the GSM band without some form of spectrum agility.

However, despite this obvious problem and the list of well-studied solutions, building efficient spectrum-agile devices is still a challenge for two main reasons. First, the current crop of commercial wireless devices are ill suited for DSA networks as they are primarily designed to use static, monolithic spectra. For example, spectrum- and bandwidth-agile platforms, such as SampleWidth [9] and FLUID [10], all have channel-switch times on the order of milliseconds. Second, the protocol stack does not fully support spectrum-agile communications. As an example, consider 802.11n OFDM frames that are detected by exploiting the self-correlation property of the preamble. This approach fails if the preamble is spread out over a non-contiguous spectrum, or in the face of interference from narrower band devices. Non-contiguous OFDM (NC-OFDM) techniques can be applied, but synchronization can be performed if and only if the set of non-contiguous subcarriers is known at the receiver beforehand.

We argue that the key capability that is missing from current state-of-the-art radio hardware is per-frame spectrum shaping. This is an important functional primitive that allows a radio to adapt to challenging channel conditions at the smallest practical unit of transmission.

2.1.1 Why Per-Frame Spectrum Shaping?

WiFi Channels. 802.11 devices are known to suffer significant performance degradation due to narrowband interference [21]. The effects of narrowband interference include timing recovery failure, the automatic gain control (AGC) failure due to an unexpected introduction of interference energy, and Physical Layer Convergence Protocol (PLCP) header processing failure.

Rapid frequency hopping (FH) by an 802.11 device [21] has been shown to improve its performance in the presence of narrowband interference. However, FH cannot avoid interference from a FH interferer, such as Bluetooth, if the hopping sequences of the WiFi and the interferer are not properly synchronized. Furthermore, collisions between multiple FH devices using different hopping sequences is a well-known challenge when scaling FH to a larger network [22].

This disadvantage of FH comes from the fact that it switches channels blindly, even when there is no interference on the channel it is currently using. This increases the possibility of the FH itself interfering with devices on other channels. We posit that a reactive approach to interference avoidance using per-frame spectrum shaping will enable 802.11 devices to avoid narrowband interference while maintaining high throughput and manageability. The use of per-frame spectrum shaping effectively re-allocates the spectrum of a transmission dynamically only when interference is detected on the channel. This minimizes the amount of spectrum touched by an 802.11 device and avoids the unnecessary channel-switch overhead when no interference is detected.

Non-WiFi Channels. Devices operating in non-WiFi channels have to contend with severe spectrum fragmentation due to multiple narrowband interferers. We illustrate this using spectrum traces [23] that took measurements from a 1.5GHz band and is centered at 770MHz frequency. This trace set thus covers multiple GSM and TV channels.

Fig. 2.1 shows the availability and outage durations of 1, 5 and 20MHz monolithic channels operating within this band. Consider, in particular, the 20MHz transmission that is typical of WiFi devices. At a first glance, the long median channel-availability duration of 3s can easily accommodate the channel-switch time of typical WiFi devices. However, we observe

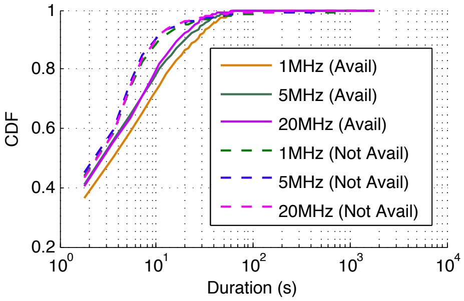


Figure 2.1: CDF of the channel busy and available durations.

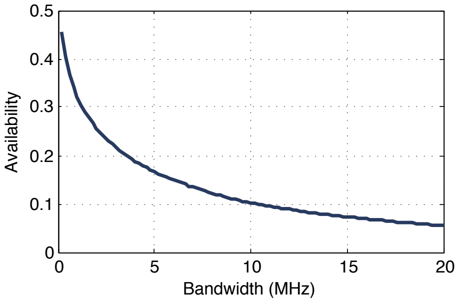


Figure 2.2: Channel availability of different transmission bandwidths.

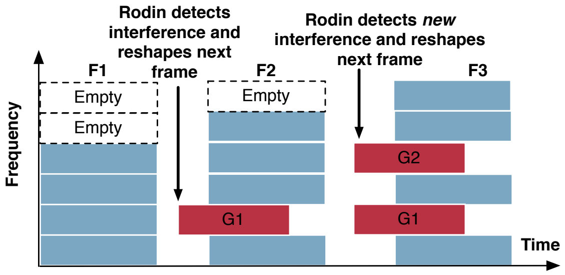

Rodin detects interference and reshapes next frame
Rodin detects *new* interference and reshapes next frame
F1
F2
F3
Empty
Empty
Empty
G1
G2
G1
Frequency
Time

Figure 2.3: Transmission of 3 frames F_1, F_2 and F_3 using Rodin. Rodin reshapes the spectrum of F_2 and F_3 to avoid interference from G_1 and G_2, respectively.

from Fig. 2.2 that monolithic 20MHz channels can transmit only about 6% of the time. This low availability is due to the presence of multiple uncoordinated narrow bandwidth interferers. Hence, in order to sustain a 20MHz transmission, multiple discontiguous 1MHz (or narrower) channels have to be bonded together. Given that the correlation between the different channels is low [20], such a device can expect to continuously reconfigure its set of bonded channels to avoid primary user interference. The otherwise long outage duration that it faces, as shown in Fig. 2.1, will severely degrade the quality of service. The ability to perform per-frame spectrum shaping is thus key for operating in non-WiFi channels as well.

2.1.2 The Limitation of SDRs

Software-defined radios (SDRs) have been used to develop the flexible RF interfaces required for DSA devices. However, SDR platforms face problems arising from poor efficiency and high complexity. SDR platforms, such as USRP [24] and SORA [25], are limited by the efficiency of a general-purpose platform in multitasking real-time DSP with other system tasks, while FPGA-based SDR platforms, such as WARP [26], are complex to work with. This complexity and inefficiency poses a significant challenge because it is necessary to re-implement the entire MAC/PHY protocol on the SDR platform in order to reap the advantage of PHY-layer flexibility.

2.1.3 The Limitation of COTS Devices

A commercial off-the-shelf (COTS) device that has its RF frontend separated from the MAC baseband chipset can facilitate easy integration between the SDR and COTS. However, COTS devices are increasingly implemented as single-chip solutions to improve power and space efficiency. This limits the flexibility of the RF frontends of COTS devices in supporting the various spectrum management policies required for per-frame spectrum shaping.

2.1.4 The Challenge

We take a very different approach to DSA and address an important question: *"What is a simple practical extension to current wireless devices that makes them spectrum agile?"* We stress that any solution must be *general* enough to apply to the majority of COTS wireless devices currently available, yet *simple* enough to minimize the additional overhead that are added to COTS devices.

The intuition behind this comes from the fact that neither COTS devices nor SDRs are individually capable of supporting the per-frame spectrum shaping necessary for DSA. Hence, a hybrid platform built using both SDRs and COTS devices is necessary. The SDR handles only the necessary PHY-layer manipulations, while the COTS device handles the main MAC/PHY processing. A practical DSA extension must have the following three important properties.

Property 1: Protocol independence. It must support as many current wireless protocols as possible. Hence, a COTS device should only have to be "plugged into" a DSA extension platform to gain spectrum agility. In reality, some modifications to the COTS platform may be necessary, but such changes must be minimal. Easy deployability of a DSA extension platform will naturally maximize the chance of its widespread acceptance. With this property, Rodin can be easily integrated into both OFDM and non-OFDM COTS devices.

Property 2: Per-frame spectrum shaping. Per-frame spectrum shaping is a general spectrum-shaping primitive that can be used to construct other spectrum-management protocols. In the absence of detailed knowledge about the behavior of other devices in the ISM or whitespace bands, a DSA platform must be able to adjust its spectral use on a frame-by-frame basis to react to unexpected transmissions by primary users.

Property 3: Fast spectrum agreement. Besides having the capability of per-frame spectrum shaping, the transmitter and receiver(s) must also agree on a common set of (possibly non-contiguous) spectrum bands before commencing transmission. Prior work on spectrum agreement made use of control channels [27], pre-defined backup channel lists [28], or centralized channel assignment [10]. Unfortunately, these approaches are too slow to meet the required delay bounds for per-frame spectrum shaping.

2.1.5 Rodin: Our Solution

We propose Rodin[1]—a hardware DSA extension to COTS devices. Rodin consists of three key components that enable it to serve as a drop-in DSA extension to arbitrary wireless devices.

Direct connection to COTS device. Rodin connects to a COTS device directly through the antenna port(s) on the COTS radio, thus upgrading unmodified COTS devices with spectrum agility.

Fast FPGA-based spectrum shaping. Rodin can split the spectrum of an unmodified signal from the COTS device into multiple non-contiguous spectrum subbands; the individual subbands are transmitted on unoccupied portions of the spectrum to avoid interference from other narrowband transmitters. Rodin does not decode the signals to and from the COTS device. Our hardware implementation achieves this spectrum subdivision of each frame

[1] Named after Auguste Rodin, the French sculptor.

within $2\mu s$ of detecting a passband signal from the COTS device.

Novel preamble design for spectrum agreement. A Rodin transmitter uses a novel preamble design to notify a Rodin receiver of the spectrum occupied by the accompanying spectrally-reshaped frame. With this preamble, Rodin eliminates the need for a separate control channel, backup channel lists or a centralized spectrum coordinator. This preamble, when combined with fast spectrum shaping, enables Rodin to rapidly adapt to any primary transmission pattern seen on channels.

To see how efficiently this can be done, consider shaping a 20MHz 802.11n frame over multiple 5MHz subbands. Spectrum agreement and shaping can be achieved in under $10\mu s$. This adds only 3.8% of additional overhead to the transmission time of an 802.11n frame without aggregation. The overhead will be even lower if frame aggregation is used. The negligible overhead enables Rodin to react to rapidly changing channel conditions on all types of channels.

Rodin is a novel RF frontend for COTS devices for cognitive spectrum management. In the short term, it extends the experimental capabilities of COTS devices but it can also be built into COTS devices to achieve integrated SDR-COTS hybrids in the future.

Our contributions in this chapter are: (a) a detailed design of spectrum shaping and agreement in Rodin, (b) an evaluation of the real-world performance of Rodin via controlled experiments with FPGA-based implementations, and (c) an analysis of the performance of Rodin using detailed channel measurements.

2.2 Overview of Rodin

Rodin is a general-purpose per-frame spectrum-sculpting platform designed for wideband frame-based COTS devices. In particular,

- Rodin is designed for wideband COTS devices that share the spectrum with other devices of narrower bandwidth. Examples of such scenarios include 160MHz 802.11ac or 40MHz 802.11n devices that share the same 5GHz band with 802.11a devices operating at 20MHz; UWB devices that share the spectrum with narrowband cellular networks.
- Rodin assumes that the maximum bandwidth of its SDR RF frontend is greater than the bandwidth of the transmitted COTS signal. Rodin shapes the spectrum of each frame while keeping the overall transmission bandwidth constant. Note that Rodin does not change the operating bandwidth of the COTS device.
- Rodin is designed for CSMA networks with multiple concurrent asynchronous transmitters that occupy non-overlapping spectra. This maximizes the frequency reuse of

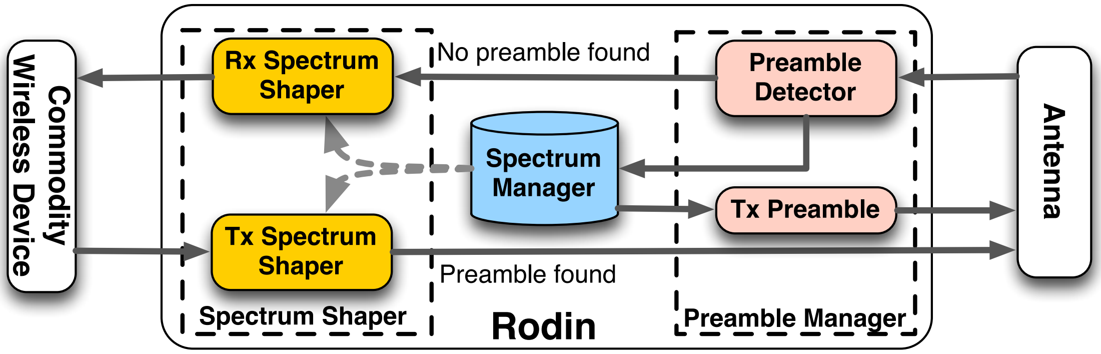


Figure 2.4: High-level architecture of Rodin.

wireless channels. However, these channels are not perfectly orthogonal to each other due to non-ideal pulse shaping filters [29].

Rodin has three key features to function as a general per-frame spectrum-shaping platform for COTS devices: (a) capability for direct connection to the COTS device, (b) FPGA-based spectrum shaping, and (c) a novel preamble design for fast spectrum agreement.

Rodin divides its total RF bandwidth B into N subbands and shapes the spectrum of a frame that occupies N_F ($< N$) of these subbands. Fig. 2.3 shows an example of Rodin reshaping a wideband transmission, with $N = 6$ and $N_F = 4$, in the face of narrowband interference. Frame F_1 can be transmitted without any additional shaping since no interfering transmission is present. However, almost immediately after transmitting F_1, Rodin detects a narrowband interference G_1 that occupies one subband. It maps the spectrum of F_2 into the remaining subbands and transmits it without interfering with G_1. This frame-by-frame spectrum reshaping is repeated for F_3 to avoid interference from G_2.

If per-frame spectrum shaping is not used, a wideband transmission would be blocked by a narrowband transmission, or a wideband transmission collides with a narrowband transmission if the narrowband transmitter does not correctly detect the wideband transmission.

These features are realized with the system architecture shown in Fig. 3.1. The **Spectrum Shaper** reshapes the signal to and from the COTS wireless device in real time, while the **Preamble Manager**, consisting of a preamble detector and a preamble constructor, uses specially-constructed preambles to exchange spectrum information between Rodin devices. The **Spectrum Manager** executes a protocol that selects the best set of spectrum bands for a particular transmitter–receiver pair.

These components are detailed in the rest of this chapter. For simplicity, our current design of Rodin is limited to SISO devices only, although an extension to MIMO devices is straightforward.

2.3 Spectrum Shaping in Rodin

Spectrum shaping divides the spectrum occupied by a COTS device into multiple discontiguous frequency bands. In order to realize real-time spectrum shaping, (a) the spectrum-shaping procedure must have low latency and (b) the spectrum shapers on the transmitter and the receiver must cooperate with minimal synchronization. Property (a) relates to the efficiency of the spectrum shaper — upon specification of the desired subbands, the shaper must quickly reshape the spectrum with minimal delay. In contrast, property (b) relates to the tolerance of the spectrum shaper to errors caused by channel distortion, timing, frequency shifts, etc. This is particularly important since different PHY protocols engage different measures to combat distortions. For example, DSSS-based protocols use Rake receivers and equalizers while OFDM-based protocols use the Schmidl-Cox algorithm. Obviously, it is not feasible for Rodin to support the wide variety of synchronization primitives to achieve protocol independence. Hence, Rodin focuses on spectrum shaping while leaving protocol-specific DSP functions (such as pilot handling) to the COTS device.

In the rest of this section, we only describe a two-band shaping process ($N > N_F = 2$) for the sake of clarity. This process can be easily extended to multi-band shaping.

2.3.1 Overview of Spectrum Shaping

Let $X(f)$ denote the original spectrum of the frame received by Rodin from the attached wireless device. The spectrum-shaping procedure for the frame *transmission* consists of the following components.

(a) Pre-filter modulation. Rodin only uses low-pass filters for spectrum shaping. Hence, the input signal $X(f)$ must be modulated to align the relevant portion of $X(f)$ with the passband of the filter $H(f)$. Let $m_1^{(a)}(t) = \exp\{j2\pi k_1 Bt/N\}$ and $m_2^{(a)}(t) = \exp\{j2\pi k_2 Bt/N\}$ be the time-domain complex-valued carrier used to modulate $X(f)$, with $k_i = 0, \ldots, N-1, \forall i = 1, 2$. The modulated spectrum is:

$$\begin{aligned} X_i^{(a)}(f) &= X(f) * \delta(f - k_i B/N) \\ &= X(f - k_i B/N), \ \forall i = 1, 2 \end{aligned} \tag{2.1}$$

where $\delta(\cdot)$ is the Dirac delta function.

(b) Filtering. Once the spectrum of the input signal has been appropriately modulated, a low-pass filter is applied to split the input spectrum into two separate subbands. Let $H_1(f)$ and $H_2(f)$ be the two low-pass filters used in this example. The two spectral subbands

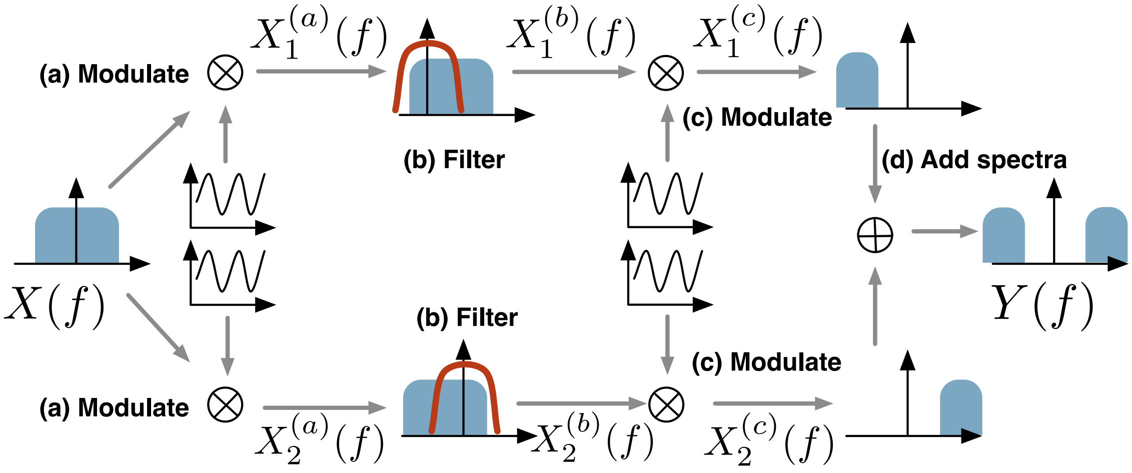


Figure 2.5: Shaping a frame occupying a contiguous spectrum $X(f)$ into two separate spectrum bands $Y(f)$. The shaping procedure is a 4-step process, labeled (a)-(d).

$X_1^{(b)}(f)$ and $X_2^{(b)}(f)$ are:

$$\begin{aligned} X_i^{(b)}(f) &= H_i(f)X_i^{(a)}(f) \\ &= H_i(f)X(f - k_iB/N),\ \forall i = 1,2 \end{aligned} \tag{2.2}$$

(c) Post-filter modulation. Each filtered subband must be transmitted at a frequency that encounters minimum interference. This modulation step uses $m_1^{(c)}(t) = \exp\{j2\pi l_1 Bt/N\}$ and $m_2^{(c)}(t) = \exp\{j2\pi l_2 Bt/N\}$ as the modulating carrier, where $l_1, l_2 = 1, \ldots, N$. The second modulation step achieves, $\forall i = 1,2$:

$$\begin{aligned} X_i^{(c)}(f) &= X_i^{(b)}(f) * \delta(f - l_iB/N) = X_i^{(b)}(f - l_iB/N) \\ &= H_i(f - l_iB/N)X(f - (l_i + k_i)B/N) \end{aligned} \tag{2.3}$$

(d) Combining spectra. Finally, the two subbands are added to produce a single spectrally non-contiguous frame. This results in a single time-domain data stream that is sent to the radio frontend of Rodin to be transmitted:

$$Y(f) = X_1^{(c)}(f) + X_2^{(c)}(f). \tag{2.4}$$

The Rodin *receiver* executes the same process as shown in Fig. 2.5 using the same low-pass filters but with the modulation sinusoids rearranged as:

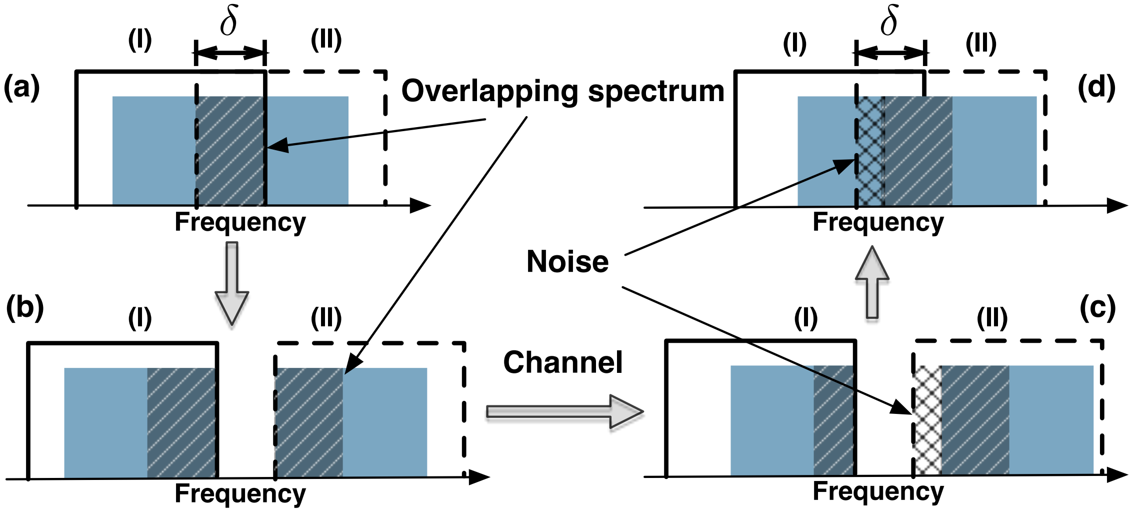


Figure 2.6: Spectrum shaping using two partially-overlapping filters. (a) Two subbands share an overlapping band δ. (b) After post-filter modulation, each subband contains a copy of the overlapping spectrum δ. (c) As a result of frequency drift at the receiver, only a portion of one subband is recovered while the other subband is recovered along with a noise band. (d) The overlapping spectrum δ ensures that the original spectrum can be reconstructed even if one subband is not recovered completely.

$$\begin{aligned} X(f) &= \hat{Y}(f), Y(f) = \hat{X}(f), \\ m_1^{(a)}(t) &= 1/m_1^{(c)}(t), m_2^{(a)}(t) = 1/m_1^{(c)}(t), \\ m_1^{(c)}(t) &= 1/m_1^{(a)}(t), m_2^{(c)}(t) = 1/m_1^{(a)}(t) \end{aligned}$$

where $\hat{Y}(f)$ is the spectrum of the received frame and $\hat{X}(f)$ is the spectrum of the reconstructed frame.

2.3.2 Filter Design for Spectrum Shaping

Prior work in spectrum shaping has largely adopted an OFDM-based approach [30, 31, 32]. While this approach draws upon many readily understood concepts similar to typical OFDM(A) modulation schemes, it has two significant disadvantages when applied to real-time spectrum shaping: (a) high overhead and complexity involved in maintaining strict time and frequency synchronization with pilot subcarriers, and (b) reduction in throughput due to the necessary use of a cyclic prefix to guard against inter-symbol interference.

Rodin mitigates these disadvantages with partially-overlapping finite-impulse response (FIR) spectrum shaping filters. Note that these FIR filters are only used for spectrum shaping. Rodin can support both OFDM and non-OFDM protocols using these FIR filters. Rodin itself is tolerant of timing drifts as time synchronization is handled by the attached COTS device as part of its PHY protocol; as long as the filtered spectrum encompasses the received frame, the COTS device can determine the appropriate frame boundary. Rodin is also resilient to frequency drifts by transmitting redundant spectral information through the use of partially-overlapping filters.

To understand this, consider the use of partially-overlapping filters to shape an input frame, as illustrated in Fig. 2.6. The two filters divide the spectrum into two portions, (I) and (II), that share a common overlapping subband of bandwidth δ, as shown in Figs. 2.6(a) and (b). A frequency shift at the receiver, as shown in Fig. 2.6(c), causes some spectrum to be lost from (I) and noise to be introduced into (II). Observe that when the two subbands are recombined, the spectral information missing from (I) can be recovered from its redundant copy in (II). The degree of resilience to frequency drift is governed by the overlapping bandwidth δ, which is a configuration parameter. We must ensure that the value chosen for δ is greater than the expected frequency drift. The lower bound on the overlapping bandwidth thus depends on the quality of the COTS device that Rodin is connected to. The effect of this noise is minimal since it is located at the very edge of the shaping filter and thus will be more heavily attenuated. Furthermore, this noise subband is typically very narrow as real-world measurements of actual frequency drift are shown to be small [33].

The overlapping bandwidth is also lower bounded by the amount of resources available on the FPGA: longer filters, which allow smaller overlapping bandwidths, require larger numbers of FPGA slices. The WARP platform used for our Rodin prototype can support a 64-tap filter.

The ideal requirements for a spectrum shaping filter are: (a) constant unit amplitude response and linear phase response in the passband, (b) narrow transition bandwidth, and (c) very high attenuation in the stopband. Unfortunately, neither the typical windowed-approach nor the Parks-McClellan algorithm can produce a filter that satisfactorily meets these three constraints. Thus, we adopt a *constrained least squares* algorithm [34] for filter design. We design our filters, using this algorithm, to have 64 taps, a passband ripple of 0.1dB and an overlapping spectrum bandwidth that is approximately 10% of the total filter bandwidth.

2.3.3 Spectrum-Shaping Latency

We have implemented the spectrum shaper using a 64-tap FIR filter on the FPGA of the WARP platform to both validate its functionality and study the latency incurred in real-time spectrum shaping. The FPGA on the WARP runs at 40MHz.

The modulation and spectral combination steps consists of time-domain multiplication and addition, respectively. Each step thus incurs a latency of 1 clock cycle. The filtering step consists of a 64-tap time-domain convolution, and incurs a latency of 64 cycles. Note that the filtering latency is *independent* of the number of subbands used since all filters run in parallel on the FPGA.

The total latency of real-time spectrum shaping is therefore $64+1+1 = 66$ cycles, or $1.65\mu s$ when running on the 40MHz FPGA. This spectrum-shaping latency is a mere 0.7% of the transmission time of a 1.5KB 802.11n frame sent at 54Mbps (Rodin currently only supports SISO). Hence, a real-time spectrum shaping extension to commodity wireless hardware is feasible.

2.4 Preamble for Spectrum Agreement

Rodin uses a unique preamble that is designed to indicate both the start of a frame as well as the spectrum bands it occupies.

2.4.1 Challenges to Spectrum Agreement

A frame sent by the transmitter can be decoded if and only if the spectrum occupied by the frame is known by the receiver. If the spectrum occupancy of a frame is unknown, the receiver can attempt to search for the frame over all the subbands. Assuming that a frame is known to occupy M out of N subbands, the receiver has to attempt to search for the frame over $N!/(M!(N-M)!)$ possible subband combinations; if the bandwidth of the frame is unknown, this search space increases to $\sum_{m=1}^{M} N!/(m!(N-m)!)$ subband combinations.

One might think of applying energy sensing to the subbands and decoding a frame using only the subbands with signal energy above a given threshold. This method, though simple, suffers from two serious limitations: (a) frequency-selective fading on the subband may result in a missed detection, and (b) in the case of multiple concurrent transmissions, each using a different set of subbands, it is impossible for a receiver to correctly map each occupied subband to its transmitter based on energy detection alone.

2.4.2 I-FOP Design

Rodin addresses this predicament by prepending a multi-subband preamble, I-FOP (In-Front Of Preamble), to the transmitted COTS frame. A unique preamble is assigned to each *flow* within the network, where a flow is simply a group of consecutive frames sent by the COTS device. This preamble must therefore be designed to (a) assign an address to each unique flow within the network, (b) specify the subband occupancy of each transmitted frame, and (c) enable the receiver to recover both the address and subband occupancy information of each frame without prior coordination with the transmitter. We stress that the spectrum occupancy can change from frame to frame even within the same flow.

A key feature that the preamble must possess is a strong correlation property — a receiver searching for a preamble P via correlations must encounter a large correlation peak if and only if P is present on the channel. Furthermore, this auto-correlation property must hold for a large set of sequences of the same length. This allows a different preamble to be assigned to each flow within a collision domain.

Zadoff-Chu (ZC) sequences [35] meet our requirements and are thus used in I-FOP. The length-L discrete ZC sequence is:

$$x_u[n] = \exp\left(-j\frac{\pi u n(n+1)}{L}\right) \tag{2.5}$$

where u is the sequence ID and $0 \leq n, u \leq L-1$. ZC sequences have strong correlation properties that make them ideal for I-FOP: (a) the auto-correlation of a length-L ZC sequence with a cyclically-shifted version of itself is zero if L is prime; (b) the cross correlation between two prime length ZC sequences is $1/\sqrt{L}$.

Rodin selects a set $\{p_0, \ldots, p_{N_F-1}\}$ of ZC sequences to address a flow. The bandwidth of each frame within the flow occupies N_F subbands. Rodin applies a random cyclic shift to each sequence before constructing the preamble for the flow. The cross-correlation property reduces the chance of collision in the event that the same ZC sequence is selected by multiple transmitters. With this approach, there is a large set of L^2 ZC sequences of length-L that can be used to construct preambles.

Let $\mathbf{f} = \{f_0, \ldots, f_{N_F-1}\}$ be the set of N_F subbands that Rodin uses to transmit a frame. The preamble constructed for this particular frame is specified by the set $\mathbf{S} = \{S_{f_k}^{p_k} : 0 \leq k \leq N_F - 1\}$, where $S_{f_k}^{p_k}$ indicates that sequence p_k is transmitted on the subband f_k and

$f_0 \leq \ldots \leq f_{N_F-1}$. The time-domain representation of the preamble is:

$$y[n] = \sum_{k=0}^{N_F-1} x_{p_k}[n] \cdot e^{-j2\pi f_k n/N} \tag{2.6}$$

for $0 \leq n \leq L-1$.

2.4.3 I-FOP Detection

We assume, for now, that the transmitter and the receiver know the set of ZC sequences, $\{p_0, \ldots, p_{N_F-1}\}$, used to address the flow between them. The receiver faces the challenge of determining the set of subbands $\{f_0, \ldots, f_{N_F-1}\}$ occupied by the transmitted frame.

Let $\hat{\mathbf{S}} = \{\hat{S}_{f_k}^{p_k} : 0 \leq k \leq N_F - 1\}$ be the preamble that is detected by the receiver. This preamble detection procedure uses the following two properties of the transmitted preamble.
(a) The known order of the sequences. Given the set of ZC sequences, $\{p_0, \ldots, p_{N_F-1}\}$, used in the preamble, $\hat{\mathbf{S}}$ must be found such that $f_0 < f_1 < \ldots < f_{N_F-1}$. This increases the number of possible preambles by allowing for different preambles to be constructed using the same set of ZC sequences, but with different subband orders.
(b) Location of the correlation peaks. Multiple ZC sequences sent by the same transmitter as part of a single preamble will arrive at the receiver at approximately the same time. However, due to frequency-selective fading, the peaks may not be precisely aligned in time. To account for this, we use a threshold, ξ, to limit the range of acceptable separation between peaks—only sets of correlation peaks that are within ξ samples apart are considered as candidates for the preamble.

Algorithm 1 shows the pseudocode of the multi-preamble detection. In lines 1–1, Rodin searches for the ZC sequence that is transmitted in each subband. Observe that we use **parallel-for** loops for this search step since in an FPGA implementation, all iterations of these **parallel-for** loops can be executed concurrently to reduce the search time. In lines 1–1, Rodin searches for a set of subbands $\{f_0, \ldots, f_{N_F-1}\}$ that contain the sequences $\{p_0, \ldots, p_{N_F-1}\}$ such that $f_0 < \ldots < f_{N_F-1}$ must hold. Note that this **for** loop cannot be parallelized since the result of each iteration depends on the result of the previous iteration.

2.4.4 Inter-Subband Interference

Observe that Rodin does not apply any filter to isolate each subband before conducting a search for a ZC sequence. This choice is made to avoid the additional delay that comes with a filtering step. However, there is now a possibility that sequences on different subbands

Algorithm 1: I-FOP detection.

```
Input  : Set of ZC sequences $\mathbf{P} = \{p_0, \ldots, p_{N_F-1}\}$ RF sampling data stream, $\hat{y}[n]$,
         Correlation threshold, $\gamma$
Output: Occupied subbands $\mathbf{f} = \{f_0, \ldots, f_{N_F-1}\}$
parallel-for $k \in 0, \ldots, N-1$ do
    /* Shift subband $f_k$ to baseband                                    */
    $w_k[n] \leftarrow \hat{y}[n] \cdot e^{j2\pi f_k n/N}$;
    parallel-for $l \in 0, \ldots, N_F - 1$ do
        /* Correlate with $p_l$                                           */
        $\rho_{k,l}[n] \leftarrow (w_k \star p_l)[n]$;
        $\lambda_{k,l} = \max_{0 \leq m \leq \xi} \rho_{k,l}[n-m]$;
    end-parallel-for
    /* Determine the ZC sequence on subband $k$                           */
    $\sigma_k \leftarrow \arg\max_{0 \leq l \leq (N_F-1)} \lambda_{k,l}$;
    $\eta_k \leftarrow \max_{0 \leq l \leq (N_F-1)} \lambda_{k,l}$;
end-parallel-for
$l \leftarrow 0$;
for $k \in 0, \ldots, N-1$ do
    $f_l \leftarrow \infty$;
    if $\sigma_k = p_l$ and $\eta_k > \gamma$ then
        $f_l \leftarrow k$;
        $l \leftarrow l + 1$;
    end
    if $l = N_F$ then
        return $\mathbf{f} = \{f_0, \ldots, f_{N_F-1}\}$;
    end
end
return $\mathbf{f} \leftarrow NULL$;
```

		Preamble Length		
		37	73	113
BW (MHz)	5	$7.4\mu s$	$14.6\mu s$	$22.6\mu s$
	10	$3.7\mu s$	$7.3\mu s$	$11.3\mu s$
	20	$1.8\,us$	$3.65\mu s$	$5.56\mu s$

Table 2.1: Time required for preambles constructed with ZC of length 37, 73 and 113 to be transmitted at 5, 10 and 20MHz bandwidths.

interfere with each other during the correlation-based search. This possibility is present regardless of the type of sequence used, e.g., Gold, ZC, Walsh-Hadamard, etc. However, we argue that the possibility of inter-subband collisions in our preamble design is very low.

A collision between two subbands can occur only if two or more different transmitters (a) select the same ZC sequence, (b) apply the same cyclic shift to the sequence, and (c) transmit at almost the same time. We posit that the probability of all three events occurring at even two non-colluding transmitters is very low. To gain some insight into this, first recall that in CSMA networks, the random backoff process undertaken by each transmitter minimizes the possibility of simultaneous transmissions. Even if simultaneous transmissions do occur, the set of ZC sequences can be made large enough to minimize the probability of collisions. For example, if we use ZC sequences of length 73, there are a total of $73 \times 73 = 5329$ possible sequences that can be used by Rodin. The probability of two devices picking the same sequence is a mere $(1/5329)^2 = 3.5 \times 10^{-8}$. Hence, inter-subband interference does not affect the performance of I-FOP.

2.4.5 I-FOP Delay

The spectrum-shaping delay incurred by I-FOP depends on two parameters: the length of the chosen ZC sequence, and the bandwidth at which each sequence is transmitted. Table 2.1 shows the transmission time required for each sequence built from ZC codes of 37, 73 and 113 samples long at 5, 10 and 20MHz. These subband bandwidths are suitable for use by 802.11 devices. The bandwidth of each transmitted sequence $S_{f_k}^{p_k}$ must be no larger than the bandwidth of each subband.

The delay at the receiver is due mainly to the processing time needed to find I-FOP. For every new sample, $\hat{y}[n]$, received by the detector in Algorithm 1, the **parallel-for** loops operate in constant $O(1)$ time while the search in lines 1-1 takes $O(N)$ time. With sufficient FPGA resources for full parallelism, the search can be completed in N clock cycles, or $(0.0225N)\mu s$ with a 40MHz FPGA.

As an example, if we spectrally shape a 20MHz 802.11n over a $B =$ 40MHz RF bandwidth using the 64-tap filter from §2.3.3 and a preamble based on a length-37 ZC sequence, the overall delay is $1.65 + 7.4 = 9.05\mu s$. This is merely 3.8% of the transmission time of a 54Mbps 802.11n frame. The delay incurred by I-FOP may exceed the SIFS delay of WiFi COTS devices and trigger an ACK timeout at the transmitter. However, these ACK timeouts can be easily changed in software [36] and do not pose a hurdle to SDR-COTS integration. This local SIFS modification allows the attached COTS device to account for the extra delay from I-FOP ; other non-Rodin WiFi devices can operate normally without modifications.

2.4.6 Preamble Address Assignment

Rodin devices must assign an address to each flow in a distributed manner before spectrum agreement between devices is completed. Addresses to new flows are assigned using an *association frame*.

An association frame is a control frame sent between Rodin devices, and is not passed to the COTS device. Each association frame is spectrally shaped to occupy only the available subbands and is prepended with a preamble constructed using a fixed set of ZC sequences. This set of ZC sequences is the *association set* and is known to all Rodin devices. The association frame contains only the IDs of the ZC sequences and the order in which they will be used.

A Rodin receiver searches all subbands for the association set. Once this association set is found, Rodin recovers the association frame using the spectrum shaper from §2.3. It then decodes the frame to obtain the ZC sequence information that will be used for subsequent frames from the same flow. Once an address has been assigned, all transmissions belonging to the same flow, even if they originate from different Rodin devices (e.g., DATA and ACK frames), use the same preamble address.

Since the information carried in the association frame is small, the size of the frame is small, especially when compared with the total size of the flow. Hence, the overhead of address assignment is negligible.

2.4.7 Subband Selection

The transmitter selects the subbands by choosing the N_F subbands that have the lowest energy levels at the point of frame transmission. We make use of an FFT (Fast Fourier Transform)-based energy detector — we take the FFT of incoming samples and measure the magnitude of the energy in each subcarrier. On the 40MHz FPGA, for example, a 128-bin FFT takes approximately $5\mu s$. Hence, energy values at any point in time are delayed by about $5\mu s$. This is acceptable since the channel state does not vary significantly over that short duration. Note that energy sensing delay decreases as the FFT length gets shorter.

On a faster and larger FPGA, we can also implement more advanced spectrum-scanning techniques, such as those based on the Spectrum Correlation Function [37]. This will enable Rodin to not only detect the currently occupied subbands, but also determine the protocol occupying them and predict future usage patterns of the interferer.

2.5 Spectrum Management

Algorithm 2: Spectrum Manager.

```
while True do
    while No frame from COTS device detected do
        $\hat{y}[n] \leftarrow$ next sample from RF frontend;
        if Preamble detected at $\hat{y}[n]$ then
            Configure Rx Spectrum Shaper to span subbands of next frame;
        end
        Send $\hat{y}[n]$ to Rx Spectrum Shaper;
        Send output of Rx Spectrum Shaper to COTS wireless device;
    end
    while Frame from COTS device detected do
        Configure filters in Tx Spectrum Shaper to appropriate subbands, if
          necessary;
        Configure Tx Preamble to tag occupied subbands;
        Transmit preamble from Tx Preamble;
        $x[n] \leftarrow$ next sample from COTS device;
        Send $x[n]$ to Tx Spectrum Shaper;
        Send output of Tx Spectrum Shaper to RF frontend;
    end
end
```

Algorithm 2 shows the pseudocode that defines the operation of the Spectrum Manager. Rodin is in the receive state until frames are detected from the COTS device. In this state, the RX spectrum-shaping filters are configured to span the occupied spectrum indicated by each received I-FOP.

When a frame is transmitted by the COTS device, Rodin first configures the TX spectrum-shaping filters and TX I-FOP to span the transmit spectrum subbands. The preamble is then transmitted while the samples from the COTS device are filtered and modulated. The spectrally shaped samples are transmitted after I-FOP transmission is complete.

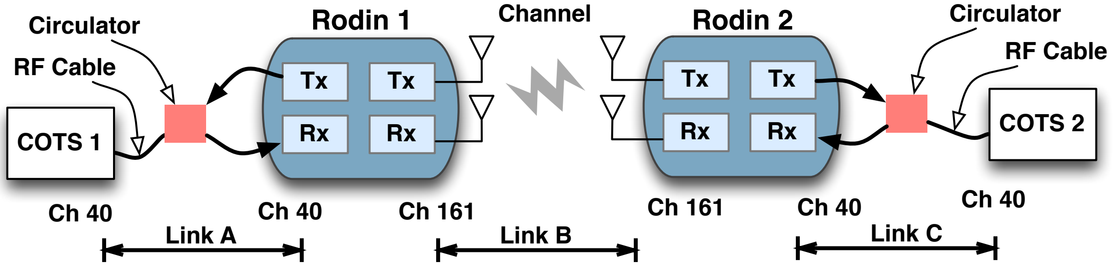


Figure 2.7: Experimental setup. Each Rodin device is connected to a COTS device via a coaxial cable.

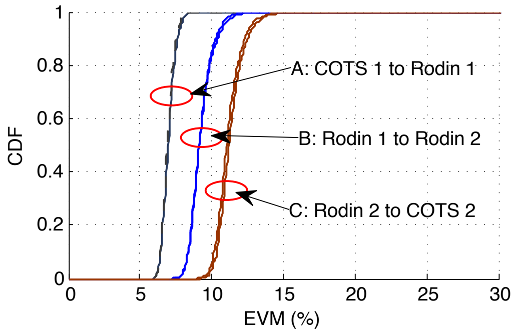


Figure 2.8: EVM of symbols in an OFDM frame with and without spectrum shaping. No interference.

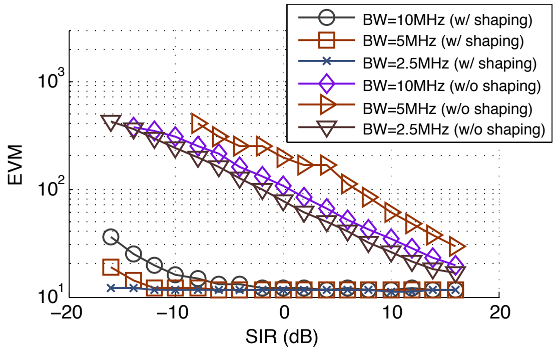


Figure 2.9: Mean EVM of OFDM frames measured at COTS 2 under different SIR levels.

2.6 Evaluation: Spectrum Shaping

2.6.1 Experiment Setup

Fig. 2.7 illustrates the setup used for evaluating the performance of individual Rodin devices. Each Rodin spectrum shaper is implemented in Verilog/VHDL and runs on the FPGA of a WARP platform with four radios. Each radio is permanently set to either the Tx or Rx mode. One pair of Tx/Rx radios from each WARP device is connected to a *circulator* that is then connected to a COTS device. These connections are made using coaxial cables. A circulator routes passband signals between the COTS device and the two radios on the WARP—analog signals coming from the COTS device is sent only to the Rx radio on the WARP, while signals from the Tx radio on the WARP is routed only to the COTS device. Signals between the Rx and Tx radios are blocked by the circulator.

The circulator is used here so that Rodin can receive frames from the COTS device without the Tx-Rx switching delay that will otherwise be incurred by the radio hardware if only one

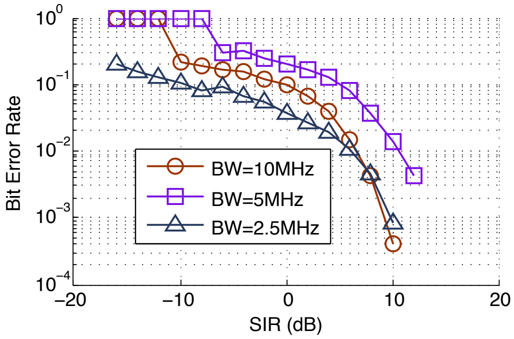


Figure 2.10: BER of OFDM frames measured at COTS 2 without shaping. No errors are encountered when spectrum shaping is used.

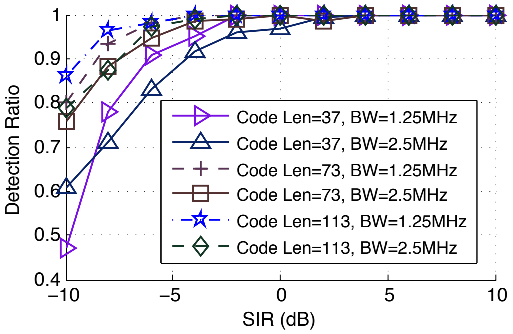


Figure 2.11: Preamble detection rate of three codeword lengths over $N = 8$ subbands on a 20MHz channel in the presence of interfering preambles. Each preamble is transmitted at 2.5MHz and 1.25MHz.

radio is connected to the COTS device. The other two Tx/Rx radios on each WARP device are connected directly to antennae. The two Rodin devices are placed approximately 2m apart. We have successfully used Ralink 802.11a WiFi card for COTS 1 and 2. However, to achieve finer-grained control of the transmitted signal for experimental purposes, we use WARP for COTS 1 and 2 for the rest of the experiments.

We send uncoded OFDM frames with a bandwidth of 10MHz between the two COTS devices. The spectrum of the OFDM frames can be shaped to span any 10MHz of spectrum within the 20MHz maximum bandwidth supported by each radio. For all experiments in this section, we split the 10MHz OFDM frame into two subbands of 5MHz each. These subbands are transmitted with a 10MHz separation between them.

Each Rodin device detects transmissions from its attached COTS device by checking the RSSI of the Rx radio that is directly connected to the circulator. If the RSSI exceeds a predefined threshold, the COTS device is assumed to be transmitting. This can be done easily as the SNR of transmissions over the coaxial cable is high. At all other times, the Tx radio continuously transmits received signals to the COTS device for receiver processing. This maintains the capability of the COTS device to overhear transmissions from other devices that share the same discontiguous spectrum.

We use two metrics to measure the performance of the spectrum shaper: *Error Vector Magnitude* (EVM), which is shown as a percentage, and *Bit Error Rate* (BER), which is the fraction of bits received in error.

2.6.2 Spectrum Shaping Results

Without Interference. We transmit 2,000 OFDM frames using QPSK symbols from COTS 1 to COTS 2 using the setup in Fig. 2.7, and measure the mean EVM of the frames between each pair of directly connected devices. This experiment is conducted twice, once with and once without spectrum shaping. Fig. 2.8 shows the CDF of measured EVM. One important conclusion from this result is: *Spectrum shaping does not distort the signal.* The CDF of the EVM over each OFDM frame is identical with and without spectrum shaping of the transmitted OFDM frame. Hence, real-time spectrum shaping can be implemented in the FPGA without any loss of signal quality.

Direct manipulation of a signal from a COTS device with an attached Rodin platform does introduce some distortion into the signal. The median EVM of frames sent over Link A of Fig. 2.7 is 7% while median EVM of the frame that is spectrally shaped and sent over Link B is 9%. Finally, the transmission over Link C to COTS 2 increases the median EVM to 11%. (An EVM of 11% is small enough not to increase BER; BER of all frames transmitted in Fig. 2.8 is zero.) These additional distortions are introduced during (a) up and down signal modulation by the AD/DA converters at both COTS devices and the radios on the WARP, and (b) time and frequency offsets between the COTS device and its attached WARP. Both of these sources of distortion can be eliminated by tighter integration between Rodin and the COTS device: distortion due to up/down converters can be reduced by passing the baseband signal directly between Rodin and the COTS device; distortion due to time and frequency offsets can be mitigated by synchronizing Rodin with the clock used by the COTS device.

With Interference. We transmit an interfering signal using another WARP device. The transmission power of this signal is varied to achieve a range of Signal-to-Interference Ratios (SIR). At each interference power level, we transmit the interference at three different bandwidths—2.5, 5 and 10MHz. Fig. 2.9 shows the EVM of a 10MHz OFDM frame sent from COTS 1 to COTS 2 that experiences interference with bandwidth 2.5, 5 and 10MHz. This experiment is conducted over a range of SIR levels, with and without Rodin spectrum shaping.

We first consider the performance of spectrum shaping. The mean EVM of the OFDM transmission when SIR is greater than -2dB is 11%. This is equivalent to a spectrum-shaped OFDM transmission in the absence of interference, as shown in Fig. 2.8. At SIR levels lower than -2dB, the impact of interference on the OFDM transmission depends heavily on the interference bandwidth — interference with a 10MHz bandwidth increases the EVM to almost 40% while it remains at 11% when the bandwidth is 2.5MHz. This variation is due to the fact that filters used to generate the interference signal are not ideal. Hence, some energy leakage occurs at the edges of the filter. Although the two subbands of the

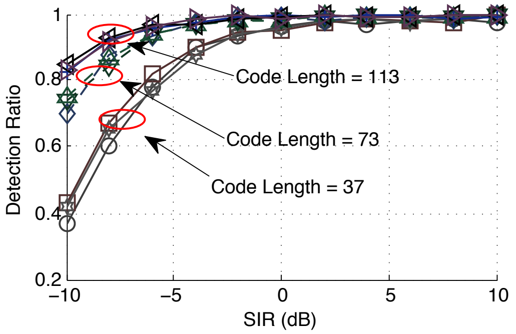


Figure 2.12: Preamble detection rate of three different codeword lengths over $N = 8$ subbands on a 20MHz channel. Each preamble is transmitted under 0, 12 and 20dB SNR.

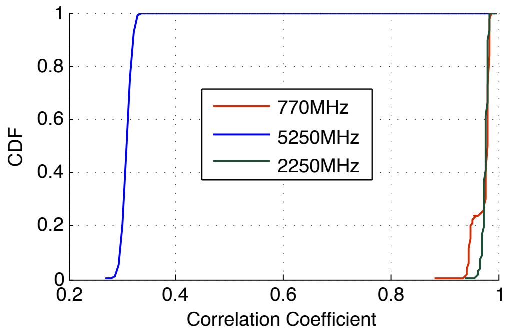


Figure 2.13: CDF of the correlation of the RSSI seen across all measurement slots over time.

spectrum-shaped OFDM frame are separated by 10MHz, they are still affected by the leaked interference energy. With a 10MHz interference bandwidth, the leakage energy is sufficient to distort the spectrum-shaped transmission. At 2.5MHz, the bandwidth of the interference is small enough that power leakage due to imperfect filters does not have a noticeable impact on the main OFDM transmission.

Without spectrum shaping, the narrowband interference has a significant impact on the OFDM transmission. For a given interference power, the smaller the interference bandwidth, the greater the interference power per subcarrier. The effect of this is seen from the fact that the distortion of the OFDM frames from the 5MHz interference is greater than that from the 10MHz frames—the increased interference power on fewer subcarriers is high enough to make up for the reduction in the number of subcarriers that encounter interference. When the interference bandwidth is at 2.5MHz, the small number of subcarriers affected allows the EVM to fall below that when a 10MHz interference is used.

This behavior is also evident when we consider the BER of the OFDM frames, as shown in Fig. 2.10. With spectrum shaping, the primary OFDM frames are sent on frequency bands that are not occupied by the interfering signal. The BER is thus zero for spectrum-shaped OFDM frames. Without spectrum shaping, the OFDM frame has a BER of 1.0 when it encounters a 10 or 5MHz interference at SIR below -12dB. The BER of the OFDM frame with a 2.5MHz interference is expectedly lower than that at interference bandwidths of 5 and 10MHz, but still stands at a high 1% at 8dB SIR.

2.7 Evaluation: I-FOP

In this section, we study the performance of I-FOP with two experiments: (a) under channels with varying SNR and SIR levels, and (b) in realistic multi-device contention scenarios.

2.7.1 SNR/SIR Performance

Experiment Setup. We evaluate I-FOP using five WARP devices placed at various locations around an office. Since the objective of this experiment is to evaluate the feasibility and performance of our preamble design, we run experiments using WARPLab+MATLAB instead of an FPGA-based WARP implementation. The results obtained using WARPLab and an FPGA implementation will be identical.

The performance of I-FOP is evaluated under SIRs ranging from -10 to 10dB. This interference consists of different I-FOPs that overlap with the transmission of the primary I-FOP. The result for each SIR is the mean of 2,000 preamble transmissions. In each transmission, we select a random receiver, transmitter and interferer from five WARP devices. We use a 20MHz channel with $N = 8$ subbands (each subband is thus 2.5MHz wide). Three different preamble lengths are evaluated: 37, 73 and 113 samples. For every preamble, we randomly select $N_F = 4$ subbands and transmit a different ZC sequence on each one. All ZC sequences are transmitted at the same bandwidth.

The receiver searches for the known ZC sequences that belong to the primary preamble transmission using the procedure shown in Algorithm 1. If the set of ZC sequences is found in the specified order, the preamble is considered to be detected. Otherwise, a missed-detection is recorded.

We also evaluate the performance of the preamble under varying SNR levels. However, due to the difficulty of accurately controlling the noise level in the channel, SNR evaluations are conducted using a simulated 802.11 channel.

Fig. 2.11 shows the detection probability of preambles with 3 different lengths, in the presence of overlapping interfering preambles. We run two experiments, with each one conducted over a range of SIR values. In the first experiment, each ZC sequence of every preamble (both the intended and interfering preambles) is sent at 2.5MHz (equal to the bandwidth of the subband); in the second experiment, each ZC sequence is sent at 1.25MHz, half the subband bandwidth. Interfering preambles are transmitted with a random time offset with respect to the non-interfering ones.

SIR Performance. Observe that for preambles with the same length, the detection accuracy is greater as the bandwidth of each ZC sequence is reduced for two reasons. First, as the sampling rate of WARP is constant, the longer correlation period that results from a lower bandwidth ZC sequence gives a higher correlation peak magnitude when a match is found. Second, when ZC sequences are transmitted at 1.25MHz, there is a guard band between sequences on adjacent subbands. This reduces the inter-subband interference that arises due to energy leakage from adjacent subbands. No guard bands are present when the ZC sequences are sent at 2.5MHz.

Also, observe that the detection ratio increases with increasing ZC sequence length. This is because the peak auto-correlation magnitude is proportional to the sequence length L, while the cross-correlation magnitude of $1/\sqrt{L}$ actually *decreases* with increasing sequence length. These two effects cause the SNR of the correlation peak to increase with increasing ZC sequence length.

SNR Performance. The accuracy of the preamble detector is similar over a wide range of SNR values, as shown in Fig. 2.12. For each ZC sequence length, we transmit the preamble at 0, 12 and 20dB SNR. Observe that accuracy is largely unaffected by the SNR level on the channel and is primarily dependent on the interference power.

In our experiments, the probability of detecting an I-FOP preamble when no I-FOP is present (false positive) is zero. False positives may occur due to ZC sequence collisions or more complicated channel fading scenarios. We can mitigate the effects of fading by using Rake correlators to search for the ZC sequences. However, false positives have limited impact on the operation of Rodin as the falsely received frame/signal are simply discarded by the COTS device.

2.7.2 Contention Performance

Experiment Setup. We use 16 WARP devices to demonstrate the accuracy of I-FOP under realistic channel-contention scenarios. For each experimental run, we use 16 devices that are non-uniformly distributed throughout an office. We randomly select four transmitters and four receivers, each using a 20MHz channel with $N = 8$ subbands. Each Tx-Rx pair uses a non-overlapping set of $N_F = 2$ subbands for communications. The four Tx-Rx pairs do not transmit simultaneously. Instead, a randomly selected jitter between 5 to $100\mu s$ is injected into each Tx-Rx pair in every experimental run. Note that this *injected* jitter is not equal to the *actual* transmit jitter due to the difficulty of synchronizing WARP devices perfectly. The actual jitter can differ from the injected jitter by up to $2\mu s$. We will show the aggregate results of 1000 such runs.

We demonstrate the accuracy of I-FOP in two ways. First, at each receiver, we show window of ξ samples within which the correlation peaks of the ZC sequences from the same transmitter are detected. The smaller the necessary ξ samples, the lower the rate of missed

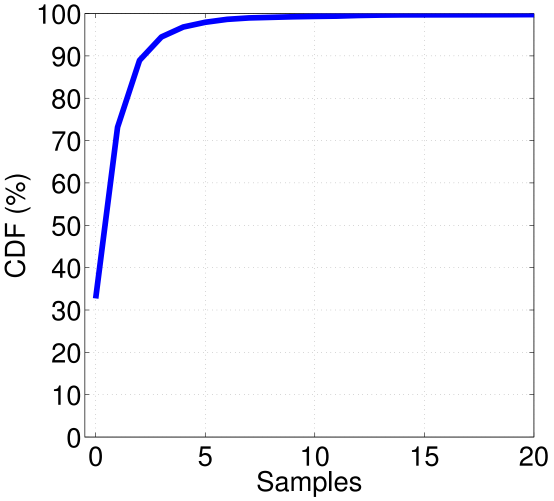


Figure 2.14: Difference between correlation peaks of ZC sequences from the same transmitter.

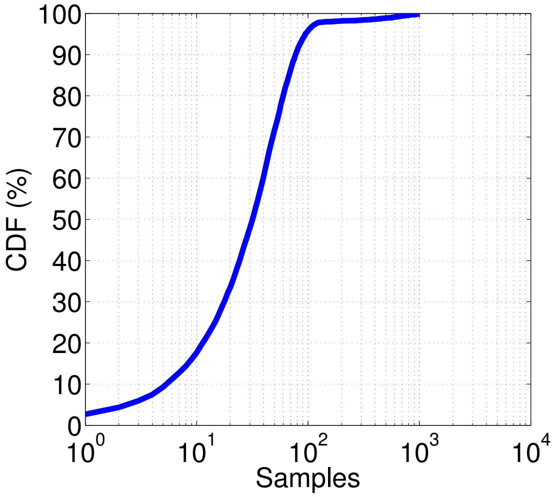


Figure 2.15: Position error of ZC sequences from different transmitters.

detections. Second, we show the accuracy at which each receiver can differentiate between preambles from different tranmitters. To do this, we search for all ZC sequences at every receiver, and compare the maximum separation between the received position of ZC sequences from different nodes to the injected jitter used in the transmission.

Correlation peaks from the same transmitter. Fig. 2.14 shows the CDF of the separation between correlation peak of ZC sequences from the same transmitter. In 1,000 experiments, over 99% of the correlation peaks of ZC sequences coming from the same transmitter are found within 5 samples ($0.125\mu s$) of each other. Furthermore, almost 100% of peaks were seen within 20 samples ($0.5\mu s$) of each other. Hence, by setting $\xi = 20$, we can use the location of correlation peaks to accurately detect almost all preambles.

Correlation peaks from different transmitters. Fig. 2.15 shows the CDF of the position error of ZC sequences from different transmitters. Observe that 99% of the ZC sequences are detected within 100 samples ($2.5\mu s$) of their transmission time. Note that this position error includes the possible difference between the actual and injected jitter from imperfect synchronization. However, this still provides strong evidence that I-FOP can successfully discriminate between transmitters if transmission times are separated by at least $2.5\mu s$.

2.8 Evaluation: Rodin

We evaluate the performance of Rodin using simulations over detailed channel measurements from [23]. These channel measurements show the usage behavior of devices that operate on three separate bands. During periods when the channel RSSI is low, primary user activity is absent and spectrum agile devices can transmit opportunistically. Our objective is to show the efficacy of per-frame spectrum shaping in using these short-term transmission opportunities.

2.8.1 Simulation Setup

Trace data. Each channel measurement of [23] spans a 1.6 GHz bandwidth that is centered at three different frequencies 770, 2250 and 5250 MHz, so they cover the 2.4 GHz and 5 GHz ISM bands used by WiFi devices. Measurements were taken over several days at three different locations: for brevity, we only show results using the data set measured at rooftop of a school. Each sweep over the entire 1.6GHz bandwidth takes about 1.8s and captures 8,192 samples, with each sample spanning 200kHz. Although the measurement data does not capture channel usage patterns shorter than 1.8s, channel statistics have been shown to remain unchanged at shorter time scales [20]. This strongly suggests that we can expect such statistics to be present at sufficiently small time scales to make Rodin useful. Hence, our analysis using this data is still applicable even when considering finer-grained channel usage patterns.

Device models. We model three different types of wireless devices in our simulations; two that support spectrum shaping and one that does not. The maximum RF bandwidth of each device is 20MHz. The bandwidth of transmitted signal is 10 MHz, with the remaining 10MHz bandwidth used for spectrum reallocation. There are three models as follows.

(1) Rodin. This model uses per-frame spectrum shaping and the multi-subband preamble. We experiment with two different SDR RF bandwidths of 20 and 40MHz; for each RF bandwidth, we use subband bandwidths of 1 and 2MHz. The bandwidth of the COTS signal is half of the SDR bandwidth, with the other half of the SDR bandwidth used for spectrum reallocation. For example, a Rodin device with a SDR and COTS bandwidth of 20 and 10MHz respectively and a subband bandwidth of 2MHz will require $N_F = 5$ subbands to span the COTS bandwidth and $N = 2N_F$ subbands to span the SDR bandwidth. At the beginning of each measurement slot (1.8s), Rodin measures the RSSI of all subbands and selects the N_F subbands with the lowest RSSI. This is equivalent to selecting the set of N_F subbands with the lowest interference powers. If all subbands have RSSIs lower than a predefined threshold, Rodin transmits a frame over those time slots. Rodin can carry out

this measure-shape-transmit process within a single time slot due to its per-frame spectrum agreement and shaping capability. The performance of Rodin is modeled based on the I-FOP detection probability measured in the previous section.

(2) COTS-Spec. This model can bond multiple subbands for a single transmission, but cannot change the bonding on a per-frame basis. The bandwidth configuration used in COTS-Spec is identical to that of the Rodin model. At the beginning of a time slot (1.8s), it selects the N_F subbands with the lowest RSSI as before. However, these selected subbands are used only in the *next* time slot. The set of subbands used for the current transmission is selected in the previous time slot. This represents the delay required by a COTS device to switch to a different set of subbands. Note that this is an optimistic model because (a) we do not consider the additional overhead required for spectrum agreement and (b) we assume that COTS-Spec can continue to transmit in the current time slot even as it is changing its set of bonded subbands.

(3) COTS-Mono. In this model, the COTS device makes use of the middle 10 or 20MHz bandwidth of the channel (depending on the bandwidth of the COTS device) for transmitting a frame, but no spectrum shaping is used. This represents a typical 802.11-type device that uses monolithic spectrum blocks for transmission.

(4) Oracle. This is the Rodin model with a subband bandwidth of 200kHz (the smallest allowable bandwidth with the trace data). This models the performance of Rodin without any limitations on the bandwidth and number of its subband filters.

Channel model. We are interested in finding the number of time slots during which each of these models can find a transmission opportunity. We evaluate the performance of the four models using two channel bandwidths of 20 and 40MHz. The RF bandwidth of the SDR is set to 20 and 40MHz respectively. To evaluate the performance of each model, we partition the frequency slots each of the three traces into non-overlapping 20 or 40MHz channels and simulate the operation of each model on all the channels. The threshold levels that we use for 770, 2250 and 5250MHz trace sets are -100, -90 and -90dBm, respectively. These are chosen to be similar to the 802.22 standard for 770MHz data set and the 802.11 standard for the others. Any 200kHz time-frequency slot with an RSSI that exceeds this threshold is assumed to be occupied by a primary transmitter. A subband is considered to be available at a particular time if and only if all frequency slots at that time have RSSIs lower than the threshold. We assume that there is only a single transmitter-receiver pair in each channel as it is sufficient to capture the behavior of the device models under a wide range of channel conditions. We leave the study of Rodin-to-Rodin interference to future work.

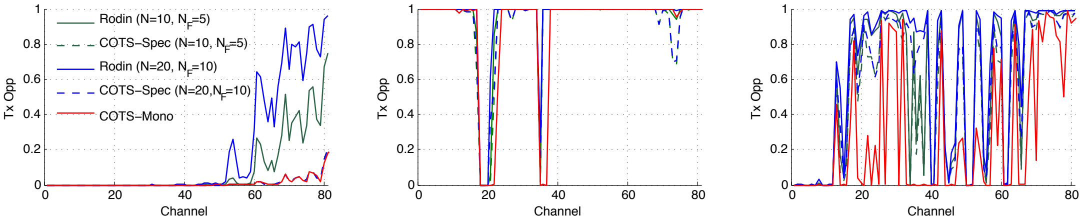
2.8.2 Simulation Results

Channel characteristics. The gain from per-frame spectrum shaping depends on the temporal variability—the more frequently the interference level on the channel changes, the greater the need for fast spectrum shaping. Fig. 2.13 shows the correlation coefficient of the RSSI on each measurement slot over time, for each trace set. Channels within the 5250MHz data set experience high temporal variability and have a median correlation coefficient of about 0.3. On the other hand, channels within the 770 and 2250MHz data sets experience minimal temporal variability, as seen by the high correlation coefficients. We expect the gain from per-frame spectrum shaping to thus be greater in the 5250MHz channels than in channels at other frequencies.

Transmission time slots. Fig. 2.16 shows the proportion of time slots in each channel in which the different devices can find transmission opportunities. Note that the channels are labeled in increasing order of their center frequencies. In the 5250MHz trace set, as shown in Fig. 2.16a, the high temporal variability of the channel means that subbands found to be available for transmission in one time slot are unlikely to still be available in the next time slot. Hence, COTS-Spec with 1MHz subbands can only transmit in up to 15% time slots. COTS-Spec with 2MHz subbands fails to find any transmission slots. A surprising result is that the performance of COTS-Mono is almost identical to that of COTS-Spec with 1MHz subbands. This shows that under highly varying channels, slow channel adaptation with narrow subbands performs almost identically to no spectrum adaptation; while slow channel adaptation with wider subbands fails to find any transmission opportunities.

The per-frame spectrum shaping of Rodin enables it to transmit on a significantly larger proportion of the time slots—up until 95% of the time slots in channel 81. Furthermore, we note that time slot utilization is increased when we use smaller subband bandwidths—Rodin using 1MHz subbands ($N = 20$, $N_F = 10$) can outperform the same device using 2MHz subbands ($N = 10$, $N_F = 5$) by more than 50% in some channels. Note that channels 1-50 in the 5250MHz data set fall into spectrum that is completely occupied by interferers. Hence, no slots can be found by any devices.

The performance of COTS-Spec improves under the low temporal variability of the 770 and 2250MHz trace sets. Fig. 2.16b shows that the fraction of time slots used by COTS-Spec is almost equal to that used by Rodin for transmissions. However, in Fig. 2.16c, we see that even in channels with high correlation coefficients, Rodin still finds more transmission opportunities than COTS-Spec at the same subband bandwidth. This is seen between channels 20 and 30. COTS-Mono performs poorly even on channels with low temporal variation, as shown in both Figs. 2.16b and 2.16c. Spectrum shaping is still necessary here as the low temporal channel variability does not imply the widespread availability of high bandwidth



(a) 5250MHz.

(b) 2250MHz.

(c) 770MHz.

Figure 2.16: Proportion of time slots that each of the devices, Rodin, COTS-Spec and COTS-Mono, can transmit in.

channels.

2.9 Discussion

Interaction with COTS devices. Increasing the SDR-COTS integration can improve the performance of per-frame spectrum shaping. Using rate adaptation as an example, the SDR can provide the COTS hints on the SNR of other channels, so that the COTS device can immediately select the appropriate rate to match the per-frame spectrum when a spectrum reallocation is performed.

Per-frame spectrum shaping in the network. Rodin transparently combines multiple spectrum fragments into a contiguous virtual channel that is seen by the COTS device. Since it obtains these spectrum fragments with a CSMA policy, we expect multiple Rodin nodes to interact without the need for more complex channel access protocols. Our current Rodin prototype is limited to single-link operation and we leave more detailed network-scale studies to future work.

COTS devices using non-contiguous spectrum. Rodin is designed for the case where the RF bandwidth of the SDR frontend is larger than that of the COTS device. At present, Rodin does not support COTS devices using non-contiguous bandwidths. As the SDR/ASIC platform evolves and supports larger bandwidths, Rodin can be extended to support non-adjacent frequency blocks.

Rodin with more than two spectrum shaping filters. Our experimental evaluation of spectrum shaping uses only two shaping filters due to FPGA resource constraints. However, given a larger FPGA, we can increase the number of shaping filters in Rodin. Furthermore, this can be accomplished while keeping the total overlapping bandwidth unchanged.

Rodin with wideband COTS devices. The variability in the channel response is known to increase with channel bandwidth. Hence if Rodin spreads a wideband spectrum (such as a 80MHz signal from an 802.11ac device) to an even wider band, additional processing steps such as Rodin-specific pilots might be necessary to compensate for the greater distortion seen on the channel. Other parameters, such as the overlapping bandwidth of the filters, might also need to be adjusted. However, since wideband COTS devices will already have built-in capability to accommodate the greater channel distortions, the modifications needed for Rodin might be minimal.

2.10 Related Work

Spectrum Agility. WhiteFi [38] is a variable-bandwidth 802.11-based prototype that provides protocols that govern channel-switch triggers, channel probing and selection in whitespaces. This idea of variable-bandwidth communications is also used by FLUID [10] in enterprise networks. Jello [30] extends this variable bandwidth idea to support non-contiguous channel bonding in challenging networks. TIMO [14] adopts a different approach to handling interference on MIMO channels, treating interference as a single MIMO streams while simultaneously transmitting frames on the remaining MIMO streams. SVL [31] and Picasso [39] are both spectrum-shaping layers for general wireless devices. However, these solutions require tight integration with the COTS device's PHY and are not fast enough to support per-frame shaping. The new IEEE 802.11ac standard draft also specifies *non-contiguous 80+80 MHz* channel bonding as an optional feature [40], but does not support per-frame shaping. SWIFT [41] supports transmissions over non-contiguous bands while avoiding interference from narrowband devices. However, it differs from Rodin as it does not support per-frame spectrum shaping and agreement. Furthermore, it is not compatible with any available COTS devices and networks.

Spectrum Agreement. SIFTs [38], part of WhiteFi, is a single-channel bandwidth-independent signal detection algorithm used for determining the transmit bandwidth of an AP. FICA [7] uses binary amplitude modulation on multiple OFDM subcarriers, together with tight time synchronization, to enable each device to contend for different spectrum bands. Preamble detection on NC-OFDM networks [42] is useful for communications over disjoint spectral bands, but a separate mechanism must first be used to agree on the spectrum bands. Other typical uses for spectrum agreement include control channels [27] and backup channel lists [28].

Chapter 3

Cooperative Compression of Wireless Backhaul Traffic

3.1 Introduction

The rapidly growing demand for wireless bandwidth in indoor environments is driving the need for novel enterprise wireless architectures [11, 43]. While the complex fading characteristics of enterprise environments mandates a dense deployment of antennas for coverage, the actual capacity improvement is limited by the corresponding increase in the complexity of distributed antenna coordination, spectrum management and interference mitigation. Software-defined cellular networks are the key techology to meet this challenge head-on. It is envisioned that these networks consist of three important components: (a) wideband Radio-Resource Units (RRUs) to support the various wireless protocols that operate over a wide frequency range; (b) a software-defined control plane that responds rapidly to changing demands in the network [44]; and (c) a flexible, integrated yet general cloud-based platform to process the myriad of supported wireless protocols [6] (e.g., 3G, GSM and LTE). These networks are built following the Cloud-RAN philosophy, where feature-limited RRUs simply transmit and receive RF signals while the upper layer protocol operations are carried out in the centralized cloud platform. Such architectures are intended to support new signal processing primitives [45], such as new Coordinated Multi-Point (CoMP) [46], to boost coverage and capacity, and new control and accounting primitives to improve manageability of the network. CoMP networks utilize the cloud-RAN architecture to achieve network MIMO transmissions via tight PHY-layer coordination between physically separate RRUs.

Network Model. We focus on the challenge of transporting I/Q samples for CoMP over indoor enterprise cellular networks, as shown in Fig. 3.1. The network consists of multiple RRUs that transmit/receive signals over the wireless channel, and a shared enterprise data center that executes the DSP and supported wireless protocols. The RRUs and DSP cloud are

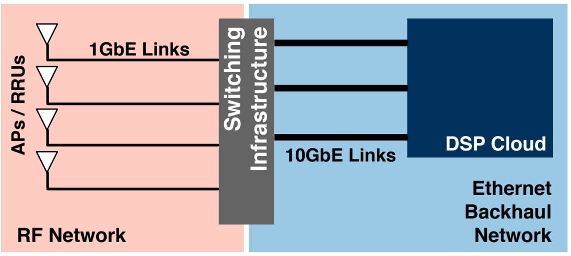


Figure 3.1: Cloud-RAN architecture used by Spiro.

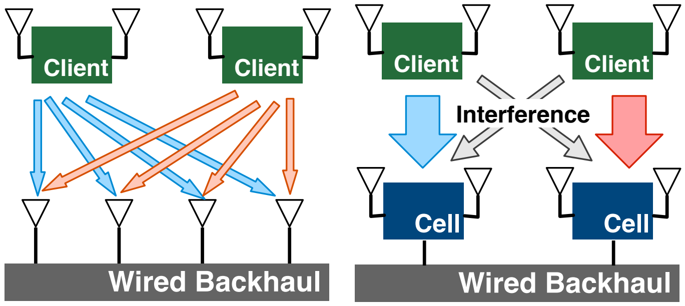
connected via a general-purpose enterprise Ethernet switching infrastructure. The datacenter and ethernet infrastructure is shared with non-wireless traffic that is carried throughout the enterprise. While it is possible to deploy dedicated connectivity and cloud resources for a CoMP network, the ability to reuse shared resources will significantly reduce the cost and complexity of CoMP deployments.

Backhaul Bandwidth Demand. An implicit, but important, assumption underlying the entire software-defined wireless architecture is that there exists a high bandwidth, low latency backhaul network that connects these three components together. This backhaul is responsible for transporting both data and control information throughout the wireless infrastructure network. However, this very assumption is also the most likely to handicap real-world deployments of software-defined wireless networks, especially in indoor environments where most of wireless access occurs.

A key challenge in such unified networks comes from the high bandwidth demand on the backhaul network. Novel DSP algorithms and CoMP techniques require the transport of modulated I/Q samples, rather than unmodulated data bits, over the backhaul network for centralized, cooperative (de)modulation. This high bandwidth load places intense strain on the backhaul network [47]. This is particularly problematic in shared enterprise networks where the Ethernet backhaul is also used for transporting backbone traffic throughout the enterprise.

State-of-the-Art. Current cellular networks address this challenge using dedicated backhaul networks to transport analog RF-over-Copper [48], RF-over-Fiber [49] or digitized I/Q signals [50]. Lossy compression schemes can be applied to these RF signals [51] to reduce the backhaul bandwidth demands at the cost of reduced wireless throughput. However, the cost and complexity of deploying specialized switches and other signaling equipment necessary to (de)modulate the analog are prohibitive, especially for smaller or cost-conscious enterprise environments.

Our Objective. We raise and address an important question: *Can we transport digitized I/Q samples in a CoMP cellular network over widely-deployed enterprise shared Ethernet networks?* Such indoor enterprise environments are characterized by the necessity for dense



(a) CoMP **(b)** Non-CoMP

Figure 3.2: Uplink transmission in CoMP and non-CoMP networks.

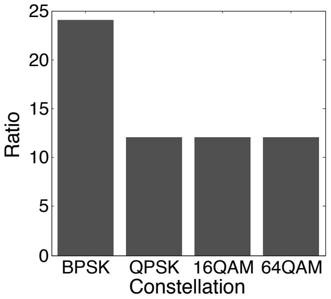


Figure 3.3: Ratio of CoMP to non-CoMP bandwidth.

RRU deployments and the presence of a shared Ethernet backbone and datacenter resources. The challenge is to time-multiplex both high-bandwidth, digitized RF traffic and existing enterprise traffic on the same wired Ethernet backhaul network while meeting their respective performance requirements.

Our Contributions. We design and implement Spiro , a CoMP transport protocol that carries I/Q data over a shared enterprise Ethernet infrastructure between the RRUs and the DSP cloud. A summary of our contributions are:

(a) Cooperative compression with little-to-none wireless capacity reduction. We demonstrate that by cooperatively compressing RF signals from coordinated RRUs, we *can* reduce overall backhaul bandwidth demands *without any loss of wireless capacity.* This result is particularly surprising and important since at the PHY layer, a critically sampled (i.e. non-oversampled) OFDM cellular signal is not sparse and thus, not losslessly compresible. Hence, typical approaches such as sub-Nyquist sampling [32] and compressed sensing [52] cannot be used to reduce the RF bandwidth.

(b) Loss-resilient PHY transport. Spiro employs a loss-resilient PHY transport protocol that allows Ethernet switches to rapidly and randomly discard I/Q samples in the event of backhaul congestion with minimal impact on the wireless capacity. This is in stark contrast to typical SDR DSP operations where the loss of even a small number of I/Q samples due to frame drops (as seen in the USRP and WARP) can result in the loss of the entire wireless data frame.

(c) Real-world evaluation on a large SDR testbed. We implement and evaluate our bandwidth reduction and PHY transport on a large SDR testbed of 16 WARP devices.

3.2 Challenges and Approaches

Challenge I: Backhaul Bandwidth vs. Wireless Capacity. CoMP networks achieve greater wireless capacity, at the cost of greater complexity due to cooperative demodulation of sampled RF signals. A CoMP network, shown in Fig. 3.2a, can utilize four concurrent spatial streams with full coordination between all antennas. On the other hand, a non-CoMP network (Fig. 3.2b) with the same number of transmit and receive antennas, can only use two spatial streams, one per client, for data transmission. The remaining stream from each client is needed for interference nullification [53].

However, the backhaul bandwidth required by CoMP is significantly greater than the non-CoMP network. The number of bits generated by the four CoMP antennas that is sent to a centralized DSP server can be expressed as

$$N_{\text{CoMP}} = \frac{2N_{\text{ant}}N_bR}{\log_2 N_{\text{const}}} \tag{3.1}$$

where N_b is the number of bits transmitted by each client, N_{const} the modulation constellation size, R the number of bits used by the Analog-to-Digital (ADC) quantizer, and N_{ant} the number of receive CoMP antennas. The factor of 2 is needed as we transmit both the I and Q samples. We ignore additional bits that may be received due to oversampling, channel probing and synchronization overheads as they can be trivially removed by the RRU before transmission over the backhaul.

The non-CoMP network with the same number of transmit and receive antennas but without cooperative demodulation, as shown in Fig. 3.2b, requires a maximum of $2N_b$ bits on the backhaul network to represent the same transmission by the two clients. Fig. 3.3 shows the ratio of the backhaul bandwidth demands of CoMP to that without cooperative demodulation. With BPSK, CoMP incurs 24× the enterprise traffic bandwidth while this ratio falls to 12× at higher modulation rates.

Approach I: RF Compression. We adopt lossless and lossy compression techniques to reduce the bandwith of the RF stream. We attain a greater reduction of bandwidth through lossy compression, but this comes at a price of reduced wireless capacity. The challenge, therefore, is to find an optimal trade-off between the achieved wireless capacity and the backhaul bandwidth demand of CoMP networks.

Uplink vs. Downlink. Uplink CoMP transmissions requires I/Q samples to be sent on the Ethernet backhaul. On the other hand, downlink transmissions only require the information bits, rather than the modulated I/Q samples, to be sent over the backhaul network to the CoMP antennas. Hence, *uplink* CoMP I/Q traffic requires *up to 24 times* (i.e. more than

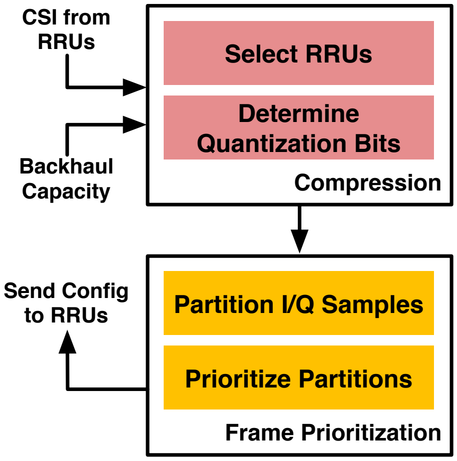


Figure 3.4: SPIRO-Cloud controller on the DSP cloud.

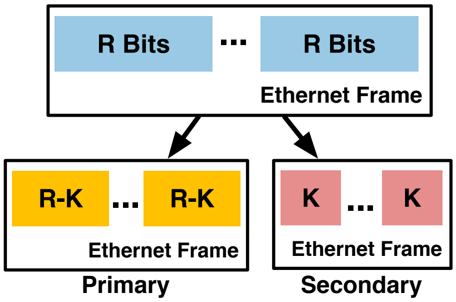


Figure 3.5: A single frame is split into two frames carrying $R-K$ and K-bit samples.

an order of magnitude) more backhaul bandwidth than downlink traffic. Given that the number of downloaded *bits* exceeds that of uploaded bits by only a factor of 6 [54], uplink CoMP backhaul traffic will easily saturate the backhaul network. Hence, we will focus on addressing the CoMP challenges for uplink traffic.

Challenge II: Variable Backhaul Capacity. There is a sizeable diversity of applications communicating over the shared wired backhaul in enterprise environments. As a result, the throughput and reliability of flows in these networks undergo significant variability [55, 56]. For example, in datacenter networks, virtualization and sharing of the network between tasks has been shown to vary between 1Gbps and zero over tens of milliseconds [57]. A resilience to such high network variability must be developed before PHY processing on commodity datacenters [6] is even possible.

An unexpected reduction in available backhaul Ethernet bandwidth will force network switches to shape the backhaul RF traffic by dropping frames containing I/Q samples. This partial loss of critical RF data will result in an unpredictable degradation of wireless capacity.

Approach II: Frame Partitioning and Prioritization. We prioritize and partition the ADC output into separate primary and secondary I/Q frames. These frames are then assigned to different priority queues. When the backhaul capacity is reduced unexpectedly, the switch can drop I/Q frames from the priority queues with minimal impact on the wireless capacity.

3.3 Overview of SPIRO

We design and implement a novel bandwidth-aware RF management protocol, called Spiro. For a given wireless channel capacity, Spiro can reduce the required backhaul bandwidth by over 50%. Specifically, Spiro is designed with the following salient properties.

Property 1: Bandwidth-Awareness. It is difficult to accurately and efficiently track the rapidly changing backhaul capacity. Hence, the compressed RF streams must be *shapable*—in the event of network congestion, the Ethernet switches must be able to randomly drop the specially-constructed frames carrying RF information without significantly affecting the wireless capacity of the CoMP network.

Property 2: Bandwidth-Compression. A CoMP system relies on both spatial diversity and multiplexing gain from multiple RRUs for cooperative demodulation. Spiro coordinates the real-time compression of RF signal from each RRU by reduces the number of bits used to quantize I/Q samples, so that the backhaul bandwidth demand of the CoMP system is reduced. The challenge in this distributed compression approach comes from the fact that it must be coordinated using only the CSI of the channel from each RRU, and without detailed knowledge of the statistics of the received data signal.

Property 3: Minimal RRU Usage. Multiple operators typically share the same CoMP deployment to reduce installation costs. Hence, CoMP network deployments must share the set of RRUs across multiple wireless protocols. Spiro aims to minimize the number of RRUs required to meet a pre-specified wireless channel capacity. The selection of RRUs must take into account the compression ratio at each RRU, and vice versa [58].

Spiro is designed to operate within a CoMP/Cloud-RAN infrastructure as shown in Fig. 3.1. The architecture consists of N_R RRUs that are deployed throughout an indoor environment, and a DSP cloud resource that processes the PHY and other components of the wireless protocol. A shared Ethernet backhaul is used to connect the RRUs to the back-end DSP cloud. Spiro manages the backhaul bandwidth demands from these N_R RRUs to support N_T concurrently transmitting client devices.

Spiro consists of 2 key components: Spiro-Cloud and Spiro-RRU. Spiro-Cloud is a controller module that executes on the DSP cloud every T_{config} time period. The period T_{config} is chosen to minimize the control overhead of Spiro, while ensuring that Spiro can respond to changes in wireless capacity demands and backhaul bandwidth availability. As an example, T_{config} in LTE networks can be selected to be 10ms—the duration of a superframe. At the start of each control interval, Spiro-Cloud computes three pieces of information: (a) $\mathbf{S_R}$, the set of RRUs that are active during the next T_{config} interval; (b) $\mathbf{R}_{\text{opt}}$, the set of optimal quantization widths used by each active RRU; and (c) the Ethernet queuing priority of all frames generated by each RRU. This information is then sent to Spiro-RRU.

Spiro-RRU, which runs continuously on each RRU, receives this information from Spiro-Cloud at the start of each T_{config} interval. If the RRU is an active one (i.e., it is in $\mathbf{S_R}$), it compresses the uplink I/Q samples from the ADC according to its pre-computed quantization width. It then transmits the I/Q samples back to the DSP cloud for processing, using Ethernet frames with the pre-determined priorities. Note that the overhead of control signaling is low as only a small amount of control information is exchanged every T_{config}. For clarity, the variables and parameters used by Spiro are listed in Table 3.1.

3.3.1 First-Order Redundancy Elimination

PHY layer transmissions include redundant information due to the OFDM cyclic prefix, oversampling, preamble and pilot tones that are used for time and frequency synchronization, and channel state measurements. These redundancies can be trivially eliminated at the RRUs and are not transmitted over the backhaul network. We emphasize that **Spiro** ***only operates on critically sampled (i.e. non-oversampled) I/Q signals that have all redundancies eliminated. Hence, all reductions in backhaul bandwidth demands by*** **Spiro** ***are achieved with respect to critically sampled I/Q signals.***

3.3.2 Lossy Compression via Quantization

The ADCs in RRUs map the analog input signal into a complex-valued fixed-point numbers with each of the I and Q components spanning R bits. Let $x^{(R)}$ be a sampled value (either I or Q) that is quantized using R bits. ADCs typically use $R = 12$ or 14 to minimize the distortion that will be introduced into a wide variety of signals.

We compress these sampled signals lossily by using $r < R$ bits to represent them. The I and Q components are rounded to the nearest r-bit fixed-point number using

$$\Delta(x^{(r)}) = round\left(x^{(R)} \cdot 2^{r-1}\right)/2^{r-1}. \tag{3.2}$$

Since actual value of each I/Q component is between $\pm 2^{-(r-1)}$, the total signal-to-quantization noise ratio (SQNR) is given by

$$\text{SQNR(dB)} = 20\log_{10}(2^r). \tag{3.3}$$

Hence, every one-bit reduction in the number of quantization bits results in a 6.02dB reduction in SQNR. Our evaluation will show that a decrease in SQNR does not necessarily decrease the wireless throughput.

3.3.3 Lossless Block Compression

Spiro can reduce the backhaul bandwidth further via lossless block compression of quantized I/Q samples. In this thesis, we will consider (a) the lower bound on the bandwidth reduction as given by entropy encoding, and (b) the achievable bound given by a real-world Huffman encoder implementation. We leave the development of a more advanced streaming I/Q block compression algorithm to future work. We stress that lossless block compression is applied *after* Spiro has optimally quantized the I/Q samples from the active RRUs. It is difficult, if not impossible, to quantize block-compressed codewords. Furthermore, our evaluation will show that the gain from block compression is greater when applied to quantized than un-quantized I/Q samples.

3.3.4 Backhaul Bandwidth Management

Cooperative processing of I/Q data can require up to 24 times the bandwidth of traditional non-CoMP networks. In order to deal with this demand, Spiro reserves a portion of the total wired backhaul capacity for transporting I/Q data to and from the RRUs. The fraction of backhaul capacity reserved is a tunable parameter that depends on minimum wireless capacity that is to be supported by the CoMP system. The larger the minimum required wireless capacity, the greater the fraction of reserved backhaul capacity required.

We reserve backhaul bandwidth with the 802.1p priority queues and Guaranteed Minimum Bandwidth (GMB) support found in the HP6600 enterprise switch.

Additionally, Spiro measures the available backhaul bandwidth every T_{config} interval. If more bandwidth is available, Spiro will opportunistically use all of that bandwidth, even if it exceeds its reserved amount. The loss-resilience feature of Spiro means that in the event of network congestion, the Ethernet backhaul can randomly drop I/Q frames to free up bandwidth for non-CoMP traffic with minimal degradation of the wireless capacity.

3.4 Detailed Design of SPIRO

3.4.1 SPIRO-Cloud

Fig. 3.4 illustrates the operation of Spiro-Cloud. At the start of each configuration interval T_{config}, Spiro-Cloud receives the CSI from all CoMP RRUs in the network and the measured available Ethernet backhaul bandwidth C_m. It then executes the *compression* and *frame prioritization* stages.

Table 3.1: Variables and parameters used in Spiro.

T_{config}	Configuration interval
R, R_n	ADC quantization width, indexed by the n^{th} RRU
$\mathbf{R}_{\text{supp}}$	All supported quantization widths
$R_{\max}$	Maximum ADC quantization width
$R_{\min}$	Minimum ADC quantization width
K	Number of low priority bits in sampled signal
N_R	Number of RRUs in the CoMP network
$\mathbf{S}$	Set of all RRUs within the CoMP network, with $\lvert\mathbf{S}\rvert = N_R$
$\mathbf{S_R}$	Set of active RRUs within an interval T_{config}
N_T	Number of concurrent mobile transmitters
N_Q	Number of priority queues
$x^{(R)}$	ADC output quantized with R bits
C	Current available backhaul capacity
C_{res}	Reserved backhaul capacity
C_m	Measured available backhaul capacity
$C_{\max}$	Maximum backhaul capacity required by Spiro

RF Compression Stage

The amount of backhaul bandwidth required by the CoMP system can be reduced by compression. Spiro compresses the I/Q samples primarily using quantization. Lossless block compression is then applied to the quantized I/Q data streams.

The backhaul bandwith demand depends on the number of active RRUs, $|\mathbf{S_R}|$, and the ADC quantization width used by the active RRUs, $\mathbf{R}_{\text{opt}} = \{R_n | n \in \mathbf{S_R}\}$. Given a CoMP transmission with N_T transmitters and $|\mathbf{S_R}|$ receiving RRUs, the achievable wireless capacity is given by [59]

$$C_{\text{wl}}(\mathbf{S_R}, \{R_n | n \in \mathbf{S_R}\}) = \log_2 \det \left(\mathbf{I} + \mathbf{H}^* \mathbf{Q}^{-1} \mathbf{H}\right) \tag{3.4}$$

where $\mathbf{H}$ is the $|\mathbf{S_R}| \times N_T$ CSI of the system and $\mathbf{Q}$ is the SNR of the system given by

$$\mathbf{Q} = \text{diag}\left([\rho_1 + \gamma(R_1), \ldots, \rho_{|\mathbf{S_R}|} + \gamma(R_{|\mathbf{S_R}|})]\right).$$

ρ_n and $\gamma(R_n)$ are, respectively, the channel and quantization noises for the n^{th} RRU, $n \in \mathbf{S_R}$. The corresponding backhaul capacity demand is proportional to

$$C_{\text{backhaul}}(\mathbf{S_R}, \{R_n | n \in \mathbf{S_R}\}) \propto \sum_{n \in \mathbf{S_R}} R_n. \tag{3.5}$$

If Spiro determines that the backhaul bandwidth demand can be increased, it can achieve a corresponding increase in wireless capacity by increasing either the number of active RRUs in $\mathbf{S_R}$, or the number of quantization bits used by each RRU, or both. However, the actual wireless bandwidth gain due to each of these options depends on (a) the channel state and (b) the noise seen at each RRU. Unfortunately, the optimal choice of active RRUs and quantization widths that gives the greatest overall wireless capacity can only be found via an exponential-time 2D search over the space defined by $\mathbf{S_R}$ and R_n. In SPIRO, instead of using such an expensive approach, we adopt the heuristic in §3.5 to obtain $\mathbf{S_R}$ and R_n.

Frame Partitioning & Prioritization Stage

SPIRO prioritizes the Ethernet frames to achieve bandwidth-aware I/Q transmission so that the Ethernet switch can drop frames, according to priority, during a congestion event without significant impact on the quality of the wireless channel.

At each RRU, let $x^{(R)}$ be the ADC output of an I/Q component that is quantized using $R \leq R_{\max}$ bits. As an example, consider the case where SPIRO partitions $x^{(R)}$ into two components $x^{(R-K)}$ and $y^{(K)}$. $x^{(R-K)}$ is simply the value of $x^{(R)}$ further quantized using only $R - K$ bits and

$$y^{(K)} = x^{(R)} - x^{(R-K)}$$

is encoded using K bits. Each RRU then creates two different Ethernet frames, one that contains only x^{R-K} samples and the other only $y^{(K)}$ samples, as shown in Fig. 3.5. We refer to these frames as the *primary* and *secondary* I/Q frames, respectively.

SPIRO partitions each $x^{(R)}$ sample into one primary frame and one or more secondary frames. To ensure decodability at the DSP cloud, the primary frame will always have a higher priority than the secondary frames.

We reconstruct $x^{(R)}$ from the primary and secondary frames according to

$$x^{(R)} = x^{(R-K)} + y^{(K)}.$$

If the secondary frame is dropped, I/Q sample information is still preserved in $x^{(R-K)}$, albeit with higher quantization noise. However, we cannot recover any information from the secondary frame alone. Hence, SPIRO assigns the primary frame a higher priority than the secondary frame.

Let N_Q be the number of priority queues available in the Ethernet backhaul network. SPIRO-Cloud sorts the primary and secondary frames from all RRUs in decreasing order of their priorities. The sorted frames are then divided equally amongst the N_Q priority frames in order of priority. For example, if $N_Q = 2$, SPIRO-Cloud maps the first half of the sorted frames to the high-priority queue and the bottom half of the frames to the low-priority queue.

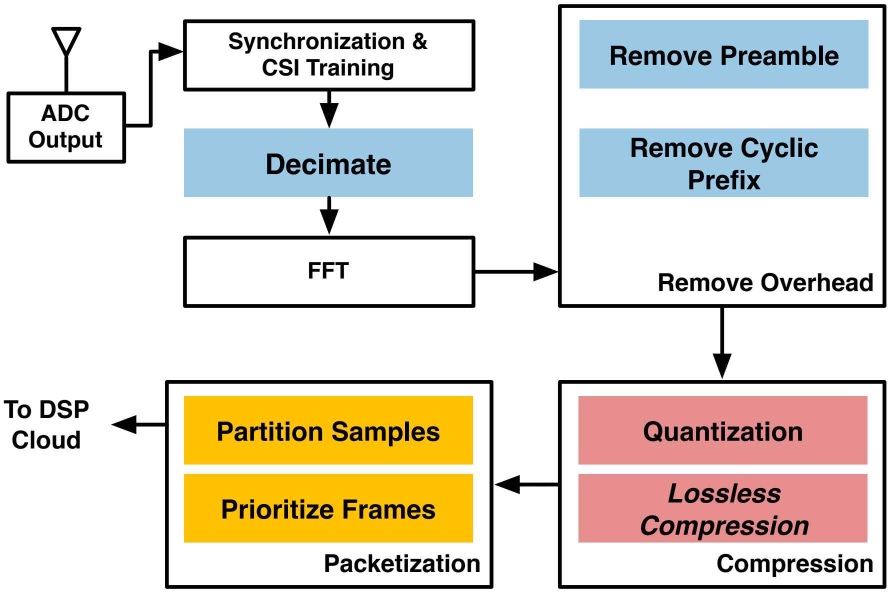


Figure 3.6: SPIRO-RRU controller on the RRU.

3.4.2 SPIRO-RRU

Fig. 3.6 shows the operation of SPIRO-RRU that executes continuously on each RRU. The SPIRO-RRU first locally processes all parts of a frame that does not require cooperative decoding. This reduces the number of I/Q samples that need to be sent to the DSP cloud, which in turn reduces the demand for backhaul bandwidth.

Compression. The I/Q samples are quantized using R_n bits, as specified by SPIRO-Cloud. SPIRO-RRU can reduce the bandwidth demands even further by using a lossless block compression algorithm to the quantized I/Q samples. However, this approach suffers from the complex processing and long latency of lossless compression algorithms. While the development of a lossless compression algorithm for streaming I/Q samples is beyond the scope of this thesis, we will still study the achievable bandwidth reduction with lossless compression.

Packetization. SPIRO-RRU then partitions the remaining I/Q data samples into primary and secondary components, and constructs the corresponding Ethernet frames from them. These frames are then sent over the shared Ethernet backhaul to the DSP cloud.

Supported range of quantization widths, $\mathbf{R}_{\text{supp}}$. For the sake of clarity, we show the quantization step in Fig. 3.6 to be after the FFT operation. However, in a hardware implementation, the quantization of data symbols can occur *before* the finite-precision FFT without incurring any additional loss of precision. For example, quantization can be carried out by using multi-resolution ADCs [60] to improve efficiency. To address this possibility, we will also evaluate the performance of lossy compression under a finite set of quantization widths, $\mathbf{R}_{\text{supp}}$.

3.5 Algorithms in SPIRO

3.5.1 Bandwidth Compression

I/Q quantization in Spiro involves a trade-off between spatial diversity and quantization noise. Spiro-Cloud can adopt two different approaches to lossy compression: uniform and non-uniform quantization.

Non-Uniform Quantization. We search over all combinations of supported ADC quantization widths, $\mathbf{R}_{\text{supp}}$, and RRU subsets to find the optimal solution pair, $(\mathbf{S}_\mathbf{R}, \mathbf{R}_{\text{opt}} = \{R_n | n \in \mathbf{S}_\mathbf{R}\})$, of quantization rates and RRUs. Unfortunately, the optimal solution is found via a complicated combinatorial integer optimization, which severely limits its applicability to real-time environments. Thus, we relax the integer constraints to obtain a convex optimization formulation that can be executed quickly.

Uniform Quantization. We simplify our compression algorithm even further by using only the same quantization for all RRUs and a sub-optimal antenna selection algorithm [58]. Our evaluation results indicate that given the same backhaul capacity constraints, it achieves similar wireless channel throughput to the non-uniform algorithm. However, the uniform quantization approach uses more RRUs than the non-uniform algorithm.

Uniform Quantization

```
Algorithm 3: Uniform quantization
Input: $\mathbf{H} = [\mathbf{H}_f, f = 1, \ldots, N_{\text{FFT}}]$ is a vector of $N_R \times N_T$ CSI matrices, one for each OFDM
        subcarrier; $C_m$ is the measured available backhaul capacity
Output: $(\mathbf{S}_{\text{opt}}, R_{\text{opt}})$
Data: $\mathbf{S}$ = Set of all RRUs in a CoMP network
begin
    $b_{\max} \leftarrow 0$;
    for $R \in \mathbf{R}_{\text{supp}}$ do
        $\mathbf{S}_\mathbf{R} \leftarrow$ FindActiveRRUs($\mathbf{S}$, $\mathbf{H}$, $R$, $C_m$);
        $\mathbf{Q} \leftarrow$ diag($\mathbf{n}^{(\mathbf{S}_\mathbf{R})}$) $+ \mathbf{I}_{|\mathbf{S}_\mathbf{R}|} \cdot 2^{-2R}$;
        $b \leftarrow \sum_{f=1}^{N_{\text{FFT}}} \log_2 \det \left( \mathbf{I}_{N_T} + \mathbf{H}_f^{(\mathbf{S}_\mathbf{R})*} \mathbf{Q}^{-1} \mathbf{H}_f^{(\mathbf{S}_\mathbf{R})} \right)$;
        if $b > b_{\max}$ then
            $b_{\max} \leftarrow b$; $R_{\text{opt}} \leftarrow R$; $\text{S}_{\text{opt}} \leftarrow \text{S}_\text{R}$;
        end
    end
end
```

Algorithm 3 describes the uniform quantization. For each supported quantization width $R \leq R_{\max}$, we determine the optimal set of RRUs, $\mathbf{S}_\mathbf{R}$, using the `FindActiveRRUs` function

```
Algorithm 4: FindActiveRRUs
Input: $\mathbf{S}$ is the set of all RRUs in CoMP network; $\mathbf{H} = [\mathbf{H}_f, f = 1, \ldots, N_{\text{FFT}}]$ is a vector of
       $N_R \times N_T$ CSI matrices, one for each OFDM subcarrier; $R$ is the ADC quantization
       width; $C_m$ is the measured available backhaul capacity
Output: $\mathbf{S_R}$ = Set of selected RRUs
Data: $N_{\text{FFT}}$ = number of OFDM subcarriers
begin
    $\mathbf{S_R} \leftarrow \mathbf{S}$;
    $V \leftarrow$ compute bits per I/Q sample from $C_m$;
    $v \leftarrow |\mathbf{S_R}| \times R$;
    while $v > V$ do
        $\mathbf{Q} \leftarrow \texttt{diag}(\mathbf{n}^{(\mathbf{S_R})}) + \mathbf{I}_{|\mathbf{S_R}|} \cdot 2^{-2R}$;
        foreach $1 \le f \le N_{\text{FFT}}$ do
            $\mathbf{B}_f \leftarrow \left(\mathbf{I}_{N_T} + \mathbf{H}_f^{(\mathbf{S_R})*}\mathbf{Q}^{-1}\mathbf{H}_f^{(\mathbf{S_R})}\right)^{-1}$;
        end
        $k_{\min} \leftarrow \arg\min_{k \in \mathbf{S_R}} \sum_{f=1}^{N_{\text{FFT}}} \left|\mathbf{H}_f^{(k)*}\mathbf{B}_f\mathbf{H}_f^{(k)}\right|$;
        $\mathbf{S_R} \leftarrow \mathbf{S_R} \setminus \{k_{\min}\}$;
        $v \leftarrow |\mathbf{S_R}| \times R$;
    end
end
```

in Algorithm 4. We then select the optimum $(R, \mathbf{S_R})$ pair that achieves the highest wireless bandwidth, under the constraint that the backhaul bandwidth demand does not exceed the measured available bandwidth.

In these algorithms, $\mathbf{n}$ is the vector of channel noise at each RRU and $\mathbf{n}^{(\mathbf{S_R})}$ is a subvector consisting only of the elements indexed by $\mathbf{S_R}$. $\mathbf{H}_f$ denotes an $N_R \times N_T$ CSI matrix of the f^{th} subcarrier and $\mathbf{H}_f^{(\mathbf{S_R})}$ denotes a submatrix using rows from $\mathbf{H}_f$.

The key operation in Algorithm 4 is found in lines **??**-**??**. Here, `FindActiveRRUs` searches for the RRU that contributes the least to the wireless capacity. This RRU will be dropped from the active set in order to reduce the backhaul bandwidth demand. Let $\mathbf{S_R}^{(-k)} \triangleq \mathbf{S_R} \backslash \{k\}$ for some $k \in \mathbf{S_R}$. The capacity of $\mathbf{S_R}^{(-k)}$ RRUs is

$$\begin{aligned} C(\mathbf{S_R}^{(-k)}) &= \log_2 \det \left(\mathbf{I}_{|\mathbf{S_R}^{(-k)}|} + \mathbf{H}_f^{(\mathbf{S_R}^{(-k)})*}\mathbf{Q}^{-1}\mathbf{H}_f^{(\mathbf{S_R}^{(-k)})}\right) \\ &= C(\mathbf{S_R}) + \log_2 \left(1 - \mathbf{H}_f^{(k)*}\mathbf{B}\mathbf{H}^{(k)}\right) \end{aligned} \tag{3.6}$$

where $\mathbf{B}$ is defined in line **??**. Removing the k^{th} RRU reduces the wireless capacity by $\sum_{f=1}^{N_{\text{FFT}}} \log_2 \left(1 - \mathbf{H}_f^{(k)*}\mathbf{B}\mathbf{H}^{(k)}\right)$. In each iteration, the RRU which incurs the smallest capacity reduction is dropped.

Non-Uniform Quantization

The set of non-uniform quantization values can be determined using the following steps.

1. For each $\mathbf{S_R} \subset \mathbf{S}$, find $\mathbf{R} = \{R_n | n \in \mathbf{S_R}\}$ using

   $$\max \sum_{f=1}^{N_{\text{FFT}}} \left| \log_2 \det \left(\mathbf{I}_{|\mathbf{S_R}|} + \mathbf{H}_f^{(\mathbf{S_R})*} \mathbf{U}^{-1} \mathbf{H}_f^{(\mathbf{S_R})} \right) \right|$$
   $$\text{s.t.} \sum_{n=1}^{N_R} 2R_n \le V, \quad R_n \in \mathbf{R}_{\text{supp}}$$

   where $\mathbf{U} = \operatorname{diag}\left(\mathbf{n}^{(\mathbf{S_R})} + 2^{-2\mathbf{R}^{(\mathbf{S_R})}} \right)$.

2. Choose the $(\mathbf{S_R}, \mathbf{R})$ pair that achieves the highest wireless capacity, as according to (3.4).

However, actually performing this optimization is challenging because (a) it requires a combinatorial search over all subsets of RRUs and (b) the optimization problem is an NP-complete integer programming problem since R_n only takes integer values. Instead, we solve a simplified problem

$$\max \sum_{f=1}^{N_{\text{FFT}}} \log_2 \det \left(\mathbf{I} + \mathbf{H}_f \mathbf{H}_f^* \mathbf{W}^{-1} \right)$$
$$\text{s.t.} \sum_{n=1}^{N_R} 2\overline{R}_n \le V, \quad 0 \le \overline{R}_n \le R_{\max}$$

where $\mathbf{W} = \operatorname{diag}\left(\mathbf{n} + 2^{(-2\overline{\mathbf{R}})} / \overline{\mathbf{R}} \right)$ and $\overline{\mathbf{R}} = [\overline{R}_n, \ldots, \overline{R}_{N_R}]$. Note that $\overline{R}_n$ are real, not integer, values. We then use the RRU-selection step, as shown in Algorithm 5, to obtain the final RRU selection and corresponding quantization width, $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$.

3.5.2 Frame Prioritization

Spiro uses Algorithm 6 to construct the quantization width used in the primary and secondary I/Q frames. We first compute the optimal $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$ given the measured backhaul capacity constraint, C_m, using either the uniform or non-uniform antenna selection. Also, let λ be the smallest number of quantization bits that is used to represent each I/Q sample in the secondary frame. In our implementation, we find that $\lambda = 2$ bits offers the best results. The frame prioritization algorithm takes $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$ and λ as input, and computes the priority of primary and secondary frames from each active RRU.

Algorithm 5: Non-uniform RRU selection

```
Input: $\overline{\mathbf{R}} = [\overline{R}_1, \ldots, \overline{R}_{N_T}]$
Output: $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$
begin
    $\mathbf{R} \leftarrow [R_1, \ldots, R_{N_T}]$ where $R_n = \max(\min(\lceil \overline{R}_n \rceil, R_{\max}), R_{\min})$ for $1 \leq n \leq N_T$;
    while $\sum_{n=1}^{N_T} R_n > V$ do
        $k \leftarrow \arg\min_{1 \leq n \leq N_T} \overline{R}_n$; $R_k \leftarrow 0$; $\overline{R}_k \leftarrow \infty$;
    end
    $\mathbf{S_R} \leftarrow \{n | R_n > 0\}$;
    $\mathbf{R}_{\text{opt}} \leftarrow \{R_n | n \in \mathbf{S_R}\}$;
end
```

Algorithm 6: Compute the priority of I/Q frame partitions

```
Input: $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$, $\lambda$
Output: $\mathbf{P}$ is the priority queue of I/Q frame partitions
begin
    $\mathbf{R} \leftarrow \mathbf{R}_{\text{opt}}$;
    $\mathbf{P} \leftarrow []$;
    while $\exists R_n \geq R_{\min} + \lambda$, $n \in \mathbf{S_R}$, $R_n \in \mathbf{R}$ do
        $b_{\max} \leftarrow 0$; $n_{\max} \leftarrow []$;
        foreach $n \in S_R$ do
            if $R_n \leq \lambda$ then continue;
            ;
            $R'_n \leftarrow R_n - \lambda$;
            $\mathbf{R}' \leftarrow [R_1, \ldots, R_{n-1}, R'_n, \ldots, R_{|\mathbf{S_R}|}]$;
            $\mathbf{Q} \leftarrow \texttt{diag}(\mathbf{n}^{(\mathbf{S_R})} + 2^{-2\mathbf{R}'})$;
            $b \leftarrow \sum_{f=1}^{N_{\text{FFT}}} \log_2 \det \left( \mathbf{I}_{|\mathbf{S_R}|} + \mathbf{H}_f^{(\mathbf{S_R})*} \mathbf{Q}^{-1} \mathbf{H}_f^{(\mathbf{S_R})} \right)$;
            if $b > b_{\max}$ then $b_{\max} \leftarrow b$; $n_{\max} \leftarrow n$; ;
        end
        $R_n \leftarrow R_n - \lambda$;
        $\mathbf{P} \leftarrow \text{append}(\mathbf{P}, (n_{\max}, \lambda))$;
    end
    while $|\mathbf{S_R}| > 0$ do
        $\mathbf{Q} \leftarrow \texttt{diag}(\mathbf{n}^{(\mathbf{S_R})}) + \mathbf{I}_{|\mathbf{S_R}|} \cdot 2^{-2\mathbf{R}^{(\mathbf{S_R})}}$;
        foreach $1 \leq f \leq N_{\text{FFT}}$ do
            $\mathbf{B}_f \leftarrow \left( \mathbf{I}_{N_T} + \mathbf{H}_f^{(\mathbf{S_R})*} \mathbf{Q} \mathbf{H}_f^{(\mathbf{S_R})} \right)^{-1}$;
        end
        $k_{\min} \leftarrow \arg\min_{k \in \mathbf{S_R}} \sum_{f=1}^{N_{\text{FFT}}} \left| \mathbf{H}_f^{(k)*} \mathbf{B}_f \mathbf{H}_f^{(k)} \right|$;
        $\mathbf{P} \leftarrow \text{append}(\mathbf{P}, (k, R_k))$;
        $\mathbf{S_R} \leftarrow \mathbf{S_R} \setminus \{k_{\min}\}$;
    end
end
```

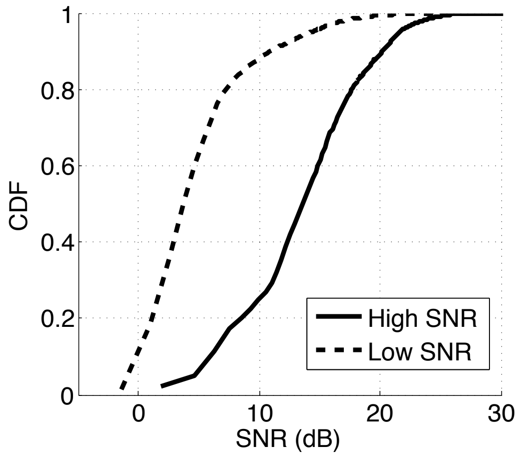


Figure 3.7: Experiments are run in two separate SNR environments.

In the first while-loop (lines **??**-**??**), we partition the I/Q samples from each RRU into multiple groups of λ bits, down to a minimum partition size of $R_{\min}$. These λ-bit partitions are then enqueued into $\mathbf{P}$ in order of increasing priority. This is followed by the second while-loop (lines **??**-**??**) where we prioritize the remaining $R_{\min}$-bit I/Q samples from all RRUs.

Each entry in the priority queue $\mathbf{P}$ is an (n, r) pair where n is the RRU identifier and r is the number of quantization bits to be used at this priority. SPIRO maps $\mathbf{P}$ to N_Q priority queues used in an Ethernet switch by partitioning the entries in $\mathbf{P}$ equally among the N_Q queues. If multiple (n, r) entries from the same RRU are in the same switch priority queue, they are merged into one larger secondary frame.

3.6 Implementation

We implement and evaluate SPIRO on a testbed of 16 WARP SDR platforms running WARPLab, each with 2 antennas, which are all connected to a single HP 6600 48-port switch. The antennas are placed throughout a large server room environment. Obstructions throughout the testbed ensure existence of both line-of-sight and non-line-of-sight channels between different antenna pairs. We use a PC connected to the same switch to manage the testbed.

In each experiment, we randomly select $N_R = 24$ antennas as uplink RRUs and $N_T = 4, 6$ or 8 antennas as concurrent transmitters. We transmit 500 OFDM frames from the N_T transmitters. Each OFDM frame spans $800\mu s$ at a bandwidth of 20MHz, and uses symbols that have 256 subcarriers and 64-tap cyclic prefixes. SPIRO uses the preamble from all N_R antennas to determine the optimal compression solution $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$ and decode the transmitted frame from the active antennas at the corresponding quantization widths. The smallest number of RRUs is always constrained by $N_R = N_T$ to ensure MIMO decodability. If $N_R > N_T$, then the wireless capacity benefits from additional spatial diversity. Our results in this thesis are obtained from experiments in two different SNR ranges, high and low, as shown in Fig. 3.7.

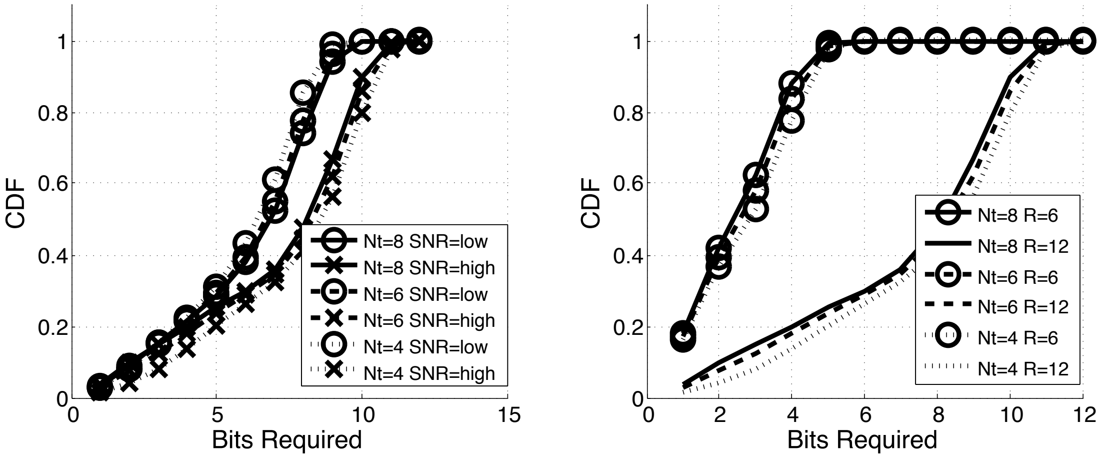
Uplink MIMO. Each of the N_R receivers measures the channel state from each of the N_T transmitters. The received data and CSI I/Q samples are sent to the SPIRO, running on the server, for processing. SPIRO computes the optimal $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$ compression configuration and quantizes the data streams from the $\mathbf{S_R}$ RRUs accordingly. The received frame is then demodulated using quantized I/Q samples from $\mathbf{S_R}$ RRUs with a zero-forcing algorithm.

Latency and Timing Jitter. CoMP networks are sensitive to latency and timing jitter in the I/Q samples received from the different RRUs. In order to determine the timing performance over the Ethernet backhaul, we transmitted 10000 CoMP frames alongside non-realtime traffic over our Ethernet. We observed that both the latency of the sample arrival as well as its timing jitter over our HP enterprise switch is always below $2\mu s$. This delay can be tolerated by an LTE CoMP network since the receiver has $3ms$ to decode a frame [6]. Furthermore, there is a strong focus on reducing switching latency even further through hardware and software techniques [8, 61]. We expect that such developments will further reduce the impact of latency and timing jitter on CoMP deployments over shared Ethernet networks.

Time Synchronization. The N_T transmitters must send OFDM frames concurrently, which are in turn received by the N_R receivers. Synchronization is achieved using an wired Ethernet control frame that is broadcast to all WARP devices. On our testbed, a control frame is broadcast in this manner from the PC to all WARP devices. Each WARP platform immediately starts to transmit or receive when it receives this control frame. We measured the Ethernet broadcast jitter over the Ethernet switch to be always less than $2\mu s$. Hence, given the 40MHz sampling frequency of the WARP platform, the jitter in the start times of the N_T transmitters is well within the duration of the cyclic prefix.

Each of the N_T transmitters prepends a preamble to the OFDM frame. At each of the N_R receivers, the position of the earliest detected preamble marks the start of the CoMP uplink frame.

Frequency and Phase Synchronization. Before each OFDM frame, we randomly select a synchronizing antenna and transmit a 10MHz sine wave for $800\mu s$. We then determine the frequency offset of all other antennas with respect to the synchronizing antenna. This offset is then applied to the OFDM frame that is subsequently transmitted from each of the N_T



(a) Variable SNR

(b) Variable Quantization

Figure 3.8: Distribution of bit lengths under different SNR and quantization levels

transmitters.

Note that *uplink* Spiro only requires frequency synchronization, but not phase synchronization. The phase offsets between the different N_T transmitters can be compensated at the receivers using the CSI. This is unlike *downlink* CoMP systems such as JMB [11] that require the phases of all transmitters to be perfectly synchronized.

3.7 Block Compression of RF Signals

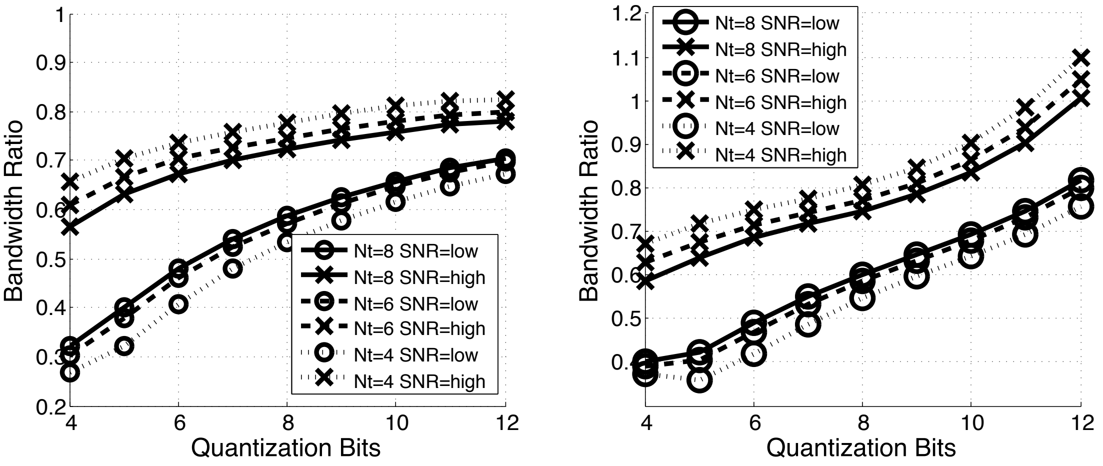
In this section, we evaluate the bandwidth reduction with lossless block compression of real-world RF transmissions in our CoMP testbed.

Metric: Bandwidth Ratio. The bandwidth ratio is defined as the ratio of the average bandwidth demand of a losslessly compressed version of $x^{(R)}$ to that of $x^{(R)}$ without any lossless compression.

3.7.1 Bit Length Distribution

The minimum number of bits required to represent an I or Q value $x^{(R)}$ is given by $B = \lceil \log_2 x^{(R)} \rceil \leq R$. Fig. 3.8a shows the distribution of B under two different SNR conditions. When the SNR is high, $B = 11$ bits are required to successfully represent all I/Q components, while with a low SNR, only $B = 10$ bits are needed.

Fig. 3.8b shows the CDF of B when two different number of quantization bits, $R = 6$ and



(a) Entropy Coding

(b) Huffman Coding

Figure 3.9: Throughput reduction with lossless compression

12, are used. We can observe that the B shows a more uniform distribution at $R = 6$. The median bit length at $R = 6$ and 12 are 3 and 8 bits, respectively.

3.7.2 Entropy Coding

Under entropy coding, the minimum number of bits used to encode $x^{(R)}$ is $-log_2\mathrm{P}(x^{(R)})$ where $\mathrm{P}(x^{(R)})$ is the probability of occurrence of $x^{(R)}$. Entropy coding thus gives an upper bound on the compressibility of the RF signals. Fig. 3.9a shows the bandwidth ratio under entropy coding. Under high SNR conditions, the bandwidth ratio with the original 12-bit RF signal ranges from 0.78 to 0.83. The achieveable compression ratio increases proportionally with the number of concurrent, interfering transmitters. As the number of quantization bits is reduced to 4 bits, the bandwidth ratio due to lossless coding can be further reduced to between 0.58 to 0.65.

Entropy coding can compress low SNR signals to a greater extent. Under low SNR conditions, the minimum number of bits required for the I/Q samples decreases, as seen in Fig. 3.8a. Hence, the bandwidth ratio of a 12-bit signal decreases to between 0.67 to 0.70. As the number of quantization bits is reduced to 4, the bandwidth ratio falls to between 0.27 and 0.32. However, unlike the high SNR case, the bandwidth ratio varies proportionally to the number of concurrent transmitters.

3.7.3 Huffman Coding

Huffman compression [62] encodes $x^{(R)}$ with variable length prefix codes. Fig. 3.9b shows the bandwidth ratio of Huffman compression. When $x^{(R)}$ spans the full width of the ADC (i.e., $R = 12$), Huffman coding increases the required bandwidth under high SNR conditions. This occurs because the overhead of the dictionary surpasses the bandwidth reduction due to variable length encoding of the data. The bandwidth gain of Huffman compression decreases as the number of quantization bits is reduced, but it still does not reach entropy coding bound. Under low SNR, we can get up to a 20% reduction in bandwidth using Huffman coding. We observed that the bandwidth ratio of Huffman coding remains largely similar even if we increase the codeword size to span multiple I/Q samples.

3.8 Lossy Compression and Prioritization

We now evaluate (a) the uniform and non-uniform quantization algorithms, and (b) the performance of frame prioritization in the event of backhaul bandwidth fluctuations.

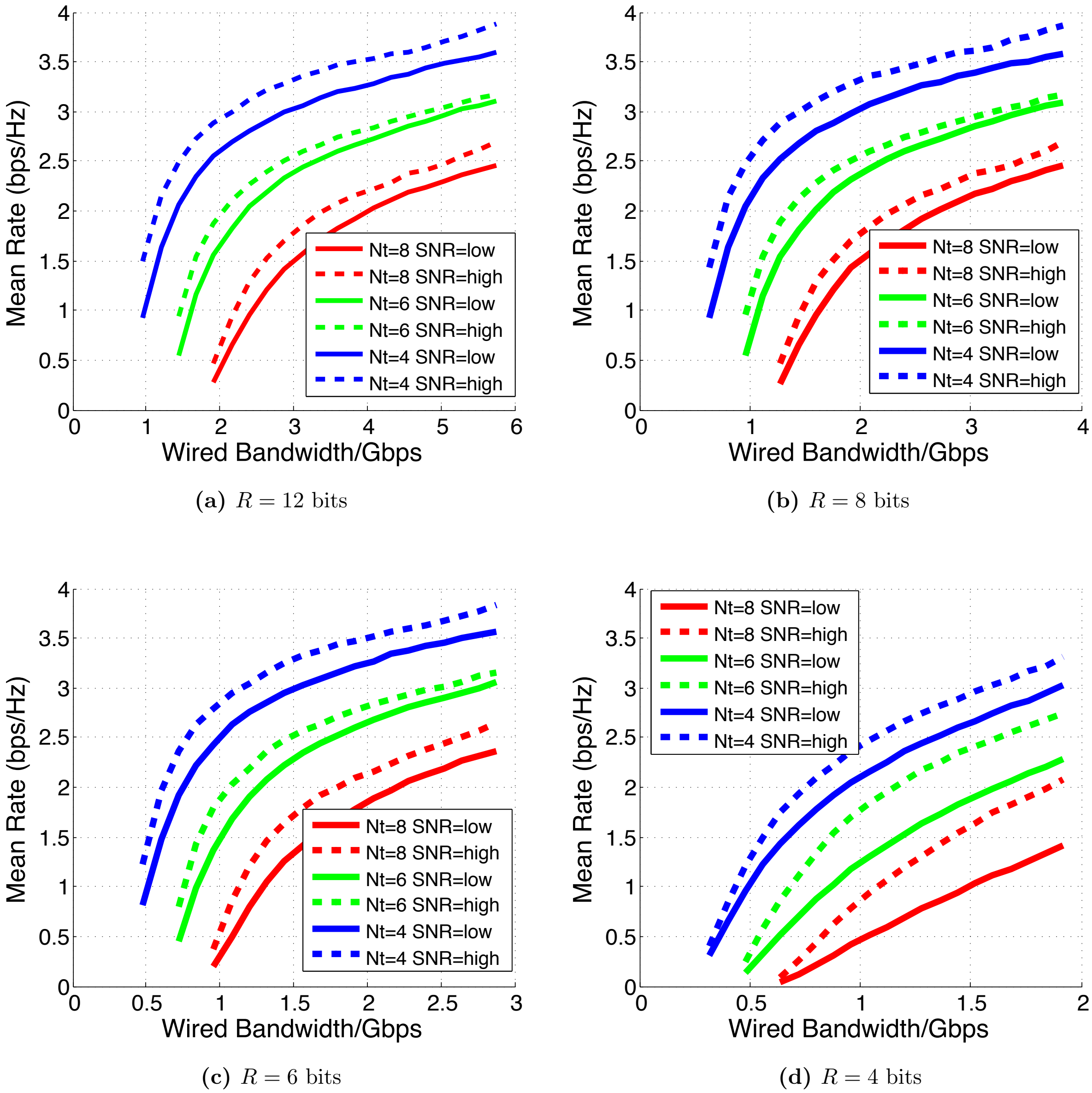
3.8.1 Quantization

What is the baseline evaluation of our uplink CoMP testbed?

Fig. 3.10a shows the wireless rate per user achieved by RRU selection under 12-bit uniform quantization as we increase the wired backhaul capacity available to Spiro. The I/Q samples here do not require any additional quantization since the WARP platforms already come equipped with 12-bit ADCSs. The achievable wireless rate depends on (a) the number of RRUs selected, (b) the number of concurrent uplink users and (c) the SNR distribution at the RRUs.

Number of active RRUs. With uniform quantization, the backhaul bandwidth demand is met by varying the number of active RRUs that send I/Q samples back to the DSP cloud. As we increase the number of active RRUs, the wireless rate per user increases due to the increased spatial diversity. For $N_T = 4, 6$ and 8 transmitters, the wireless rate per user reaches a maximum of 3.8, 3 and 2.55 bits/s/Hz under high SNR when all 24 RRUs are active.

Number of concurrent transmitters. The achieveable mean wireless rate per user decreases as we increase the number of concurrent users. This is due to the increased interference encountered from the imperfections in time and frequency synchronization that is found in real-world uplink transmitters. Such imperfections lead to power leakage from the



(a) $R = 12$ bits

(b) $R = 8$ bits

(c) $R = 6$ bits

(d) $R = 4$ bits

Figure 3.10: Spiro with uniform quantization

channel of one transmitter to another, thus reducing the SNR of each of the N_T decoded frames.

SNR. The wireless rate per user is lower with the low SNR experiment as expected. However, the rates achieved by the low and high SNR experiments are within 10% of each other.

How much backhaul bandwidth can we save by reducing the quantization width of all RRUs?

Figs. 3.10b and 3.10c show the wireless rate per user under increasing wired bandwidth constraints when we quantize the I/Q samples with 8 and 6 bits, respectively. Note that one can quantize I/Q samples from our testbed using 6 bits (down from the original 12-bit ADC output) without any loss of wireless performance. There are two key findings to observe.

First, *given the same target rate per user, when we reduce the number of quantization bits from 12 to 6, the backhaul bandwidth requirement is reduced by 50% from the original 12-bit I/Q samples and the number of RRUs required is unchanged.* This bandwidth reduction from lossy compression is greater than that achieved by lossless entropy coding (Fig. 3.9a).

Second, *under uniform quantization, the achievable wireless capacity is dominated by the degree of spatial diversity as we reduce the number of quantization bits to 6.*

However, we cannot quantize the I/Q samples with fewer than 6 bits without any loss in wireless capacity. As an example, compare the performance of $R = 6$ with that of $R = 4$. When we have a backhaul capacity limit of 1Gbps, we achieve 2.3bits/s/Hz when using $R = 4$ and 2.8bits/s/Hz with $R = 6$. This is in spite of the fact that the 12 RRUs are active with $R = 4$ while only 8 are used with $R = 6$. This disparity is evident even at other backhaul bandwidth constraints. Hence, when we use fewer than 6 quantization bits, the increase in quantization noise overwhelms any gains we obtain from increased spatial diversity.

Can we reduce the number of active RRUs?

We can reduce the number of active RRUs with non-uniform quantization. We use $\mathbf{R}_{\text{supp}} = \{4, \ldots, 12\}$ to demonstrate this. Fig. 3.11 shows the rate per user of $N_T = 4, 6$ and 8 with non-uniform quantization under high SNR conditions. We also plot the rate per user with uniform $R = 6$ quantization on the same figure for comparision. Observe that *for a given backhaul bandwidth constraint, non-uniform quantization can achieve the same wireless rate the uniform quantization approach.* Furthermore, non-uniform quantization comes with an added benefit.

Fig. 3.12 compares the number of RRUs used by non-uniform quantization and $R = 6$ uniform quantization algorithms, for $N_T = 4, 6$ and 8 transmitters. *Non-uniform quantiza-*

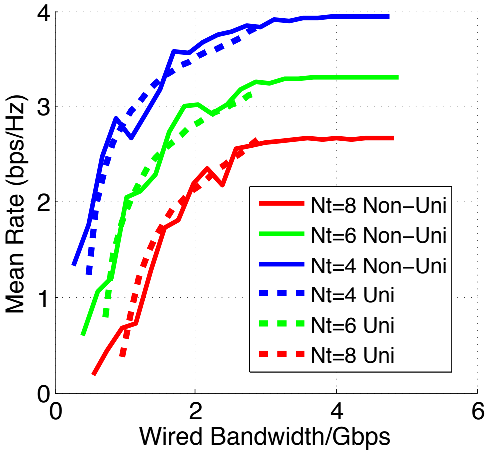


Figure 3.11: Mean rate of non-uniform vs. uniform quantization under the same backhaul capacity bound

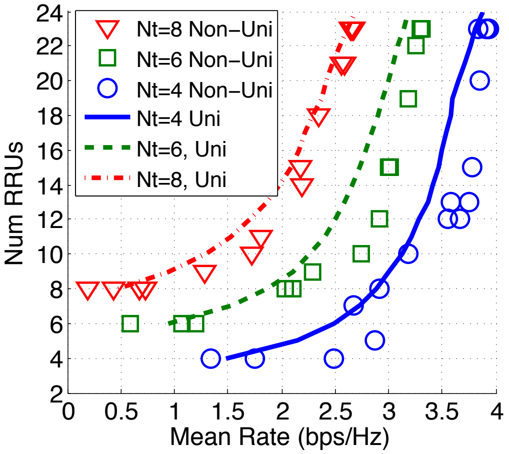


Figure 3.12: Non-uniform quantization requires up to 43% fewer RRUs than uniform quantization

tion requires up to 43% fewer active RRUs to attain the same wireless throughput as uniform quantization.

Hence, when compared to a CoMP network that relies only on an RRU selection algorithm to manage the backhaul bandwidth demands, the *non-uniform scheme requires 50% less backhaul bandwidth and 43% fewer RRUs to maintain the same wireless channel rate per uplink user.*

How much more backhaul bandwidth reduction can we obtain by combining lossless and lossy compression?

Fig. 3.13 shows the additional bandwidth reduction that comes from using entropy coding after quantization. With the block compression algorithms, we can further reduce the bandwidth in high and low SNR scenarios by up to 40% and 72%, respectively.

Can we achieve the same CoMP performance with fewer number of quantization widths?

If quantization is implemented using multiple ADCs or multi-resolution ADCs, then a smaller number of required quantization widths translates into a more efficient hardware implementation. We consider three different quantization ranges: $\mathbf{R}_1 = \{4, 12\}$, $\mathbf{R}_2 = \{4, 8, 12\}$ and $\mathbf{R}_3 = \{4, 6, 8, 10, 12\}$. When $N_T = 4$, the reduction in wireless rates under a 1Gbps (and greater) backhaul constraint is less than 5% when $\mathbf{R}_2$ and $\mathbf{R}_3$ are used. Such small reductions

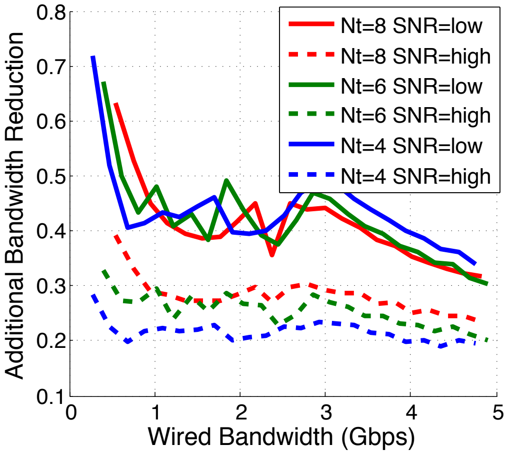


Figure 3.13: Additional bandwidth reduction from lossless compression

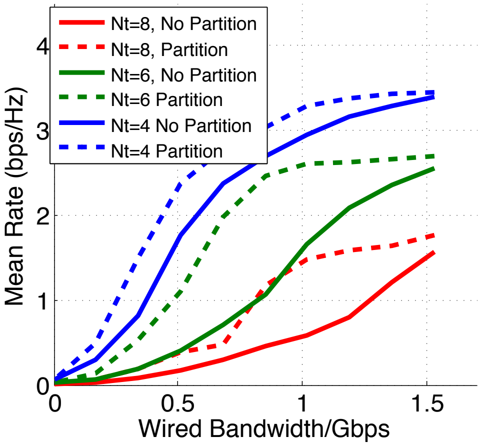


Figure 3.14: Rate per user with frame prioritization

can also be seen with $N_T = 6$ and 8. However, we see more drastic reductions in throughput with $\mathbf{R}_1$. In particular, when $N_T = 8$, up to 75% relative reduction in wireless rate is seen under a 1Gbps backhaul constraint.

3.8.2 Frame Partitioning and Prioritization

How much benefit do we get from frame partitioning?

Fig. 3.14 compares the wireless rate per user using frame prioritization with and without frame partitioning, under the high SNR scenario. To obtain these results, we first compute the optimal $(\mathbf{S}_{\mathbf{R}}, \mathbf{R}_{\text{opt}})$ solution given a backhaul capacity C_m of 1.5Gbps using non-uniform quantization. The partitioned and unpartitioned I/Q streams are generated using $\lambda = 2$ and $\lambda = 0$ in Algorithm 6, respectively. We then reduce the backhaul bandwidth usage by discarding Ethernet frames carrying I/Q samples at the switch, in order of priority. To ensure optimal prioritization, we use $N_Q = 80$ priority queues—each primary or secondary frame will thus be in its own queue and in the event of congestion, frames are dropped in a strict order of priority.

By partitioning the I/Q samples into primary and secondary Ethernet frames, we ensure that frame losses will primarily increase quantization noise, while maintaining spatial diversity for as long as possible. This has two primary consequences: *(a) frame partitioning and prioritization has greater benefits for transmissions with a larger number of concurrent users (i.e $N_T = 6$ and 8)* and *(b) in the event of frame losses at the switch, we retain up to 3 times more wireless capacity with* SPIRO *frame partitioning and prioritization.* Fig. 3.15

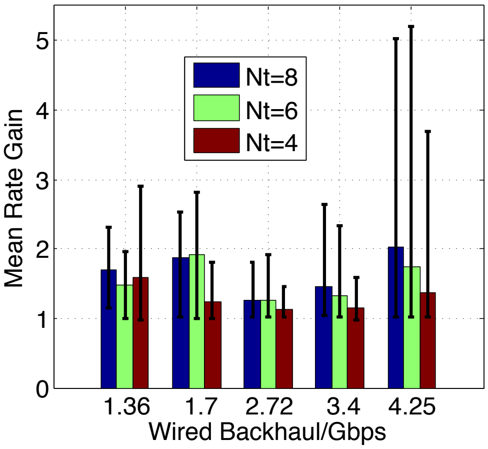


Figure 3.15: Rate gain with frame partitioning vs without partitioning, under different backhaul capacity constraints, C_m and $N_Q = 80$.

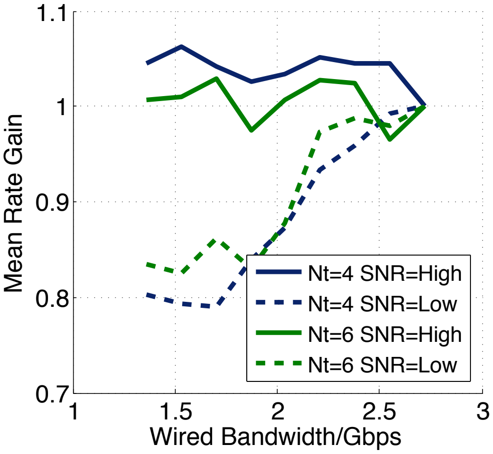


Figure 3.16: Wireless rate gain of priority-based frame drops vs optimal compression using $(\mathbf{S_R}, \mathbf{R}_{\text{opt}})$.

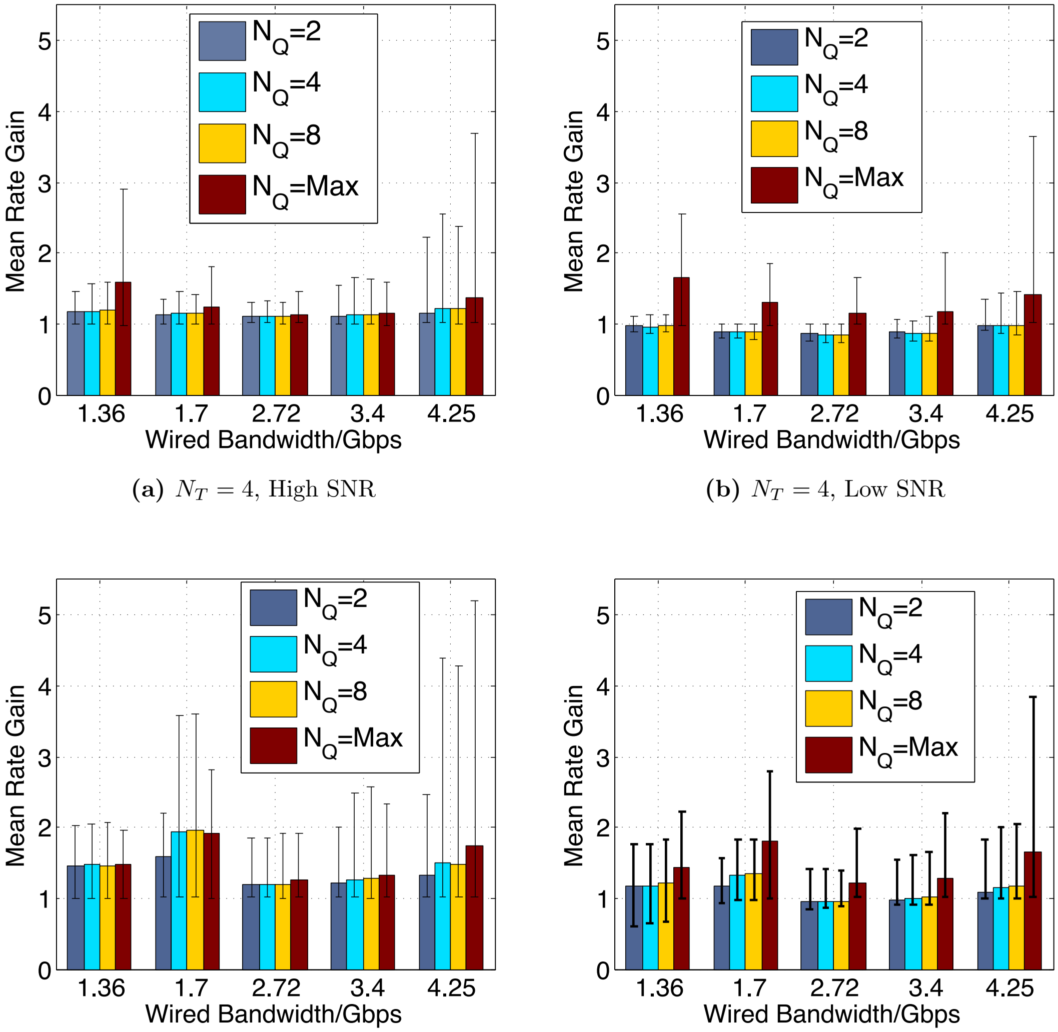
shows that this observation holds at other backhaul constraints C_m. Here, each bar shows the average gain in the wireless rate per user, while the error bars demarcate the maximum and 5^{th} percentile gains.

How does frame partitioning perform with fewer priority queues?

Commercially available Ethernet switches have far fewer than 80 priority queues. However, we can still benefit from frame partitioning and prioritization with fewer queues. Fig. 3.17 shows the gains under $N_Q = 2, 4$ and 8 priority queues. Under high SNR situations, improvements in per-user rates can be achieved with fewer priority queues, with situations involving a larger number of concurrent users, $N_T = 6$, seeing larger gains than those with fewer concurrent users, $N_T = 4$. However, under low SNR conditions, frame partitioning and prioritization have a small negative impact on the per-user rates when $N_T = 4$ concurrent users are active. In such situations, a larger number of priority queues is necessary to obtain the benefits of frame prioritization in Spiro.

How well does priority-based frame-drops compare to optimal I/Q compression?

We compare the wireless rate achieved by using priority-based frame-drops with that obtained by our optimal bandwidth compression in Fig. 3.16. For the frame prioritization algorithm, we use $C_m = 2.4$Gbps. We can see that under high SNR, the wireless rate achieved by frame prioritization and drops, is similar to that obtained by optimal compres-



(a) $N_T = 4$, High SNR

(b) $N_T = 4$, Low SNR

(c) $N_T = 6$, High SNR

(d) $N_T = 6$, Low SNR

Figure 3.17: Gains in wireless rate per user from frame partitioning and prioritization. Each bar shows the mean gain, while the error bars denote the maximum and 5th percentile gains.

sion. However, at low SNR, optimal compression can achieve up to 20% higher wireless rate than frame dropping at the switch.

3.9 Discussion

Hardware Complexity of RRUs. Spiro requires the RRUs to include a limited set of DSP blocks such as FFT and preamble correlation to perform (a) first-order redundancy elimination and (b) network MIMO transmissions. We note that RRUs in CoMP settings must support additional features such as distributed time and phase synchronization, along with feedback mechanism for distributed beamforming [11, 5]. Such features do make use of hardware DSP resources on the RRU. For example, frequency drift tracking for zero-forcing beamforming [5] requires an FFT block. Spiro can reuse such basic DSP blocks on the RRUs and therefore, incurs only a very minor additional hardware overhead.

Distributed vs Centralized CoMP. Spiro adopts a centralized CoMP architecture where all cooperative DSP processing occurs in a centralized datacenter/cloud. Such centralization enables straightforward implementation of complex DSP algorithms that have a global view of the network, and allow efficient provision of compute resources that can closely match the wireless traffic on the network. Alternatively, we can reduce the actual backhaul bandwidth using distributed CoMP architectures that employ distributed interference cancellation [63]. However, distributed architectures have increased complexity due to (a) unpredictable communication patterns between RRUs that are influenced by the time-varying channel characteristics at each RRU and (b) complicated RRU designs as complex interference cancellation algorithms are now performed on the RRU itself. It is in our opinion that the true benefit of Cloud-RANs are better achieved through an efficient centralized architecture.

Real-World Block Compression. Our evaluation of lossless compression gains is based on optimal entropy and Huffman coding schemes. Such compression schemes require accurate statistics of the I/Q codewords in order to achieve maximum compression. Unfortunately, measuring the statistics of an I/Q stream in real time will incur an additional delay. In real-world deployments, we can make use of a hardware accelerator and adaptive compression schemes [64] to acchieve the optimal compression performance. We leave the study of such schemes to future work.

3.10 Related Work

Practical network MIMO or CoMP schemes [65, 53] usually assume that the backhaul is capable of transporting the I/Q samples necessary for centralized (de)modulation. However, this assumption may not hold in the presence of interfering cross traffic over the shared Ethernet backhaul. Quantization of RF data [51, 66, 67] has been proposed to reduce the backhaul bandwidth demands of next-generation LTE networks. These proposals focus on compressing RF data from each RRU individually, and do not exploit spatial diversity between antennas. To address this limitation, distributed Wyner-Ziv [68] encoding has been used to jointly compress signals from multiple antennas. Compressed sensing [69, 70, 71] takes a different approach where the the signal is compressed before sampling and digitization by the ADC. However, most WiFi and LTE data signals are not transmitted sparsely, thus limiting the applicability of compressed sensing to these scenarios.

Datacenters in Cloud-RAN deployments are known to have rapidly changing flow behaviors [72, 73] and congestion patterns. Incast TCP traffic [74] also leads to sporadic congestion and packet drops within datacenters. Spiro accommodates such variability by supporting traffic shaping at the switch in the event of congestion.

Chapter 4

Spectrum Coordination

4.1 Introduction

Dynamic spectrum use is a well-known approach to increasing the throughput and the utilization of high-bandwidth WLANs [30, 7, 10] and improving energy-efficiency [9]. However, this approach to spectrum use amplifies the following two aspects of wireless networks.

P1. Multi-channel transmissions. Wireless devices usually combine multiple fragmented spectrum bands [38, 30] to achieve sufficient bandwidth to meet high throughput demands. For example, Jello [30] uses per-session FDMA spanning non-contiguous bands to reduce the proportion of the time that an application experiences high frame losses to a mere 10%. FICA [7] combines channelization of wideband spectrum and frequency-domain contention to achieve up to a 4-fold gain in efficiency over 802.11n. This is a significant departure from the current 802.11 infrastructure WLANs where AP channels are determined at the time of deployment and remain fixed during their operation.

P2. Partially-overlapping channels. Dynamic spectrum use increases the chance of interference between transmissions on partially-overlapping channels [10]. A node that detects a partially-overlapping OFDM frame cannot recover any bits from the non-overlapping subcarriers [75], thus becoming unable to decode it correctly. This problem is well recognized and its existing solutions include centralized spectrum allocation [10], and subcarrier remapping and retransmission [76].

The overhead of accommodating multi-channel and partially-overlapping transmissions will be particularly acute in control frames, since their length is typically small. An analysis of the network traces collected during SIGCOMM 2008 [77] reveals that even though 802.11 management and control frames only make up 12% of the total number of transmitted frames, they occupy 34% of the airtime on the channel. We expect the proportion of the airtime that is taken up by these control frames to increase if we adopt, for instance, a channel-switching approach to multi-channel communications in 802.11 networks. The median channel-switching delay of 15ms [9] is a steep price to pay for transmitting a small control frame.

The primary reason for the high cost of transmitting control frames comes from the fact that they are typically handled similarly to data frames. However, two properties of control frames set them apart from data frames: (1) control frames are typically consumed by network/MAC/PHY protocols and ignored by the upper layers; (2) the bandwidth consumed by control frames is typically low. For example, RTS/CTS frames serve to convey only one bit of information: "is anyone else transmitting right now?" In 802.11 networks, this single bit of information consumes at least two control frames along with the overhead of a CSMA protocol. Clearly, more efficient ways of control frame exchange are desired.

4.1.1 Our Solution: Aileron

In this chapter, we present Aileron, a novel approach to control frame exchange that eliminates the overhead involved in traditional control frame exchange. The key insight behind Aileron is that the information can be encoded using the *modulation rate* (e.g., BPSK, QPSK, 8PSK, etc.) of the individual subcarriers. Control information transmitted in this way is received by recognizing the modulation rate used, and requires little to no frame synchronization. It is resilient to distortions—such as noise, frequency and time drift—due to the channel and hardware imperfections.

Aileron overlays a low bitrate control channel on top of OFDM frames: data is packed into the subcarriers of the OFDM frame by the PHY protocol, while the control information is encoded using the *modulation rate* of these data subcarriers. To see how Aileron works, consider an example case where BPSK, QPSK, and 8PSK are mapped to values 0, 1, and 2, respectively. A transmitter that needs to send an integer-valued control frame first converts the base-10 integer to a ternary number. The modulation rate of each subcarrier in the Aileron control channel is then set according to the value of its corresponding ternary digit. At the receiver, the control frame is recovered by recognizing the modulation rate of each subcarrier and reconstructing the corresponding ternary number. Note that no CSMA overhead is incurred for the control frames transmitted by Aileron.

4.1.2 Where can Aileron be used?

Aileron achieves the capability of asynchronous and simultaneous transmissions of both control and data frames by operating on OFDM *symbols* rather than *frames*, as is the case with typical wireless protocols. This key distinction eliminates much of the coordination overhead incurred by devices operating in multi-channel and partially-overlapping channel

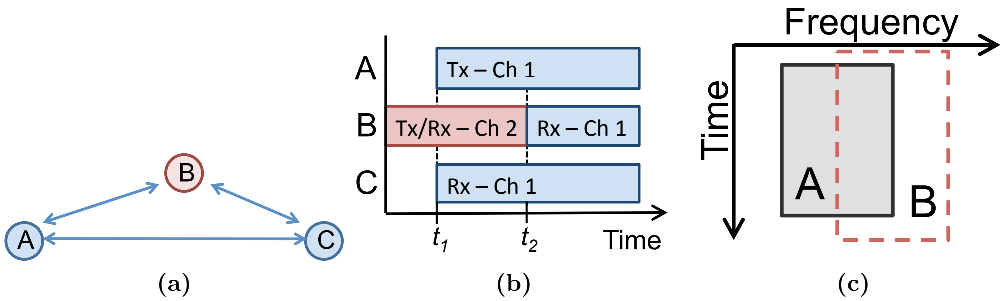


Figure 4.1: (a) Network of 3 nodes; B is Aileron-enabled (b) Multi-channel WLAN: B recovers the modulation types from the partially-overheard frame from A to C. (c) Partially-overlapping channels: B recovers the modulation types from only a fraction of the subcarriers used by A.

environments.

Asynchronicity in multi-channel networks: The importance of asynchronicity in the *time* domain can be easily seen in multi-channel wireless networks. Figs. 4.1a and 4.1b show a 3-node network and its corresponding temporal behavior, respectively. At time t_1, both A and C are communicating with each other over channel 1 while B is communicating with some other device on channel 2. At t_2, B switches to channel 1. Without Aileron, B cannot receive/decode any useful information from A's transmission, since B must achieve proper time and frequency synchronization with A using the frame preamble before any data symbols can be decoded. With Aileron, B can detect the subcarrier modulation rate using any subset of the transmitted symbols (not just from the beginning of the frame) and can thus readily begin decoding the information on the control channel from A at time t_2. We stress that at t_2, A is *simultaneously* sending data to C and control information to B without switching channels.

Asynchronicity in partially-overlapping channels: Alternatively, if A and B are on partially-overlapping channels at time t_1 (as shown in Fig. 4.1c), then Aileron's asynchronicity in the *frequency* domain can be used to bridge this communications gap. In this scenario, Aileron constructs a control channel using the overlapping subcarriers shared by A and B. Control information can be seamlessly passed from A to B without any additional frame synchronization or channel-switching overheads.

Integration into existing networks. An Aileron client can be deployed in networks where only a portion, or even none, of the other nodes support Aileron. When no other Aileron device is present, it functions as a modulation identifier for each subcarrier of an OFDM frame. Consider the two scenarios in Fig. 4.1 again, except that now A and C

are unmodified WLAN devices while B is an Aileron node. In both the multi-channel and partially-overlapping channel scenarios, B identifies the modulation rate of the individual subcarriers. Using this information, B can infer the state of the channel between A and C since the modulation rate is typically selected by an auto-rate algorithm to match the estimated channel condition [78].

When Aileron nodes are mixed with non-Aileron ones, control signaling between Aileron nodes can be done without modifications. Aileron frames will simply be treated as erroneous frames or noise by non-Aileron nodes. To the best of our knowledge, Aileron is the first to encode information in the the modulation rate of subcarriers.

4.1.3 Contributions and organization of the chapter

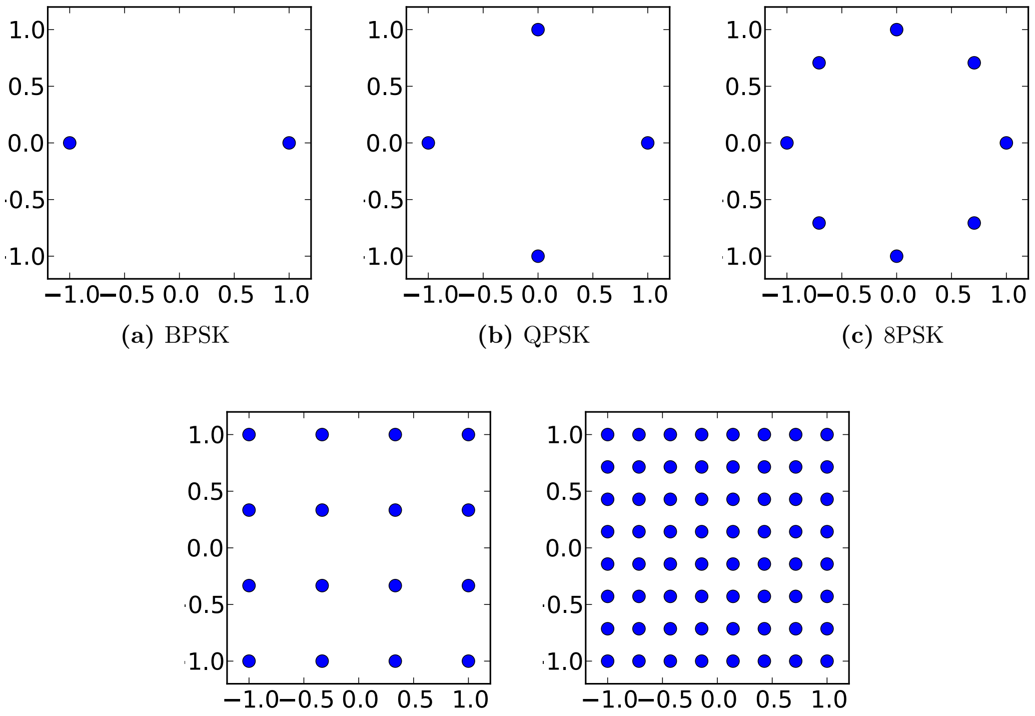
Our contributions can be summarized as follows. First, we design a reliable, low-overhead modulation-based signaling scheme, Aileron. Second, we implement Aileron on a USRP2 platform and demonstrate, via experimentation, its efficiency and robustness. We also evaluate it under a wide range of channel conditions, demonstrating its superior performance under varying channel and mobility conditions. Third, we demonstrate how Aileron can be combined with a FICA-style frequency-domain contention scheme to enable frame aggregation in dynamic spectrum access networks.

The chapter is organized as follows. We give an overview of Aileron in Section 4.2 and describe the key ideas and techniques behind modulation-based signaling in Section 4.3. We then evaluate Aileron using simulations and real-world experiments in Sections 4.4 and 4.5, respectively. We briefly discuss other real-world concerns of Aileron in Section 4.6. To further motivate the benefits of Aileron in real-world networks, we demonstrate two applications of Aileron in Section 4.7. We discuss related work in Section 4.8.

4.2 Aileron Overview

Aileron has active and passive modes. In both of these modes, control information is encoded in terms of the modulation rate of each subcarrier of transmitted data.

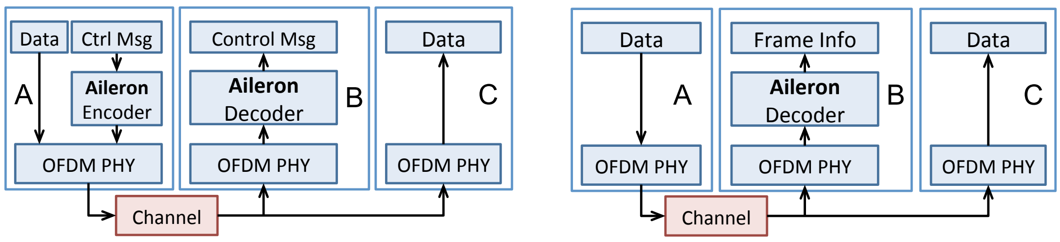
Fig. 4.2 shows the five constellations recognized by Aileron: BPSK, QPSK, 8PSK, 16QAM, and 64QAM. Each point in a constellation diagram is used to encode $\log_2 M$ bits, where M is the total number points in the diagram. For an arbitrary subcarrier, the constellation diagram chosen to encode its bits determines its modulation rate. The PSK and QAM constellations in Fig. 4.2 are each chosen such that lower-level modulations are subsets of higher-level modulations—the QPSK constellation includes the two points of the BPSK



(a) BPSK (b) QPSK (c) 8PSK

(d) 16-QAM (e) 64-QAM

Figure 4.2: Phase-Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM) constellations recognized by Aileron.



(a) Active-mode. *A* is an Aileron-enabled (WLAN) transmitter while *C* is an unmodified (WLAN) receiver.

(b) Passive-mode. *A* and *C* are unmodified (WLAN) devices while *B* is an Aileron node.

Figure 4.3: Active and passive Aileron.

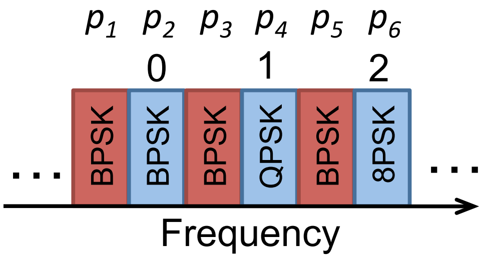


Figure 4.4: Active-mode Aileron used to encode the value 5_{10} that is equal to 012_3.

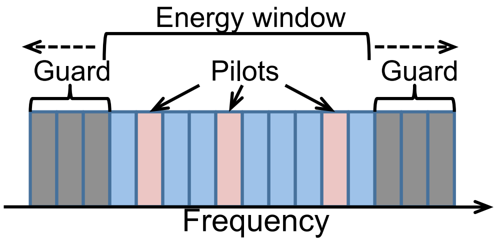


Figure 4.5: An energy window is slid over the OFDM subcarriers to find the coarse frequency offset.

constellation, and likewise, the 8PSK constellation contains the points in both QPSK and BPSK. QAM constellations differ from the PSK constellations in that no constellation point exists along the in-phase and quadrature-phase axes. However, QAM constellations still maintain the subset property, although no QAM constellations are subsets of any PSK constellation, and vice versa.

4.2.1 Active-mode Aileron

Fig. 4.3 illustrates the architecture of the Aileron transmitter and receiver of the example 3-node network in Fig. 4.1. The transmitter, node A, contains an Aileron encoder module that maps the control frame into the modulation rates of the *Aileron subcarriers*. The modulation rates of these subcarriers are limited to BPSK, QPSK and 8PSK, which correspond to the ternary bases, 0, 1 and 2. Subcarriers that are not used for Aileron signaling (i.e., non-Aileron-subcarriers) are not restricted to these constellations. Additionally, the subcarrier that precedes the Aileron subcarrier must be forced to the BPSK modulation. This is done to accommodate the OFDM symbol acquisition algorithm of Aileron as detailed in Section 4.3. The OFDM PHY at A uses these selected modulation rates to generate the OFDM frame that it transmits to C. The Aileron decoder at B recovers the control frame from the symbols received by the OFDM PHY from a multi-channel or partially-overlapping transmission.

For example, Fig. 4.4 shows a set of 6 consecutive OFDM subcarriers from a single 802.11g OFDM symbol, $p_1, \ldots, p_6$, that are used to represent a ternary value, with p_1 being the least significant ternary digit. Suppose that BPSK, QPSK and 8PSK map to integers 0, 1 and 2, respectively. In order to encode the base-10 number 5_{10} to 012_3, we set the non-Aileron-subcarriers p_1, p_3 and p_5 to be BPSK-modulated, and set the Aileron-subcarriers p_2, p_4 and p_6 to be BPSK, QPSK, and 8PSK-modulated, respectively.

4.2.2 Passive-mode Aileron

Rate-control algorithms in wireless networks select the fastest modulation rate given channel conditions. Hence, the state of the channel between a pair of communicating nodes can be inferred from the modulation rate used by them. For example, this information can be integrated into CMAPs [79] to increase the spatial reuse of more challenging whitespace and multi-channel networks.

Passive-mode Aileron does precisely this, identifying the modulation rate of each subcarrier of an unmodified OFDM frame. Fig. 4.3b shows an example Aileron device B that can overhear transmission between two non-Aileron devices A and C. A transmits frames to C using a standard 802.11a/g/n protocol and the Aileron decoder in B identifies the subcarrier modulation rates from the overheard OFDM symbols recovered by the PHY. Aileron can differentiate between BPSK, QPSK and 8PSK. It can also differentiate between PSK and QAM, but the identification of 16QAM and 64QAM is more involved and the subject of our future work.

4.2.3 Automatic modulation recognition

In both passive and active Aileron, the Aileron decoder employs Automatic Modulation Recognition (AMR) [80] to determine the modulation type of each subcarrier in a group of N identically-modulated OFDM symbols. Let $S_k = \{s_{k,1}, \ldots, s_{k,N}\}$ be a sequence of received samples of the k^{th} subcarrier of N consecutive OFDM symbols. These samples are modulated using a constellation $C = \{c_1, \ldots, c_M\}$ with M points. These OFDM symbols must satisfy:

$$\rho(s_{k,i}) = \rho(s_{k,j}),\ i \neq j,\ 1 \leq i, j \leq N,\ 1 \leq k \leq K. \tag{4.1}$$

where $\rho(s_{k,n})$ is the modulation rate of $s_{k,n}$ and K is the total number of subcarriers in each OFDM symbol. Note that it is possible for $\rho(s_{k,n}) \neq \rho(s_{k',n})$ when $k \neq k'$. How to differentiate between these modulations is described in Algorithm 7. Each of the modulation rates—BPSK, QPSK and 8PSK—has an associated decision rule, indicated by the functions `is_bpsk`, `is_qpsk` and `is_8psk`, respectively. Active-mode Aileron only uses BPSK, QPSK and 8PSK: it matches the signal samples against the BPSK and QPSK rules. If the samples match neither of these rules, the modulation is declared to be 8PSK. Passive-mode Aileron matches the signal against all three rules and if no match is found, the modulation of samples is declared to be "QAM". Passive-mode Aileron does not differentiate between 16QAM and 64QAM because the constellation points of QAM are encoded using both magnitude and phase. It is not possible to accurately recover the magnitude without proper calibration using the frame preamble. On the other hand, because it is easy to differentiate between

Algorithm 7: Automatic modulation recognition.

```
Data: $S_k$ is a sequence of $N$ constellation points
Result: Identified modulation
begin
    if is_bpsk($S_k$) then
        return "BPSK";
    else if is_qpsk($S_k$) then
        return "QPSK";
    else if Active-mode or is_8psk($S_k$) then
        return "8PSK";
    else
        return "QAM";
    end
end
```

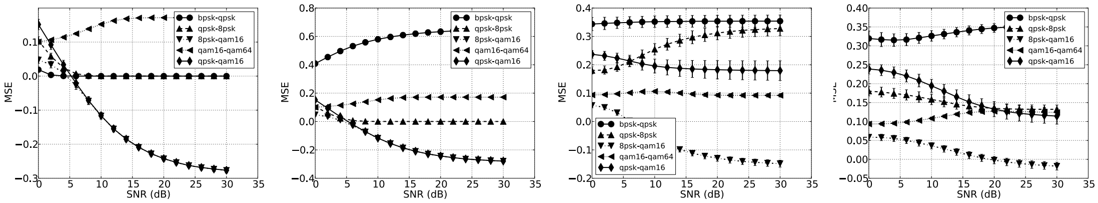
the three PSK schemes, we will restrict the allowable modulation schemes in active-mode Aileron to the PSK modulations to improve signaling reliability.

4.3 Aileron Algorithm Details

4.3.1 How does Aileron acquire an OFDM symbol?

Aileron identifies subcarrier modulation rates from the OFDM symbols that are recovered from arbitrary locations of the transmitted frame. Aileron differs from traditional communication protocols in that it operates on individual OFDM *symbols* rather than *frames*. In typical wireless protocols such as 802.11 and WiMAX, frame acquisition and synchronization is performed using a frame preamble. In Aileron, individual symbols must be acquired *without* any help from the frame preamble. Hence, standard frame synchronization algorithms, such as the Schmidl-Cox algorithm [81], cannot be used here. Here, we describe the detection of OFDM symbols, along with frequency-drift correction and timing-drift compensation.

Symbol recovery and frequency drift correction. The frequency drift θ encountered in an OFDM block can be expressed as $\theta = \Omega + \epsilon$, where Ω is the *coarse* frequency-drift component and is an integer multiple of the subcarrier bandwidth; ϵ is the *fine* frequency-drift component and is smaller than the bandwidth of a subcarrier. A maximum-likelihood acquisition algorithm [82] is used to both acquire the symbol and correct its fine frequency drift. Once the OFDM symbol is identified, an FFT operation is applied to obtain its frequency-domain subcarriers. We correct the coarse frequency drift in the frequency domain by sliding a window, with a bandwidth equal to that of the data and pilot subcarriers, over all subcarriers of the OFDM symbol, as shown in Fig. 4.5. At each window position, the energy of all



(a) BPSK **(b)** QPSK **(c)** 8PSK **(d)** 16-QAM

Figure 4.6: Differences in MSE values for input sequences of different modulation rates

subcarriers within the window is summed. The offset of the window, from its ideal central position, with the highest total energy from the subcarriers is the coarse frequency offset Ω.

Timing-offset compensation. The OFDM acquisition algorithm in [82] cannot always guarantee perfect timing recovery. This timing recovery error induces a phase error in the subcarriers, due to the known property of Discrete Fourier Transforms: a timing offset of l samples introduces a phase error of $e^{-j2\pi kl/M}$ in the k^{th} subcarrier. The corrected symbol Y_k in the k^{th} subcarrier is obtained using the relation:

$$Y_k = X_k \cdot X_{k-1}^{*}/|X_{k-1}| \tag{4.2}$$

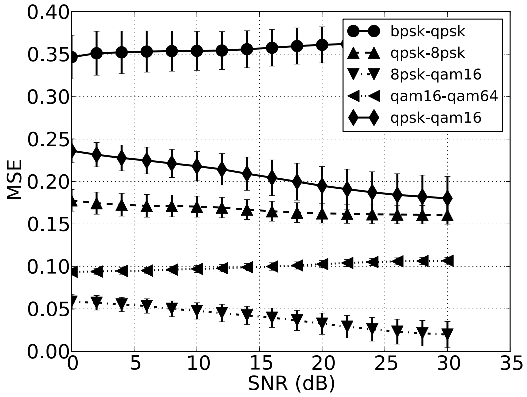
where the $(\cdot)^*$ operator denotes the complex conjugate and X_k is the uncorrected symbol in the k^{th} subcarrier. If the symbols X_k and X_{k-1} are from the same constellation, then this correction will preserve the modulation scheme for subsequent recognition by Aileron. For example, if X_k and X_{k-1} are modulated using QPSK, then Y_k will definitely be one of the QPSK constellation points. However, the actual constellation point held by X_k is lost, thus preventing the original bit content from being recovered. This does not affect Aileron since only the modulation type is of our interest.

In Aileron, if X_k is used to encode a bit of control information, then the modulation rate of X_{k-1} is set to BPSK to maximize the probability of correctly identifying the modulation type of X_k.

4.3.2 What are the decision rules?

Consider a sequence of subcarrier values, S_k, modulated with C. The normalized mean squared error (MSE) between S_k and the ideal constellation points is

$$\text{MSE}_C(S_k) = \frac{1}{N}\sum_{n=1}^{N}\left(\min_{c_m \in C}\left\{\frac{s_{k,n}}{|s_{k,n}|} - \frac{c_m}{|c_m|}\right\}\right)^2. \tag{4.3}$$



(e) 64-QAM

Figure 4.5: Differences in MSE values for input sequences of different modulation rates

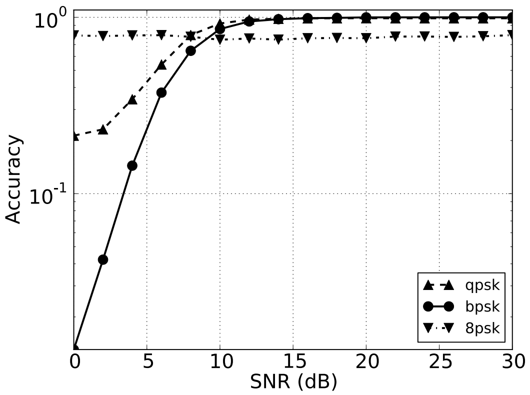


Figure 4.6: Accuracy of active-mode Aileron over a simulated channel with no doppler shift and an AMR window of 10.

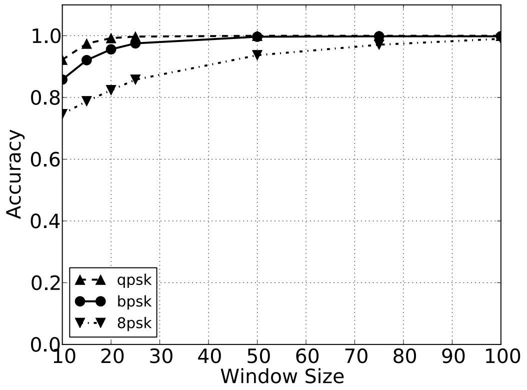


Figure 4.7: Accuracy of active-mode Aileron with different AMR window sizes, no doppler shift and a SNR of 10dB

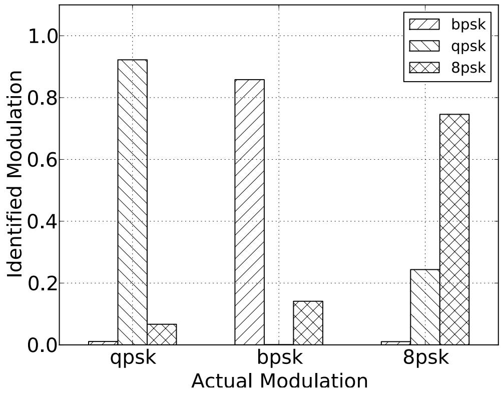


Figure 4.8: Modulation detection profile with an AMR window of 10, a SNR of 10dB and no doppler shift.

The normalization of S_k and C minimizes the errors due to the randomly varying magnitude of the received samples.

A straightforward way of recognizing PSK modulations is to use the fact that each received, distorted PSK modulation will have the smallest MSE with respect to its ideal constellation. For example, if a received sequence S_k is BPSK-modulated, then $\text{MSE}_{\text{BPSK}}(S_k)$ will be smaller than all other $\text{MSE}_C(S_k)$, $C \neq \text{BPSK}$. This is the principle employed in [83] for differentiating between PSK modulations. However, this approach does not allow us to differentiate PSK from QAM modulations accurately. QAM constellations contain significantly more points than PSK constellations, thus making it easier for a received sequence of PSK-modulated symbols to have a smaller MSE with respect to QAM than to other PSK schemes.

The decision rule for each modulation scheme is based on the *difference* between the MSE of S_k to the constellations:

$$\Gamma_{C_1,C_2}(S_k) \triangleq \text{MSE}_{C_1}(S_k) - \text{MSE}_{C_2}(S_k). \tag{4.4}$$

Fig. 4.6 shows the mean and standard deviation of the difference in MSE of the received symbols of each supported modulation scheme with respect to the ideal constellations. For every supported modulation, we transmit 320 symbols using 10 OFDM blocks of 32 subcarriers each over an AWGN channel with varying SNR levels. This is repeated 10000 times for each SNR level and the corresponding mean and standard deviation are plotted. In each figure, we use the notation "$C_1 - C_2$" to represent $\Gamma_{C_1,C_2}(S_k)$.

In the rest of this section, we will use these figures to illustrate the rationale behind the decision rules for each of BPSK, QPSK and 8PSK modulation rates.

(a) Recognizing BPSK: The decision rule used to recognize received symbols that are modulated with BPSK is

$$\Gamma_{\text{16QAM,64QAM}}(S_k) \geq \Gamma_{\text{BPSK,QPSK}}(S_k), \quad \text{and} \tag{4.5}$$

$$\Gamma_{\text{16QAM,64QAM}}(S_k) \geq \Gamma_{\text{QPSK,8PSK}}(S_k), \quad \text{and} \tag{4.6}$$

$$\Gamma_{\text{16QAM,64QAM}}(S_k) \geq \Gamma_{\text{8PSK,16QAM}}(S_k). \tag{4.7}$$

By comparing Fig. 4.6a with the other sub-figures in Fig. 4.6, one of the defining characteristics of the BPSK modulation is found to be the fact that the mean value of $\text{MSE}_{\text{16QAM}}(S_k) - \text{MSE}_{\text{64QAM}}(S_k)$ is greater than all other MSE differences at SNRs greater than 2dB. This is precisely the characteristic used in Eqs. (4.5), (4.6) and (4.7) to identify BPSK.

(b) Recognizing QPSK: The decision rule to recognize an input stream modulated with

QPSK is

$$\begin{aligned}\Gamma_{\mathrm{BPSK,QPSK}}(S_k) &\geq \Gamma_{\mathrm{16QAM,64QAM}}(S_k)\\ &\geq \Gamma_{\mathrm{QPSK,8PSK}}(S_k), \quad \text{and} \qquad (4.8)\\ \Gamma_{\mathrm{16QAM,64QAM}}(S_k) &\geq \Gamma_{\mathrm{8PSK,16QAM}}(S_k). \qquad (4.9)\end{aligned}$$

The input symbols are first matched against the BPSK decision rule and the QPSK decision rule is considered only if the BPSK decision rule does not evaluate to be true on the sequence of input symbols. Fig. 4.6b shows the differences in MSE values of a QPSK input sequence with respect to the various ideal constellations. Obviously, the ideal BPSK constellation only contains half the points of the QPSK constellation. Hence, the mean distance between the QPSK input symbols to BPSK constellation points is significantly larger than the distance to the QPSK constellation points, thus making the QPSK constellation a "better" match for the input symbols than the BPSK constellation. As a result, we now have the properties

$$\Gamma_{\mathrm{BPSK,QPSK}}(S_k) \geq \Gamma_{\mathrm{QPSK,8PSK}}(S_k), \quad \text{and} \qquad (4.10)$$

$$\Gamma_{\mathrm{BPSK,QPSK}}(S_k) \geq \Gamma_{\mathrm{16QAM,64QAM}}(S_k) \qquad (4.11)$$

that hold true for expected MSE values. Since the mean distance of the QPSK- and BPSK-modulated received symbols to the other constellations is largely similar, Eqs. (4.6) and (4.7) still hold. Hence, we obtain the QPSK decision rule by combining Eqs. (4.6), (4.7), (4.10), and (4.11).

(c) Recognizing 8PSK: The decision rule to recognize a sequence of input symbols modulated using 8PSK is

$$\begin{aligned}\Gamma_{\mathrm{QPSK,8PSK}}(S_k) &\geq \Gamma_{\mathrm{QPSK,16QAM}}(S_k), \quad \text{and} \qquad (4.12)\\ \Gamma_{\mathrm{QPSK,16QAM}}(S_k) &\geq \Gamma_{\mathrm{16QAM,64QAM}}(S_k), \quad \text{and} \qquad (4.13)\\ \Gamma_{\mathrm{QPSK,16QAM}}(S_k) &< 0, \quad \text{and} \qquad (4.14)\\ |\Gamma_{\mathrm{8PSK,16QAM}}(S_k)| &\geq \alpha, \quad \text{and} \qquad (4.15)\\ |\Gamma_{\mathrm{QPSK,8PSK}}(S_k) - \Gamma_{\mathrm{16QAM,64QAM}}(S_k)| &\geq \beta. \qquad (4.16)\end{aligned}$$

The 8PSK decision rule is used after both the BPSK and QPSK decision rules have been evaluated to be false on the input symbols. Hence, the 8PSK decision rule only needs to differentiate 8PSK from 16QAM and 64QAM constellations. It is obvious from Figs. 4.6c, 4.6d and 4.6e that at SNRs greater than 6dB, Eqs. (4.12)–(4.14) represent the key characteristics of the mean MSE differences that distinguish 8PSK from 16QAM and 64QAM. However, we also observe that with a 16QAM-modulated input sequence (Fig. 4.6d), at SNRs greater than 18dB, the mean values of $\mathrm{MSE}_{\mathrm{QPSK}}(S) - \mathrm{MSE}_{\mathrm{8PSK}}(S_k)$, $\mathrm{MSE}_{\mathrm{QPSK}}(S_k) - \mathrm{MSE}_{\mathrm{16QAM}}(S_k)$ and

$\mathrm{MSE}_{\mathrm{16QAM}}(S_k) - \mathrm{MSE}_{\mathrm{64QAM}}(S_k)$ are close enough such that Eqs. (4.12)–(4.14) will hold true for a significant proportion of the actual MSE difference values, thus increasing the probability that 16QAM will be mis-recognized as 8PSK. To prevent this, Eqs. (4.15) and (4.16) ensure that these MSE differences must not be "too close" in order for the 8PSK modulation to be correctly identified, with the degree of closeness to be defined by the parameters α and β. In our evaluation, we have found that $\alpha = \beta = 0.03$ gives the highest accuracy in differentiating 8PSK from QAM constellations.

4.3.3 What is the appropriate size of N?

The variance of the MSE and the corresponding accuracy of Aileron depends on the length (N) of the input sequence S_k—AMR accuracy improves with longer input sequences but at the cost of a longer recognition delay.

The *AMR window* refers to the number of OFDM symbols used by each iteration of the AMR algorithm. This directly affects the length (N) of the sequence of input symbols S_k to the MSE equation (4.3). With active-mode Aileron, since every signaling subcarrier can use a different modulation scheme, an AMR window of length N (i.e., N OFDM blocks) will only produce N input symbols from a single subcarrier position. On the other hand, with passive-mode Aileron, all the data subcarriers use the same modulation scheme, so an AMR window of length N will contain $N \cdot K$ input symbols, where K is the number of data subcarriers per OFDM symbol. Our evaluation of Aileron will study the effects of the AMR window length on its accuracy.

4.4 Evaluation Using Simulated Channels

In this section, we evaluate the accuracy of Aileron under a wide range of simulated channel conditions. Some of these conditions, such as the doppler frequency seen at 100m/s, cannot be easily created on a testbed. Thus, we use simulated channels to conduct a thorough evaluation of Aileron.

PHY Parameter	Value
Center frequency	2.4GHz
Total bandwidth	12.5MHz
Total subcarriers	1024
Cyclic prefix length	256
No. of subchannels	16
No. of subcarriers per subchannel	64
No. of active-mode signaling subcarriers per subchannel	6
No. of guard subcarriers per subchannel	32

Table 4.1: Parameters used in the OFDMA PHY.

4.4.1 Experimental setup

We implemented Aileron using an OFDMA PHY in GNURadio with the parameters listed in Table 4.1. We assume that Aileron is used in conjunction with a MAC protocol to coordinate channel access between transmitters. Hence, a single transmitter–receiver pair is sufficient to understand the performance of Aileron. The transmitted samples are filtered using a simulated channel in MATLAB, using the parameters in Table 4.2, before being passed to the receiver.

Aileron is evaluated using the following JTC [84] channel models in MATLAB: `jtcInResC`, `jtcInOffC`, `jtcInComC`, and `jtcOutUrbHRLAC` that correspond to Indoor residential C, Indoor office C, Indoor commercial C, and Outdoor urban high-rise areas–Low antenna C, respectively. Note that the variation of the doppler frequency from 0 to 800Hz in 80Hz increments correspond to movement speeds of 0 to 100m/s in increments of 10m/s at a center frequency 2.4GHz. The set of chosen channel models, doppler frequencies and SNRs represent a wide range of possible channel conditions under which the AMR algorithm has to operate. The SNR of the channel is representative of the interference seen on the channel. Due to space limitation, we will only present the evaluation results obtained using the `jtcInOffC` channel. The performances of Aileron under the other channel models are very similar.

4.4.2 Aileron accuracy in static environments

Active-mode Aileron accuracy under different SNRs. Fig. 4.6 shows the accuracy of active-mode Aileron over channels without mobility: symbols are sent over the fading channel with no doppler shift, which is representative of a typical indoor office WLAN environment. This accuracy of Aileron is computed over 50000 AMR windows of 10 OFDM symbols each.

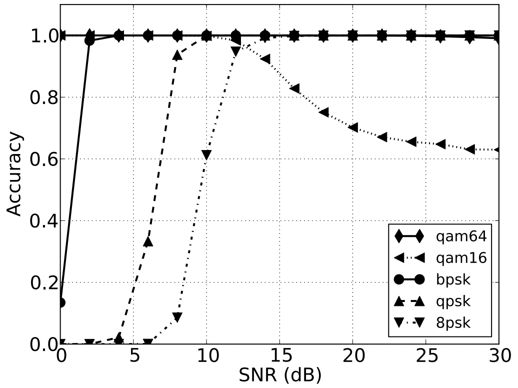


Figure 4.9: Passive-mode Aileron accuracy in a simulated channel with no doppler shift and an AMR window of size 10.

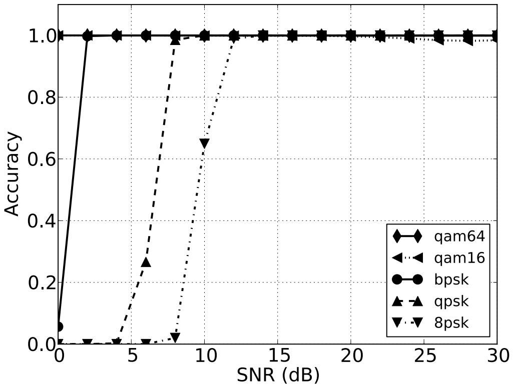


Figure 4.10: Passive-mode Aileron accuracy in a simulated channel with no doppler shift and an AMR window of 20.

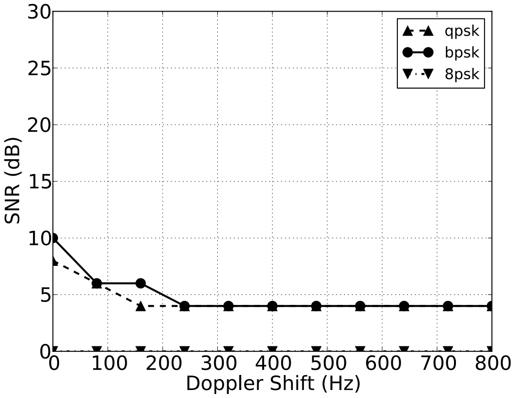


Figure 4.11: Lowest SNR level at which the accuracy of active-mode Aileron exceeds 90%, using an AMR window of 50.

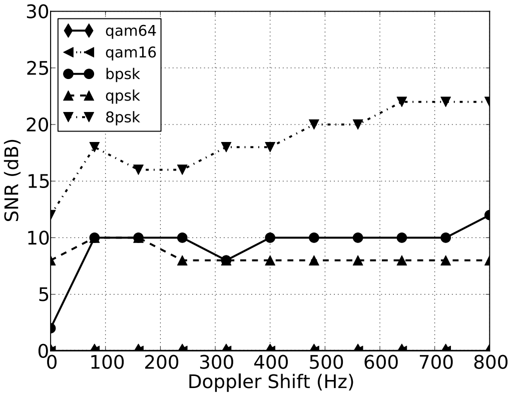


Figure 4.12: Lowest SNR level at which accuracy of active-mode Aileron exceeds 90%, using an AMR window of 10.

Emulation Parameter	Value
Channel Model	`jtcInResC`, `jtcInOffC`, `jtcInComC` and `jtcOutUrbHRLAC`
Doppler Frequency	0 - 800Hz in 80Hz increments
Signal-to-Noise Ratio	0 - 30dB in 2dB increments
Modulation Rates	BPSK, QPSK, 8PSK, 16QAM and 64QAM
AMR Window	10, 15, 20, 25, 50, 75 and 100

Table 4.2: Parameters used in the simulated channels. The names of the channel model correspond to those used by MATLAB.

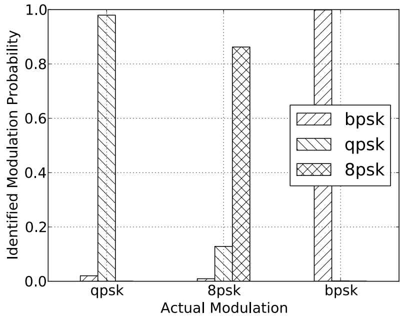


Figure 4.13: Active-mode Aileron accuracy over the good-quality channel.

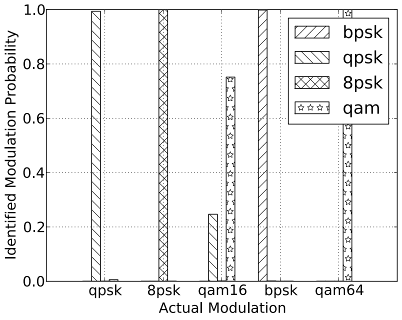


Figure 4.14: Passive-mode Aileron accuracy over the good-quality channel.

Aileron is shown to be able to recognize BPSK and QPSK modulations with practically perfect accuracy at SNR above 16dB. 8PSK is detected correctly approximately 79% of the time at all SNR levels. It must be stressed that this level of accuracy is achieved using only *10 received symbols*. As expected, the active-mode Aileron detection accuracy improves as we increase the number of symbols used by the AMR.

Active-mode Aileron accuracy under different AMR window sizes. Fig. 4.7 shows the AMR accuracy of active-mode Aileron when channel SNR and doppler shift are fixed at 8dB and 0Hz, respectively. BPSK and QPSK modulations are recognized with 99% accuracy with 25 received symbols while 75 received symbols are required to achieve the same accuracy with 8PSK. This trend—where 8PSK is recognized less accurately than BPSK and QPSK, given the same number of received symbols—persists even at higher SNR levels.

Active-mode (mis)detection performance Fig. 4.8 shows the detection probability of the all the possible modulation schemes that can be used in active-mode Aileron. BPSK and QPSK can be easily distinguished from each other but when the received symbols are modulated using 8PSK, approximately 22% of the symbols are mis-recognized as QPSK. This error is due to the increased variance in the MSE differences used by the detection rules that is brought about by the multipath fading channel.

Passive-mode Aileron accuracy. Fig. 4.9 shows the accuracy of passive-mode Aileron when applied to data subcarriers from a single OFDMA subchannel. Since there are 32 data subcarriers in each OFDMA subchannel, 10 OFDM blocks will give 320 data symbols—significantly more than that obtained from the active-mode Aileron. The larger number of received data symbols increases the accuracy of Aileron: BPSK modulation is recognized with accuracy 100% of the time at SNRs greater than 2dB while perfect identification of QPSK and 8PSK occurs at SNRs above 10dB and 16dB, respectively. The AMR algorithm can always differentiate between the PSK modulations: mis-identified QPSK and 8PSK modulations are always labeled as QAM, rather than another PSK scheme.

For the QAM schemes, 64QAM is accurately identified at all SNR levels while 16QAM is correctly identified only up to 12dB, above which the recognition accuracy of 16QAM encounters a significant drop as it is consistently mis-identified as QPSK. This is because at higher SNRs, the mean value of $\mathrm{MSE}_{\mathrm{QPSK}}(S) - \mathrm{MSE}_{\mathrm{8PSK}}(S)$, $\mathrm{MSE}_{\mathrm{QPSK}}(S) - \mathrm{MSE}_{\mathrm{16QAM}}(S)$ and $\mathrm{MSE}_{\mathrm{16QAM}}(S) - \mathrm{MSE}_{\mathrm{64QAM}}(S)$ of a 16QAM-modulated input converges, as seen in Fig. 4.6d. With an AMR window size of 10 OFDM blocks, the variance of MSE differences is large enough for 16QAM to be mistaken for QPSK with a high probability. If we double the input AMR window size to 20 blocks, 16QAM will be identified with perfect accuracy, as shown in Fig. 4.10.

4.4.3 Aileron accuracy in mobile environments

Mobility in wireless networks is characterized by the presence of doppler shift in transmissions over the channel. The comparative performance of Aileron with respect to the different input modulations in a mobile environment is similar to that described in Section 4.4.2, albeit with different accuracy values.

Fig. 4.11 shows the lowest SNR at which active-mode Aileron can achieve 90% accuracy for BPSK, QPSK and 8PSK modulations under different mobility speeds. The accuracy of Aileron is computed using 50000 AMR windows, each with a length of 50. In this environment, BPSK and QPSK modulations can be correctly identified 90% of the time at SNR greater than 10dB and 7dB, respectively, while greater than 90% accuracy in recognizing 8PSK is achieved for all the considered SNR levels and doppler shifts.

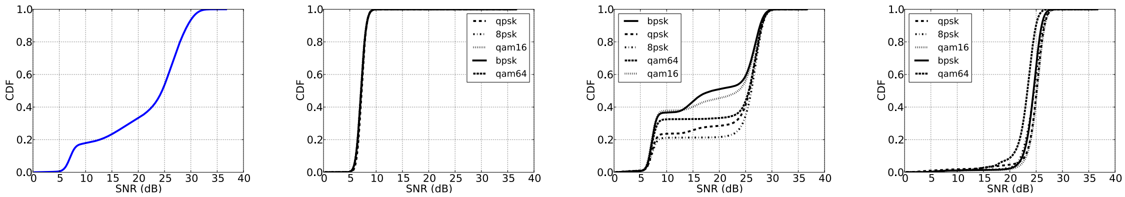
Fig. 4.12 shows the results of minimum SNR at which passive mode Aileron can achieve 90% accuracy. We use an AMR window size of 10. At SNR greater than 12dB, BPSK and QPSK can be correctly recognized with 90%, while at 22dB SNR and above, 8PSK can be recognized with 90% accuracy with a doppler frequency of up to 800Hz.

4.5 Evaluation Using Real Channels

4.5.1 Experimental setup

We evaluate the accuracy of modulation-based signaling using USRP2 devices deployed over 8 locations of a single floor of an academic department. The GNURadio implementation of modulation-based signaling from Section 4.4 with the parameters in Table 4.1 is used in these experiments. A trace collection proceeds as follows. A transmitter is placed at one of the 8 locations and it transmits approximately 10000 frames using 5 randomly-selected OFDMA subchannels. All five modulation rates are simultaneously used to transmit a frame. The nodes placed at the other 7 locations receive and decode this transmission. Each transmitter repeats the 10000-frame transmission 10 times, with a different set of 5 subchannels selected each time.

This collection procedure is performed at each of the 8 node positions to collect a total of approximately 100 million OFDMA blocks. Since the traces are collected during normal working hours, the recorded channel conditions include environmental mobility effects due to the movements of people around the office floor. In the rest of this section, we will present the accuracy of Aileron based on these traces with an AMR window of size 10.



(a) All channels **(b)** Low quality channel **(c)** Intermediate quality channel **(d)** High quality channel

Figure 4.15: SNR of channels encountered during experimental evaluations with the USRP

4.5.2 Channel SNR characteristics

SNR for each subchannel is computed only for every AMR window using only the PSK-modulated subcarriers as these have a known transmission magnitude. The received signal energy is estimated using the mean magnitude of the received PSK symbols while the noise power is estimated using the variance of this magnitude over the AMR window. The ratio of this estimated signal-to-noise power is the SNR of the subchannel and is presented here in decibels (dB).

Fig. 4.15a shows that the distribution of the overall SNR of all non-overlapping AMR windows across all point-to-point links varies over a wide range, from 5dB to 32dB. 18% of the AMR windows have SNRs between 5 and 7dB while 60% have SNRs between 23 and 32dB. The remaining 22% of the AMR windows have SNRs between 7 and 23dB. The SNR distribution of each link can differ significantly from that shown in Fig. 4.15a. To illustrate the performance of Aileron across a wide range of channel conditions, we focus on traces from three channels with distinctly different SNR distributions: poor, intermediate and good quality channels. The SNR distributions of these three channels are plotted in Figs. 4.15b, 4.15c and 4.15d, respectively.

4.5.3 Aileron Performance under varying SNR

Under the high-SNR channel, active-mode Aileronis very accurate, as shown in Fig. 4.13. BPSK, QPSK and 8PSK are correctly recognized with a probability of 100%, 98% and 86%, respectively. This matches the performance of Aileronunder a simulated channel, as shown in Fig. 4.6. Note that this level of accuracy is achieved using an AMR window size of 10 under realistic conditions with environmental mobility. This shows that in high-SNR channels, modulation-based signaling with Aileron is reliable and feasible for low rate coordination purposes.

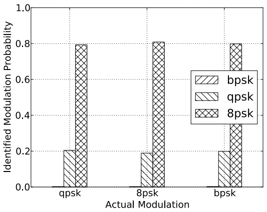


Figure 4.16: Active-mode Aileron accuracy over the poor-quality channel.

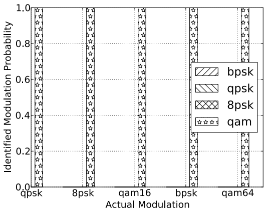


Figure 4.17: Passive-mode Aileron accuracy over the poor-quality channel.

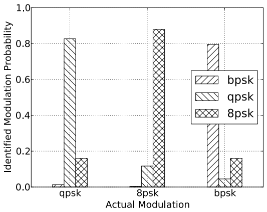


Figure 4.18: Active-mode Aileron accuracy over the intermediate-quality channel.

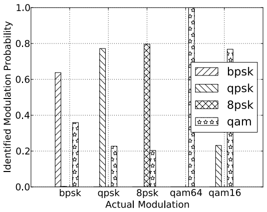


Figure 4.19: Passive-mode Aileron accuracy over the intermediate-quality channel.

Fig. 4.14 shows the performance of passive-mode Aileron with data subcarriers in the good channel. For each modulation rate transmitted over this subchannel, we plot the probability of it being detected as "BPSK", "QPSK", "8PSK" or "QAM" by the AMR as described in Algorithm 7. Under the high-SNR channel, BPSK, QPSK, 8PSK and 64QAM modulation schemes in the data subcarriers are detected with 100% accuracy. 16QAM, on the other hand, is only correctly identified 75% of the time. Again, this matches the results obtained using the emulated channel as shown in Fig. 4.9.

The SNR of the poor-quality channel varies between 5 and 9dB. At such low SNRs, both active and passive mode Aileron have low accuracy, as shown in Figs. 4.16 and 4.17. This is consistent with the results in Figs. 4.6 and 4.9 that are obtained over the simulated channel. However, note that passive-mode Aileron is not confused between the different decision rules and returns the default "QAM" result in every case where it cannot correctly identify the modulation scheme used.

With an intermediate quality channel, we can see from Fig. 4.15c that up to 38% of the SNR values are below 10dB while at least 40% of the SNR experienced is above 25dB. Under such mixed conditions, active-mode Aileron can correctly recognize BPSK, QPSK, and 8PSK with 80%, 82% and 89% of the time, respectively. This demonstrates that modulation-based signaling is reliable over channels that experience highly variable SNR.

Passive-mode Aileron can also accurately determine the modulation rate in data subcarriers, as shown in Fig. 4.19. Notice that with data subcarriers, similar to the case of low-SNR channels, no PSK scheme is confused as another.

4.6 Discussion

4.6.1 Increasing detection accuracy

Both our simulated and real-world experiments are designed to closely match the capabilities of our USRP configuration. As a result, all signals are processed at the Nyquist rate. In practical implementations of Aileron, *oversampling* can be used to improve its detection accuracy significantly. An oversampling factor of k means that the data frame is received at k times its Nyquist bandwidth.

Fig. 4.21 shows how the root-mean-squared Error Vector Magnitude (EVM) of symbols in 20MHz 802.11a frames varies when different oversampling factors are used. At each modulation rate, the EVM is computed over 10000 802.11a frames that are transmitted over a `jtcInOffC` channel.

The EVM of the received signals decreases with increasing oversampling factors and we can expect a similar detection improvement in Aileron with oversampling. Oversampling is a technique widely employed by commercial wireless devices and can thus be easily integrated into Aileron.

4.6.2 Rate-delay tradeoff

Aileron is used to concurrently send control information to receivers that are otherwise unable to decode the primary transmission. For example, an AP in a multi-channel WLAN can concurrently send ACK and data frames to two WLAN clients that are on different channels. However, encoding information using modulation rates can cause the data frame to be transmitted at a sub-optimal rate. Even so, this data-rate reduction compensated by a significant reduction in the network coordination overhead due to the seamless exchange of Aileron control frames. In the multi-channel WLAN scenario, the median channel switching delay of 15ms [9] is an order of magnitude larger than the data transmission time (less than $1ms$ at 54Mbps). This delay constitutes a significant overhead in typical multi-channel transmissions, especially with short packets such as ACKS. Aileron eliminates this coordination overhead when short control frames have to be sent to out-of-band receivers. We believe that

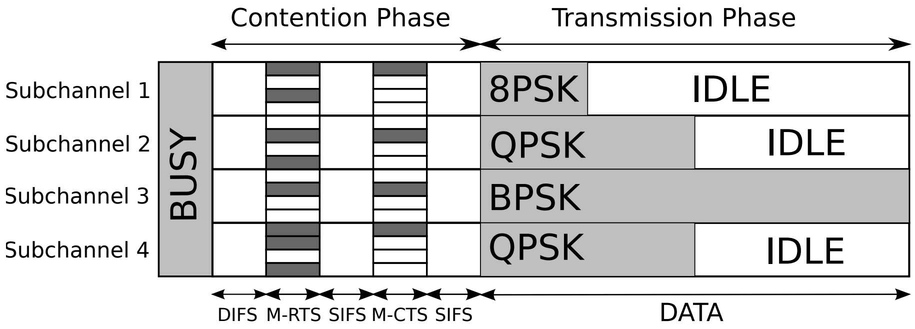


Figure 4.20: Example channel utilization without Aileron.

this presents a beneficial rate-delay tradeoff when dealing with challenging wireless networks, such as multi-channel and cognitive radio networks.

4.6.3 Fading channels

Modulation identification in Aileron is conducted over a window of OFDM symbols and the window size can be extended to neutralize the effects of channels with particularly long fading durations. Our choice of a 10-symbol window size is based on real-world measurements and has been shown to offer good performance over actual real-world fading channels.

Bit interleaving and channel coding are typically used to increase the resilience of 802.11 frames to the effects of channel fades. Such techniques are orthogonal to Aileron, which employs a predominantly PHY layer signaling mechanism. Cross-layer integration of these techniques, though possible, are beyond the scope of this work.

4.7 Use Cases

To demonstrate its utility, we apply Aileron for (1) improvement of channel utilization and (2) efficient handling of acknowledgements. These two uses cases and their evaluation are detailed next.

4.7.1 Improvement of Channel Utilization

Channelization of a wideband spectrum [7, 30] is a well-known approach to improving the utilization of a wireless channel. FICA [7] is an example PHY that adopts channelization and frequency domain contention [85, 86, 87] to improve wireless channel utilization. However, a key limitation of FICA comes from the fact that after each contention round, only a fixed, predefined number of OFDM symbols can be transmitted on each subchannel. This is to ensure that each wireless node occupies a constant amount of airtime, regardless of the modulation rate used. Obviously, this approach can be limiting for traffic that is bursty or consists of a large range of frame size, as is the case with interactive web traffic and multimedia streaming applications.

We note that frame aggregation is orthogonal, yet complementary to channelization. Channelization increases the number of concurrent transmitters, but the bandwidth available to each transmitter at any time is stochastic in nature. On the other hand, frame aggregation gives the transmitter the flexibility to maximize the use of its available bandwidth.

In this section, we demonstrate how Aileron can be used to replace the fixed transmission portion of FICA with one that allows each node to transmit a variable number of frames. We dynamically determine the number of aggregated frames to be transmitted by each node from the relative modulation rates of the other concurrent transmitters, but without any explicit coordination between any pair of nodes. This mechanism is simple but can be easily extended to encompass more complex aggregation protocols. We leave such exploration as future work.

Protocol description

FICA divides the wireless channel into multiple non-overlapping subchannels. Each subchannel has a set of subcarriers, known as the *contention band*, that is used for channel contention.

Actual channel use is separated into the contention phase and the transmission phase, and progress from one phase to the other is time-synchronized across all clients. When the entire channel is sensed to be idle for a length of time equal to the DIFS, each client sends a frequency-domain Binary Amplitude Modulation (BAM) signal on a randomly-selected contention band. The AP then waits for a further SIFS-specified duration before picking a winning subcarrier in each contention band. It sends a BAM ACK signal on the winning contention bands and the clients associated with those bands then proceed with data transmissions.

Two key observations can be made here. First, during channel contention, the AP does not know the ID of any contending client. Second, at the end of the channel contention, each client only knows if it has won access to its selected channels. Clients do not know the winner of any other non-selected subchannel or of any selected channel that it fails to win access to.

Before describing our extension to FICA, we make the following assumptions. First, a fixed number of subcarriers, known as *control subcarriers*, at known positions in each subchannel are used by active-mode Aileron to encode the address of the transmitting client. Second, the modulation rate of the remaining subcarriers are selected by a rate-control algorithm.

Third, all data frames are of the same length 1.5KB, which is typically the case for bulk data transfer scenarios. Finally, different subchannels can use different modulation rates, but all data subcarriers in the same subchannel must use the same modulation rate.

Under these assumptions, if the capability to transmit multiple frames is not available, the channel utilization will resemble the illustration in Fig. 4.20: the "good" quality subchannels that can transmit frames at higher bit-rates will suffer from lower utilization. With Aileron, clients can opportunistically transmit additional frames during these idle periods while maintaining the high channel utilization of FICA. We combine the channel contention phase of FICA with a transmit scheduling algorithm, shown in Algorithm 8, that uses Aileron.

Algorithm 8: Search for transmission opportunities.

```
Input: $\mathcal{N}$ is the set of all active nodes in the current transmission phase; $\mathcal{C}$ is the set of all
       subchannels; $P$ is the size of each transmitted frame; $k$ is the ID of the node executing
       this search algorithm; $C_n$ is the set of channels assigned to node $n \in \mathcal{N} \setminus \{k\}$; $R_c$ is
       the transmission rate of each channel $c \in \mathcal{C}$; $M_k$ is the total number of frames sent in
       the current transmission phase.
begin
    $T[k] \leftarrow P / \sum_{c \in C_k} R_c$;
    for $n \in \mathcal{N} \setminus \{k\}$ do
        $r \leftarrow \sum_{c \in C_n} R_c$;
        $T[n] \leftarrow P/r$;
    end
    $m \leftarrow \max_{n \in \mathcal{N} \setminus \{k\}} T[n]$;
    if $m - T[k] \cdot (M_k + 1) \geq T[k]$ then
        $M_k \leftarrow M_k + 1$;
        Schedule another frame for transmission;
    else
        $M_k \leftarrow 0$;
        Wait for the next contention phase;
    end
end
```

Let $\mathcal{N}$ be the set of Aileron clients and C_n be the channels assigned to each client $n \in \mathcal{N}$ for the current data transmission phase. During the data transmission phase, each node encodes its ID in the predefined subcarriers within its assigned subchannels. When a node k completes its transmission, it enters the idle state. It listens for N OFDM blocks on each subchannel and uses passive Aileron to determine the modulation rate of each subchannel. The node k also determines the set of channels in use by each neighbor, C_n for $n \in \mathcal{N} \setminus \{k\}$, from the IDs encoded in the control subcarriers. Full duplex wireless communications [88] can be used to collect these N OFDM blocks concurrently with the transmission to minimize the overhead of Aileron.

With these two pieces of information, the node can determine the transmission time required by each neighbor and the remaining transmission duration of the slowest node. Note that the transmission time in use by a node depends on both the rate used in each of its subchannels and the total number of subchannels assigned to it. Let M_k be the total number of frames sent by node k in the current transmission phase. If the channel occupancy of the slowest node is greater than the time required for node k to transmit $M_k + 1$ frames, then an additional frame is sent within this remaining duration using its assigned subchannels. Otherwise, it simply waits for the next transmission round.

Simulation setup

We demonstrate the improvements achieved by Aileron in FICA using a custom simulator that models the Aileron performance in detail. In our simulation, we evaluate Aileron using the same PHY and channel parameters as shown in Tables 4.1 and 4.2, except that we limit the size of the AMR window to 10, 15, 20 and 25 blocks. FICA utilizes two frequency backoff policies, AIMD and RMAX, but we only present results that use RMAX as it has been shown in [7] to outperform AIMD. All the SNR values between every pair of nodes and between each node on the AP are governed by identical and independently distributed random variables that follow the distribution shown in Fig. 4.15a. The modulation detection accuracy at various SNR and doppler shift values follow the simulated results described in Section 4.4.

Each simulation run consists of a single AP and 10 contending FICA clients. We run the simulation for 1000000 time units, where a single time unit is equivalent to the transmission time of a single OFDM block. The results shown here are obtained from 20 simulation repetitions.

In our evaluation, we do not explicitly model the effects of an auto-rate algorithm. Instead, given the SNR of the channel, we simply pick the highest modulation rate from the known bit error rate (BER) graph [89] that can meet a maximum BER of 10^{-4}.

The three performance metrics that we use are:

M1. Per-Node Channel Utilization. This is the ratio of the total transmission time of a node during a single transmission phase to the duration of the entire transmission phase. The duration of the transmission phase is lower-bounded by the slowest transmitting rate among all the active nodes.

M2. Airtime Fairness. We use the Jain's fairness index to determine how the channel is shared among the competing nodes. Since the channel access time of every node is affected by its utilization, this essentially illustrates how the channel utilization varies across the Aileron nodes.

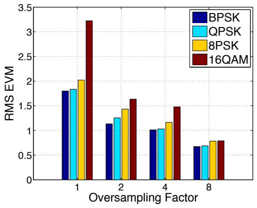
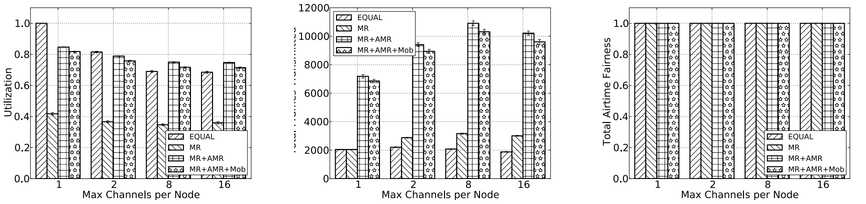


Figure 4.21: EVM of symbols in a 20MHz 802.11a frame at different modulation rates



(a) Channel utilization **(b)** Total number of frames transmitted **(c)** Throughput fairness

Figure 4.22: Mean and standard deviation of the three evaluation metrics. The mean is represented by the height of the bar while the error bars indicate the standard deviation

M3. Total Frames Transmitted. This is a simple count of the total number of frames that are transmitted over the duration of the simulation and is a measure of the throughput.

The four scenarios considered in our simulations are:

S1. Equal rate (EQUAL). All clients transmit with the same modulation rate during each transmission phase. This rate is chosen such that a BER of at most 10^{-4} is achieved on the channel with the lowest SNR.

S2. Multi rate (MR). During each transmission phase, the highest modulation rate on each channel, with respect to the SNR, that can achieve a BER of at most 10^{-4} is chosen.

S3. Multi rate with AMR (MR+AMR). This is similar to MR, except that Aileron is now used to find transmission opportunities for nodes with high transmission rates.

S4. Multi rate with AMR and mobility (MR + AMR + Mob). This is MR+AMR with the addition of mobile nodes. Node velocities are randomly assigned and are characterized by the presence of doppler shift in the channel.

Aileron is not used in EQUAL and MR. Hence, only one frame is sent in each transmission opportunity in EQUAL and MR.

Simulation results

For brevity, we only show the results obtained with an AMR window of 10 since the results obtained with larger AMR window sizes show similar behavior. Fig. 4.22a shows the mean and standard deviation of the channel utilization of EQUAL, MR, AMR+MR and AMR+MR+Mob with different numbers of maximum channels per node.

Observe that EQUAL with only one channel per node achieves maximum utilization of the channel, since all frames are transmitted at the same rate and the channel is never idle. However, when each transmitter under the EQUAL scenario is allowed to contend for more than one channel, channel utilization drops from 82% with up to 2 channels per node, to

70% when each node can contend for all 16 channels. Varying the number of channels per node effectively varies the throughput by each node. The resulting idle periods belonging to nodes with high throughput reduces overall channel utilization. This effect of heterogeneous throughput on channel utilization is even more dramatic when the modulation rates of different subchannels are allowed to vary under the MR scenario—mean channel utilization drops to under 40%, regardless of the number of allowable channels per node.

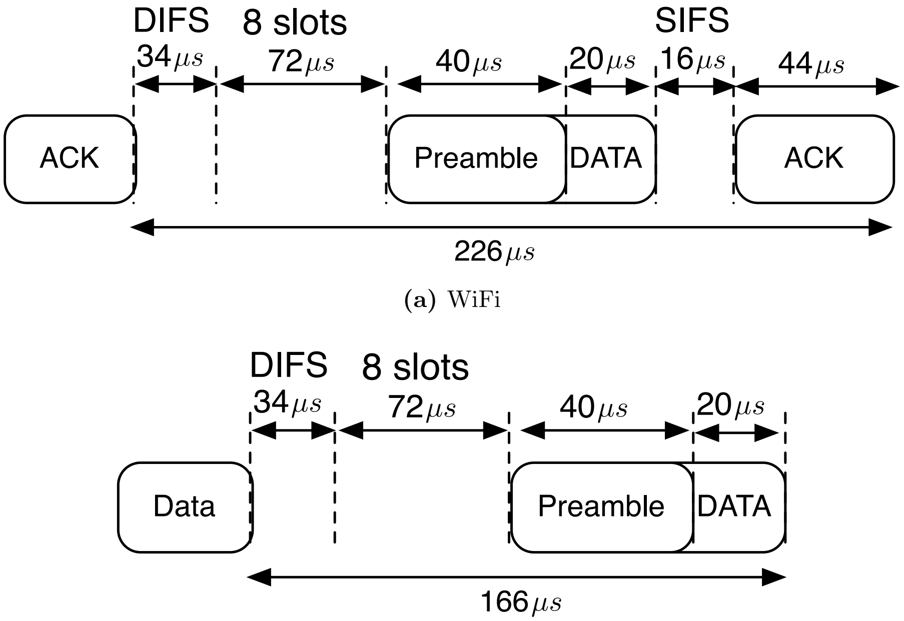
Aileron can improve the channel utilization by opportunistically sending a frame if sufficient time remains before the slowest node completes its transmission. With up to 2 channels per node in the MR+AMR scenario, Aileron can achieve 79% channel utilization. When each node can contend for all channels, Aileron achieves 76% channel utilization, which is above that achieved in the EQUAL scenario. This significant improvement in channel utilization is present even with node mobility.

Besides the improvement in channel utilization, Aileron also increases the mean throughput of each node, as shown by the count of transmitted shown in Fig. 4.22b. When all nodes are limited to only one channel, there is no throughput difference between the EQUAL and MR scenarios since high throughput nodes in MR are still limited by the low throughput nodes. When the number of allowable channels increases, nodes in the MR scenario have a higher throughput than those in the EQUAL scenario. This reflects the advantage of a per-channel modulation rate adaptation. Aileron is able to significantly increase the achievable throughput via appropriate opportunistic transmissions. When the clients can contend for up to 8 channels, almost 11000 frames are transmitted on average using Aileron while only 2000 and 3000 frames are transmitted in the EQUAL and MR scenarios, respectively. This throughput increase achieved by Aileron does not come at the expense of throughput fairness among the transmitting nodes, as shown in Fig. 4.22c.

4.7.2 Efficient Handling of Wireless ACKS

The rising popularity of high bandwidth interactive streaming video (such as Skype video chats and Google Hangouts) increases the importance of efficiently using the available spectrum. However, it is well known that simply increasing the bandwidth of 802.11 wireless networks actually *decreases* their efficiency [7, 90] due to the high protocol overhead, of which wireless ACKs make up a significant portion.

Fig. 4.23a shows the breakdown of the delays incurred when transmitting a 802.11 frame at 600Mbps [90]. In addition to transmitting the actual data, a successful frame transmission also requires a DIFS, a backoff (of 8 slots in this case), a PHY layer preamble, an SIFS after the data transmission and the accompanying ACK frame. Observe that at 600Mbps, the



(a) WiFi

(b) Aileron

Figure 4.23: Time required to transmit a WiFi frame at 600Mbps

ACK and associated SIFS make up for almost 27% of the overall frame transmission time.

Using Aileron, we can eliminate the ACK delays by sending acknowledgements simultaneously with the data transmission. Fig. 4.23b illustrates how the transmission delay, at 600Mbps data rate, can be reduced by almost 38% with Aileron.

Encoding packet IDs with Aileron

For the sake of clarity, we will explain the frame exchange scheme using two Aileron devices, Alice and Bob. The ACK message only needs one bit of information to acknowledge a successful frame transmission. Let N_{ack} be the number of Aileron-subcarriers used for encoding the ACK. We select two unique ternary numbers a_1 and a_2 from $\{0, \ldots, 3^{N_{ack}}\}$, to indicate a successful and an unsuccessful frame transmission respectively. a_1 and a_2 must be selected such at the hamming distance between then is maximized. This minimizes the chance that a_1 is misidentified as a_2 and vice versa.

Assume that Bob transmits a frame to Alice. If Alice can correctly decode Bob's transmission, she sends encodes a_1 with Aileron into her next frame and transmit it to Bob. Otherwise, if Alice either fails to decode Bob's frame or a timeout occurs, she encodes a_2 into her next frame and sends it to Bob. Bob can now take one of four possible actions: (a) if he correctly receives the frame from Alice and recovers a_1 using Aileron, he simply encodes

a_1 into the next frame in the queue and transmits it to Alice; (b) if he correctly receives the frame from Alice and recovers a_2, he encodes a_1 into the current frame and retransmits it to Alice; (c) if Bob fails to decode the frame from Alice or a timeout occurs, he encodes a_2 into the current frame and retransmits it to Alice; (d) if Bob can correctly decode the frame from Alice but cannot decode the ACK message, he retransmits the current frame with a_1 as the ACK message. Note that we opt to be conservative with (d) since Bob does not know if Alice successfully received his previous transmission.

Efficiency improvement

Let the time taken for the WiFi transmission to be

$$t_{\text{wifi}} = t_{\text{difs}} + W \cdot t_{\text{slot}} + t_{\text{preamble}} + t_{\text{data}} + t_{\text{sifs}} + t_{\text{ack}} \tag{4.17}$$

where W is the number of slots used for contention resolution. Fig. 4.23 shows the transmission time when $W = 8$. Similarly, the transmission time when using Aileron is

$$t_{\text{Aileron}} = t_{\text{difs}} + W \cdot t_{\text{slot}} + t_{\text{preamble}} + t_{\text{data}} \tag{4.18}$$

Let ν be the probability at which the Aileron-encoded ACK is decoded incorrectly. An additional transission will occur if the ACK message cannot be decoded correctly to either a_1 or a_2. The efficiency improvement due to Aileron is thus

$$\eta = \frac{t_{\text{wifi}} - (1+\nu) \cdot t_{\text{Aileron}}}{t_{\text{wifi}}} \tag{4.19}$$

$$= 1 - \frac{(1+\nu) \cdot t_{\text{Aileron}}}{t_{\text{wifi}}} \tag{4.20}$$

Fig. 4.24 shows the gains that can be achieved using Aileron for inband ACKS. If the ACK message can be received with no Aileron error, we can obtain up to 28% reduction in the overall transmission time. Even at 10 and 20% decoding error, Aileron still saves approximately 20% and 13% of the transmission time, respectively.

4.8 Related Work

Control Channel Design. Typical control channels can be classified to be in-band or out-of-band. In-band control channels carry control frames in the same channel as that used for data frames. Examples include in-band medium access control using CSMA [91] and slotted ALOHA [92]; probe frames for auto-rate selection [78]; link-quality measurements in mesh

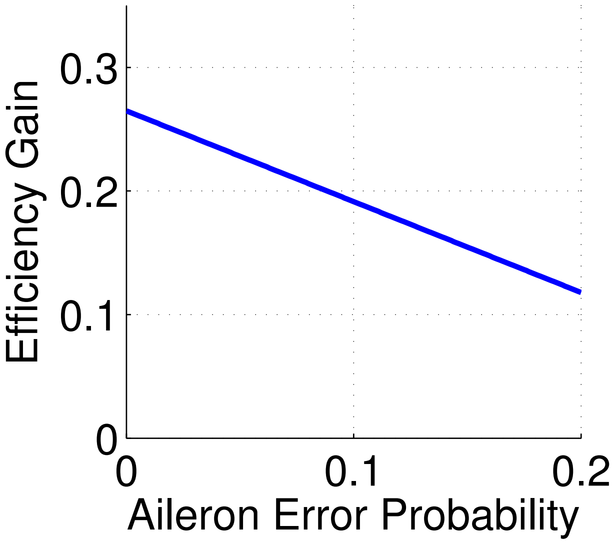


Figure 4.24: Spectrum efficiency due to inband ACKs

networks [93]; transmitting control frames using side-channels [94] and inter-frame gaps [95]. SMACK [96] extends the in-band control to the PHY layer through its use of on-off OFDM subcarrier signaling for sending acknowledgements. Out-of-band approaches are characterized by the use of a dedicated channel for control frames. If only one wireless interface is available [28], the need for it to be switched between the control and data channels incurs a significant coordination overhead. If multiple interfaces are available [97], the coordination overhead is reduced at the cost of higher hardware and power requirements.

Modulation Recognition. The method of modulation recognition in [83] is based on the differences of MSE, but its recognition algorithm is too simplistic to be able to differentiate PSK from QAM modulations. Other recognition methods include the use of higher-order statistics [98], wavelet transform [99], and cyclic features of the digital transmission [100].

Chapter 5

Spectrum Aggregation

5.1 Introduction

The proliferation of unplanned high bandwidth 802.11a/g/n APs in urban areas offers the potential for WLANs to be strong complement to cellular networks in providing ubiquitous connectivity [101, 102, 103]. However, this potential has to be tempered by the fact that the APs (a) are deployed chaotically and are not under any centralized control, (b) are connected to broadband backhaul links with bandwidths that are significantly lower than that of the WLAN channels, and (c) can be cellular 4G routers where the backhaul link, being an LTE or WiMAX channel, is subject to the usual vagaries of wireless networks. For example, 802.11n can achieve a throughput of at least 300Mbps [16], which is typically an order-of-magnitude higher than that of broadband backhaul networks.

Wireless clients can overcome this limitation by aggregating backhaul links from multiple APs [104, 105]. In such a protocol, a WLAN client connects to multiple APs, one at a time, with the order and duration of each connection determined by the parameters—such as bandwidth, queue length, congestion, etc.—of both the backhaul and the WLAN channel. However, two significant obstacles stand in the way of the efficient scheduling of connectivity across multiple APs with only one WLAN interface on the client. First, the client node can typically only communicate with one AP at a time. This gives rise to an obvious chicken-and-egg conflict: the client needs to know the available bandwidth from an AP before it can construct a connection schedule, but it can only know the available bandwidth after it has connected to the AP and measured or downloaded traffic statistics. Second, the time-varying nature of traffic on both the wireless and the backhaul links means that an aggregating client who only obtains bandwidth information after its AP association will never be able to track the bandwidth variation accurately and thus, cannot adjust its connection schedule to maximize the achievable backhaul throughput. Figure 5.1 illustrates the number of bytes downloaded by a static Bittorrent client in consecutive $100ms$ intervals over a WiMAX network in Korea. Note that the steady-state bandwidth can vary by more than three orders-of-magnitude and change significantly as seen at the 1000s mark. Hence, an efficient and accurate method of measuring the available backhaul and WLAN bandwidths is of paramount importance to effective aggregation of bandwidth from multiple WLAN APs.

AP aggregation is further complicated by the growing acknowledgment that fine-grained channelization and dynamic spectrum access [30, 7, 41] is critical to enhancing the utilization of wireless channels. Such fine-grained spectrum-usage patterns increase the chance of interference from partially overlapping transmissions, which are not decodable by current PHY/MAC protocols.

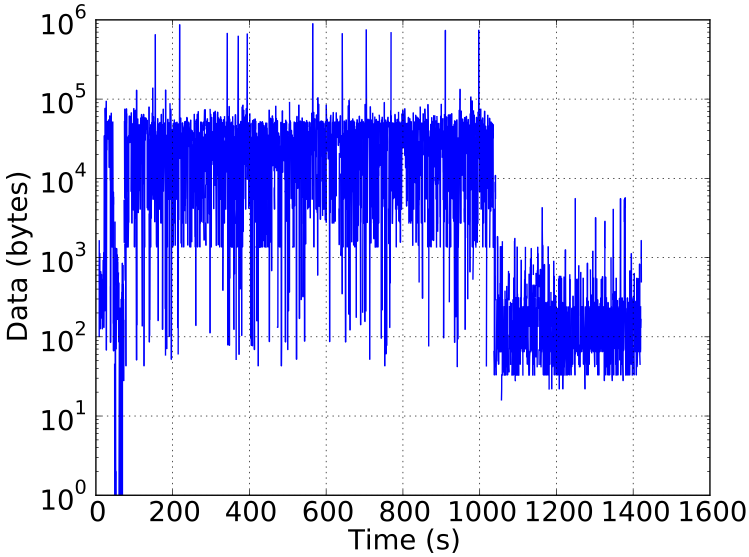


Figure 5.1: Number of bytes received by a static Bittorret client in consecutive 100ms windows over a WiMAX network in Seoul [1].

In this chapter, we present Sidekick—a simple yet novel 802.11a/g/n AP aggregation protocol that achieves efficient multi-AP communications by enabling the APs to take an active role in aggregation by notifying clients of the exact number of backlogged packets through an in-band signaling channel that is based on Aileron [106]. A key innovation here comes from the fact that the clients *need not be on the same channel as the AP to receive this status information.* The in-band signaling technique can efficiently and accurately convey bandwidth information to clients that are tuned to channels that only partially overlap with the channel of the AP. Sidekick also includes a MAC-layer protocol that integrates this real-time traffic information into an optimal schedule that maximizes the achievable throughput over multiple APs.

Sidekick offers the following benefits over existing AP aggregation techniques:

Retrieval of traffic information over partially-overlapping channels. Sidekick nodes can exchange traffic information as long as the spectrum of the channel used by the AP partially overlaps with the spectrum used by the client; the client and AP do not have to be tuned to the same channel. Communication through partially overlapping channels has been used in [107], but that method is only applicable to the older 802.11b standard and cannot be employed with OFDM-based 802.11a/g/n networks. Aileron offers a novel and reliable signaling channel with a performance that is independent of the bandwidth of the overlapping spectrum.

Low overhead signaling. Sidekick nodes can exchange traffic information with very low overhead. With Aileron, a Sidekick AP can embed queue length information in a side-channel using the RTS/CTS or data frames that are used for regular co-channel transmissions; Sidekick clients on partially overlapping channels can extract this queue information from the side-channel without requiring any additional signaling or synchronization bits. This feature stands in stark contrast with regular co-channel communications where proper synchronization in the form of a known preamble along with channel access procedures involving SIFS and DIFS delays are needed to accurately transmit network state information from an AP to a client.

Accurate tracking of time-varying channel state. As a net result of effect communication over partially overlapping channels and low-overhead signaling, a Sidekick client can efficiently determine the number of queued packets for itself at every AP with minimal probing overhead.

The rest of the chapter is structured as follows. We discuss related work in §5.2 and give details on the design of Sidekick in §5.3 and §5.4. We evaluate Sidekick in §5.5 and §5.6.

5.2 Related Work

Multi-Net [108] is the first virtualization platform for wireless interfaces. It consists of a specially crafted device driver that exposes multiple virtual devices, one for each available AP, to the rest of the network stack; a fixed and an adaptive scheme are used to govern the switching policies among different APs. Juggler [109] is built upon Multi-Net and improves its ability to quickly switch between multiple APs, thereby allowing efficient use of AP aggregation under dynamically changing network conditions. FatVAP [104] is an AP aggregation scheme that focuses on achieving maximum aggregate throughput by optimizing the duration and the order of AP connections using dynamic programming. Arbor [110] is a similar aggregation scheme with the added focus on aggregation over secure wireless networks. THEMIS [105] takes a different approach with a focus on fairness between multiple aggregating clients; in a blind aggregation scheme such as FatVAP well-connected clients can easily consume an excessive amount of bandwidth at the expense of more poorly connected clients.

ViFi [111] is an extension of this multi-AP aggregation concept to the mobile scenario: it exploits the diversity offered by simultaneous use of multiple APs to provide continuous WiFi access to moving vehicles. JellyNets [112] is another interesting integration of AP aggregation with pocket hypervisors on mobile devices.

5.3 Sidekick MAC Protocol

Sidekick consists of both PHY and MAC-layer protocols. The PHY-layer design enables accurate communication across multipe partially overlapping channels while the MAC-layer harnesses this ability to efficiently aggregate multiple backhaul links across different APs. The *connection schedule* computed by each Sidekick client determines both the duration and the order in which the client connects to the multiple APs. We present two different algorithms for computing the schedule, `Sidekick-ILP` and `Sidekick-Greedy`. `Sidekick-ILP` constructs the schedule using an Integer-Linear Program (ILP), similar to that used by FatVAP [104], while `Sidekick-Greedy` visits the APs greedily in order of decreasing queue length.

5.3.1 Overview

We consider a scenario with with N Sidekick APs $X_1, \ldots, X_N$ and a single Sidekick client. A single wired backhaul link is connected to each AP. Each AP X_i, $i \in \{1, \ldots, N\}$ has a backhaul link with throughput of b_i. This backhaul link can be wired, as is the case for home broadband networks, or wireless, as is the case for 3/4G routers. The wireless throughput between X_i and the client is denoted by w_i. In order for the aggregation of multiple backhaul links to be feasible, the inequality $b_i < w_i$ must be met. As is the case with ordinary WLAN clients, the Sidekick client is assumed to know the channel of each available AP.

The connection schedule is represented by a pair of lists (P, D). P is a list of APs to be visited and each $X_i \in P$ has a corresponding entry $t_i \in D$ representing the length of time that the client should remain connected to AP X_i. When a Sidekick client switches away from an AP, it uses the 802.11 power-save mode feature to ensure that packets that arrive at the AP in its absence are buffered.

5.3.2 Sidekick-ILP

Given APs $X_1, \ldots, X_N$, each with backhaul and wireless bandwidths b_i and w_i, $i \in \{1, \ldots, N\}$, respectively, the schedule can be computed using an algorithm similar to that used in [104]:

$$\max \quad \sum_i f_i p_i \quad \text{s.t.} \tag{5.1}$$

$$\sum_i (f_i T + \lceil f_i \rceil s) = T \tag{5.2}$$

$$\forall i \quad 0 \leq f_i \leq \min\left\{\frac{q_i}{p_i}, 1\right\} \tag{5.3}$$

where s is the delay incurred when switching from one AP to another, T is the time quantum of the schedule, q_i is the length of the queue at AP X_i and $p_i = w_i T$ is the maximum number of packets that can be transmitted from X_i to the client within the time duration T. The connection schedule is then constructed from the solution of the optimization algorithm as (P, D) where $P = [X_1, \ldots, X_N]$ and $D = [f_1, \ldots, f_N]$.

This optimization algorithm seeks to maximize the total number of packets downloaded within a time interval T by determining the optimal length of the duty cycle, $f_i T$, that should be spent at each AP X_i. The length of this duty cycle is proportional to the ratio of the current queue length to the maximum number of packets that can be transmitted over the wireless link within one time quantum. This time quantum, T, determines the maximum duration of all duty cycles and is an upper bound on the TCP acknowledgement delay from the wireless node. We select $T = 100ms$ so that a fair performance comparison can be made with FatVAP. The constraint (5.2) ensures that the time consumed by the duty cycles and the switching overhead do not exceed the stated time quantum.

The optimization algorithm shown here does not explicitly ensure an upper bound on the time interval between two consecutive visits by the client to the same AP. Hence, it is possible for the length of the queue at some AP X_i to grow beyond the number of packets that can be transmitted over the wireless link within one time quantum. The resulting ratio $q_i/p_i > 1$ will cause constraint (5.2) to be violated. Constraint (5.3) ensures the feasibility of the optimization by restricting the upper bound of f_i for all APs X_i.

This optimization problem can easily be reformulated as an Integer-Linear Program

$$\max \quad \sum_i f_i p_i \quad \text{s.t.} \tag{5.4}$$

$$\sum_i (f_i T + y_i s) = T \tag{5.5}$$

$$\forall i \quad 0 \leq f_i \leq \min\left\{\frac{q_i}{p_i}, 1\right\} \tag{5.6}$$

$$f_i \leq y_i \leq 1, \; y_i \in \mathbb{Z}, \tag{5.7}$$

thus allowing the use of off-the-shelf optimization routines.

5.3.3 Sidekick-Greedy

Sidekick clients have up-to-date information on the length of the packet queues at the APs. Hence, a simple greedy algorithm can also be employed where the client connects to APs in decreasing order of queue lengths. In contrast to `Sidekick-ILP`, `Sidekick-Greedy` returns an *ordered* connection schedule; the AP connections under `Sidekick-ILP` are not guaranteed to be carried out in any particular order. Fig. 9 shows the pseudocode for `Sidekick-Greedy`.

In `Sidekick-Greedy`, a max-heap is used to keep track APs, in decreasing order of queue lengths, that have not yet been scheduled. For each AP X_i at the top of the heap, the total time needed to empty the queue, T_i is calculated first. If this time T_i can fit into the current schedule without the total schedule time exceeding the time quantum T, then X_i and T_i are appended to the schedule lists P and D, respectively. Otherwise, the remaining available time in the schedule, if any, is assigned to X_i and the completed connection schedule is returned.

5.3.4 Using the Entire Time Quantum

Under both `Sidekick-ILP` and `Sidekick-Greedy`, the total connection time in the schedule may be less than the time quantum. Hence, we adjust the connection times of the client to each AP to be proportional to the relative queue length of that AP. The pseudocode for this step is shown in Fig. 10.

This adjustment to the connection schedule is made to improve the overall utilization of the wireless channel. In the adjusted schedule (P, D'), the client visits each AP once during each time quantum, as opposed to multiple times per time quantum without the adjustment, and therefore, reduces the time wasted on the AP switching.

Algorithm 9: Sidekick-Greedy algorithm.

```
input  : The queue length, $Q = q_1, \dots, q_N$, and bit rate, $R = r_1, \dots, r_N$, of each AP $X_i$,
         $i \in 1, \dots, N$
output: The connection schedule $P$ and connection duration $D$ for all APs
begin
    $T \leftarrow$ time quantum, $s \leftarrow$ switching time;
    $h \leftarrow$ make_max_heap($Q$), $t \leftarrow 0$;
    $P \leftarrow$ empty_list(), $D \leftarrow$ empty_list();
    while $h$ is not empty do
        $q_i \leftarrow$ pop_heap($h$);
        $T_i \leftarrow q_i / r_i$;
        if $t + T_i + s \leq T$ then
            $t \leftarrow t + T_i + s$;
            $P \leftarrow$ append($P, X_i$);
            $D \leftarrow$ append($D, q_i / r_i$);
        else
            $P \leftarrow$ append($P, X_i$);
            $D \leftarrow$ append($D, T - t - s$);
            $t \leftarrow t + T_i + s$;
            break;
        end
    end
    return $(P, D)$;
end
```

Algorithm 10: Adjusting the connection schedule to ensure that the entire time quantum is utilized.

```
input  : Connection schedule $(P, D)$.
output: Adjusted connection schedule $(P, D')$ such that $|D| \cdot s + \sum_{d_i \in D'} d_i = T$, where $T$ is
        the time quantum and $s$ is the switching delay.
begin
    $T_D \leftarrow |D| \cdot s + \sum_{d_i \in D} d_i$;
    $T_R \leftarrow T - T_S$;
    $D' \leftarrow$ empty_list();
    for $k \in 1, \dots, |D|$ do
        $D'[k] \leftarrow D[k] + (D[k] / T_D) \cdot T_R$;
    end
    return $(P, D')$;
end
```

5.3.5 Responding to Bandwidth Changes

Sidekick uses partially overlapping channels for control messages, thus adapting to varying bandwidth is an integral portion of the Sidekick protocol. When a new Sidekick AP comes online, Sidekick nodes exchange information on the bandwidth increases using a protocol that has two distinct portions: a broadcast protocol that is run on the AP and an adaptation protocol that is run on the client.

The broadcast protocol used by the AP is straightforward: an AP broadcasts its available aggregation capacity by piggy-backing such notifications on the RTS/CTS frames that are used for co-channel communication. Such broadcasts occur at least once per time quantum. If the co-channel transmission rate is lower than one packet per time quantum, the AP will broadcast its available capacity using a special short broadcast frame. This frame will be described in §5.4.

The adaptation protocol running on the client responds to these broadcast messages and adds the newly-available APs to the pool of APs considered by the scheduling algorithms. The client assigns a new TCP flow to each new AP that broadcasts its availability. New APs are added to the scheduling algorithm one at a time. This is to ensure that the client can allocate sufficient connection time to an AP to allow TCP to quickly run through its slow-start phase to reach its steady-state transmission rate. When a new-AP broadcast is detected by a client, it connects to the AP for a duration of $T/2$ and starts a new TCP connection through that AP. This AP is then added to the pool of APs for use by the next iteration of the scheduler. If multiple broadcasts are detected, the APs are added one at a time in a random order, with only one AP added between consecutive calls to the scheduling algorithm.

If no packets are detected from an AP for a duration of $10T$, the AP is assumed to be offline and will be removed from the pool of APs used by subsequent invocations of the scheduling algorithm.

5.3.6 Overall Sidekick MAC Protocol

Figures 11 and 12 show the pseudocode of the overall Sidekick MAC for the AP and the client, respectively. Note that the Sidekick client needs to handle the situation where the length of all queues of active APs are zero (lines 19 and 20 of Fig. 12). This can occur sporadically due to the bursty nature of TCP packet arrivals and the fact that the wireless bandwidth can be significantly larger than the backhaul bandwidth.

Algorithm 11: Sidekick MAC protocol on the access point

```
input : Time quantum, $T$
begin
    while true do
        if New AP then
            Broadcast available capacity;
            sleep (T);
        else if Transmitting RTS or CTS frame then
            Embed queue lengths and IDs of at most 8 randomly selected clients into the
            RTS/CTS frame;
        end
        else if No RTS/CTS transmission for $T$ seconds then
            Broadcast queue lengths and IDs of at most 8 randomly selected clients into a
            Sidekick broadcast frame;
        end
    end
end
```

5.4 Sidekick PHY Protocol

5.4.1 Design of the Control Channel

Sidekick uses two different control messages to convey queue information from the AP to the client: broadcast and directed. Broadcast messages are used by APs to notify clients of new transmission opportunities and are described in §5.3.5; directed messages are sent from an AP to a specific client and are used to notify the client of the number of its packets queued at the AP.

The PHY-layer design of the two messages is illustrated in Fig. 5.2. Here, we focus on the 40MHz channel of the 802.11n network with 128 OFDM subcarriers, but the control message design is similar for networks of other bandwidths.

A key feature of Sidekick is its ability to pass queue length information between the AP and the client nodes regardless of the bandwidth of the overlapping spectrum between the AP and client using Aileron. Recall from §4.3 that Aileron encodes information in the modulation of the subcarrier rather than its symbol value. In 802.11n WLANs, adjacent channels are separated by 5MHz, which is spanned by 16 subcarriers. Sidekick takes advantage of this fact and divides the 128 subcarriers into 8 groups of 16 subcarriers each. We refer to each group of 16 subcarriers as a *subcarrier group*. In order to minimize interference between adjacent subcarrier groups, a single subcarrier between two adjacent subcarrier groups is designated as the guard subcarrier and is not used for data transmission. Of the 15 remaining subcarriers, 8 are used as spacing subcarriers, as required by Aileron, and are only modulated with

Algorithm 12: Sidekick MAC protocol on the client

```
input  : Time quantum, $T$
new_ap ← empty_list ();
active_aps ← empty_list ();
ap_queue ← empty_list ();
/* Wireless bandwidth                                                  */
wl_rate ← empty_list ();
while true do
    if Broadcast from new AP, $X$, received then
        new_ap ← append (new_ap, $X$);
    end
    if |new_ap| $> 0$ then
        $X$ ← remove_head(new_ap);
        Connect to $X$ for $T/2$ seconds and start new TCP connection;
        $Q_X$ ← queue length of $X$;
        $R_X$ ← wireless bandwidth between client and $X$;
        active_aps ← append(active_aps, $X$);
        ap_queue ← append(ap_queue, $Q_X$);
        wl_rate ← append(wl_rate, $R_X$);
    end
    if length of all data queues of active APs is zero then
        Associate with a random AP and wait for queue length updates via control messages ;
    end
    $(P, D)$ ← Sidekick-ILP (ap_queue, wl_rate) or Sidekick-Greedy (ap_queue, wl_rate);
    Connect to the APs according to the connection schedule $(P, D)$;
    if No queue length update from $X$ for $10T$ seconds then
        Remote $R_X$, $Q_X$ and $X$ from wl_rate, ap_queue and active_aps respectively;
    end
end
```

BPSK; the other 7 data subcarriers can be encoded with either BPSK, QPSK or 8PSK as described in §4.3

Sidekick uses these 7 data subcarriers as follows: 3 subcarriers are used for an address, which is the client address in a directed message, or a special broadcast address for broadcast messages; 4 subcarriers are used to encode the queue length. Sidekick can therefore transmit queue lengths of up to 16 packets and any queue containing more than 16 packets is simply encoded using the largest supported value. Note that the division of subcarriers between client addresses and queue lengths can be varied according to the network configuration. We leave such configuration details to future work.

Sidekick encodes a different client address and associated queue length in each subcarrier group, as shown in Fig. 5.2, with a different random mapping between client addresses and subcarrier groups in every control message.

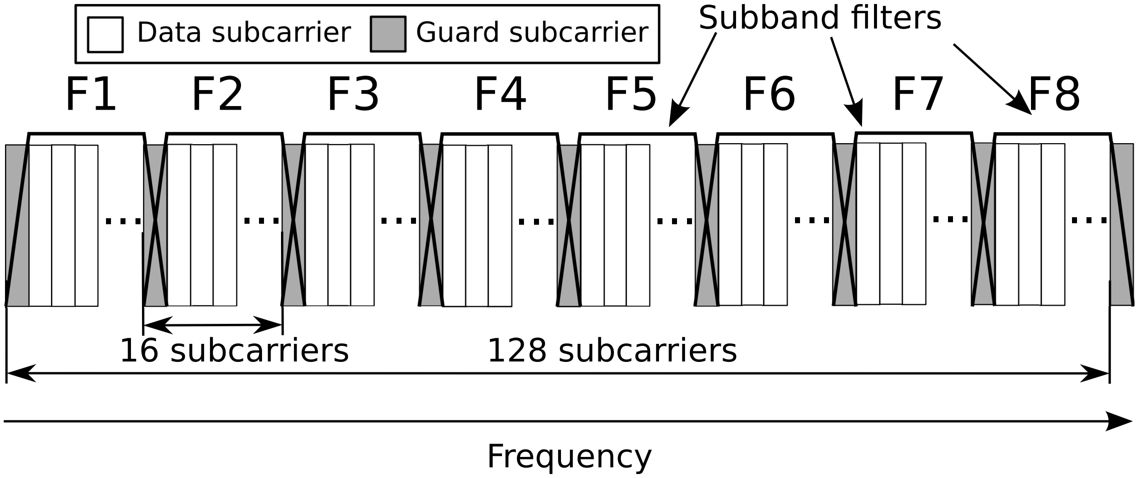


Figure 5.2: PHY-layer signaling frame.

Aileron constructs the control message by repeating the PHY-layer layout and modulation encoding of Fig. 5.2 in at least 10 consecutive OFDM symbols. Sidekick transmits these control messages using two possible methods: embedded into an RTS/CTS frame or as a separate control frame. With embedded transmission, the modulation used in subcarriers of the RTS/CTS frame are set according to that shown in Fig. 5.2; if a separate control frame is used, Sidekick transmits 10 consecutive OFDM frames carrying random data, with the modulation rates of the subcarriers also set as shown in Fig. 5.2.

5.4.2 Addressing the APs

The source address of partially overlapping Sidekick APs can be transmitted in three different ways:

(a) Encoded using Aileron. Some of the subcarriers in each overlapping subcarrier group can be used for encoding the AP IDs. We can also increase the total number of subcarriers in each OFDM symbol to obtain more subcarriers for encoding the AP/client addresses and the queue lengths.

(b) AP-specific preambles. Each AP can use a unique preamble that is repeated in every subcarrier group. This preamble is generated based on the ID of the AP.

(c) Fixed channel-to-AP mapping. Each Sidekick AP can be assigned to a unique channel that is not occupied by any other Sidekick AP. The address of the AP can then be inferred from the offset of overlapping AP transmission from the channel of the Sidekick client. The advantage of this approach is that no additional subcarriers are needed to encode the AP address.

5.4.3 Receiving Control Messages

There are three key steps for a client to correctly decode the queue length information in the Aileron packet: detecting the transmission, finding the edge of the partially overlapping packet and finally decoding the modulation-encoded message.

Detection. The RTS/CTS and separate Aileron control frames are significantly shorter than the standard WLAN data frames. Hence, the Sidekick client can differentiate control from data frames from the duration of the energy burst [95]. After the transmission has been detected, edge detection is carried out.

Edge Detection. A partially overlapping transmission will only occupy a fraction of all the OFDM subcarriers available to the client. Edge detection enables the client to determine the subcarrier groups that contain a valid transmission from an AP. The client, when operating over a 40MHz 802.11n channel, uses 8 channel filters, each spanning a 5MHz bandwidth. Fig. 5.2 illustrates the arrangment of these filters as well as the associated labels $F_1, \dots F_8$. The edge of the partially overlapping transmission can be determined using the algorithm shown in Fig. 13. Here, the *lower limit* refers to the edge of a partially overlapped transmission that spans F_1 to F_k for $2 \leq k < 8$, while the *upper limit* refers to the edge of a transmission that spans F_k to F_8 for $1 \leq k < 8$. If the limits cannot be found (lines 18-19), that means that the received Aileron control message was transmitted from an AP tuned to the same channel as the client. Note that we assume that the wireless channels of all nodes have the same bandwidths, thus a control message sent over a partially overlapping control cannot have both a upper and lower limit; we leave the case of networks with heterogenous channel bandwidths to future work.

Decoding. Once we have located the boundary of the partially overlapping transmission, we can decode the modulation-encoded message (i.e. client address and queue lengths) from all the subcarrier groups that it occupies. The decoding accuracy depends on the Signal-to-Interference-and-Noise Ratio (SINR) of the channel. The interference is due to other partially overlapping transmissions to the same client node.

5.4.4 Multiple Sidekick Clients

For simplicity, our exposition of Sidekick has thus far focused on the multi-APs-single-client case. We now give an overview of the simple extensions needed for Sidekick to operate in a multi-APs-multi-clients environment. We leave the detailed evaluations of Sidekick in this multi-clients scenario as our future work.

Sidekick APs maintain a separate packet queue for each Sidekick client. The Sidekick APs then embeds the ID of a client and its associated queue length in the transmitted control

Algorithm 13: Search for the upper and lower limits of partially overlapping control messages transmitted over a 40MHz 802.11n channel with 8 subcarrier groups.

```
input : Aileron Control Message
begin
    lower_limit $\leftarrow -\infty$;
    upper_limit $\leftarrow \infty$;
    for $k \leftarrow 1$ to 7 do
        if energy($F_k$)/energy($F_{k+1}$) $> \delta$ then
            lower_limit $\leftarrow k$;
            break;
        end
    end
    for $k \leftarrow 8$ to max(lower_limit, 2) do
        if energy($F_k$)/energy($F_{k-1}$) $> \delta$ then
            upper_limit $\leftarrow k$;
            break;
        end
    end
    if lower_limit $>$ upper_limit then
        return NULL;
    else if lower_limit $= -\infty$ and upper_limit $= \infty$ then
        return Co-channel control message received;
    end
    return (lower_limit, upper_limit);
end
```

messages. The Sidekick PHY can transmit information on up to 8 different clients in a single broadcast message. If the number of clients is greater than 8, the AP will simply embed queue information on 8 randomly selected clients in each control message.

A Sidekick client that decodes this control message can receive information on its queue on an AP if (a) it is one of the 8 random clients selected by the AP and (b) its queue length information lies in the overlapping subcarriers of the client and AP. If a Sidekick client does not find its queue information in the control message, it simply omits the current AP from the schedule computation.

In such a scenario, Sidekick clients may not have complete information on the state of the AP queues. Sidekick will not be able to find a schedule that maximizes the transmission opportunities at the APs, thus resulting in a reduced aggregated throughput. However, we expect this reduced throughput to still be greater than the throughput that can be achieved without Sidekick and we leave detailed evaluations to future work.

5.5 Evaluation of the Sidekick PHY

5.5.1 Experimental Setup

We implemented Sidekick PHY using GNURadio and evaluated it over simulated channels. This use of simulated channels allows us the flexibility of systematically exploring the performance of the Sidekick PHY over a wide range of channel conditions, without any constraints imposed upon us by the physical layouts of our office environment. The parameters used to evaluate the Sidekick PHY is summarized in Table 5.1.

In order to evaluate Sidekick PHY in a simulated environment, we first generate two partially-overlapping transmissions: the data transmission, S_D, spans $F_1, \ldots, F_5$ while the other interfering transmission, S_I, spans $F_3, \ldots, F_8$. These streams are passed through a MATLAB filter that combines them and fading and shadowing effects, along with Gaussian noise, to the signal. MATLAB keeps the signal energy of S_D constant while varying that of S_I to produce different Signal-to-Interference values (SIR); the added noise energy is also varied to control the Signal-to-Noise (SNR) of the output signal. This distorted signal is then passed to the Sidekick receiver where the original modulation-encoded information is recovered. The Sidekick PHY is evaluated using the following channel models in MATLAB: `jtcInResC`, `jtcInOffC`, `jtcInComC` that correspond to "Indoor Residential C", "Indoor Office C" and "Indoor Commercial C". We only show the simulation results using `jtcInOffC` as it is similar to the performance of Sidekick under other channel models.

PHY Parameter	Value
Center frequency	2.4GHz
Total bandwidth	12.5MHz
Total subcarriers	512
Cyclic prefix length	256
No. of subcarrier groups	8
No. of subcarriers per subchannel	64

Table 5.1: Parameters used in the Sidekick PHY.

5.5.2 Results

Fig. 5.3 shows a contour plot of the probability of correctly detecting the edge of the data transmission, S_D, under different interference and noise energy levels. Observe that the ability of Sidekick to accurately locate the edge of a transmission is highly dependent on the interference energy: at an SIR above 6dB, Sidekick can determine the edge of a partially-overlapping transmission with over 90% accuracy. Furthermore, there is a sharp change in the edge detection probability: from 0 to 6dB, the probability of accurately finding the edge increases rapidly from 0 to 90%. Also note that the levels of Gaussian noise energy has little impact on edge detection accuracy: for a given SIR level, the edge detection accuracy remains fairly constant over all SNR levels.

Fig. 5.3 illustrates the edge detection performance with only a single interfering transmission. However, the results shown here are representative of a *lower bound* on detection accuracy. This is because the SIR level at which 90% edge detection accuracy occurs actually *decreases* with increasing numbers of interfering transmissions: by the Law of Large Numbers, as the number of interfering transmissions increases, the statistical properties of the interference approaches that of Gaussian noise, which has very limited impact on the edge detection accuracy of Sidekick.

After Sidekick detects the edge of a partially-overlapping transmission, it decodes the data encoded in the Aileron packet. Fig. 5.4 shows this decoding accuracy at different interference and noise levels. Observe that a 90% decoding accuracy can be achieved only at SIR and SNR above 14dB and 16dB, respectively. In contrast to the edge detection performance, both the interference and noise energy levels have significant effects on the decoding accuracy. Hence, as long as the client ensures that SIR and SNR on the operating channel are at 14dB and 16dB, respectively, it can be assured that the edge and decoded data can be recovered with at least 90% accuracy.

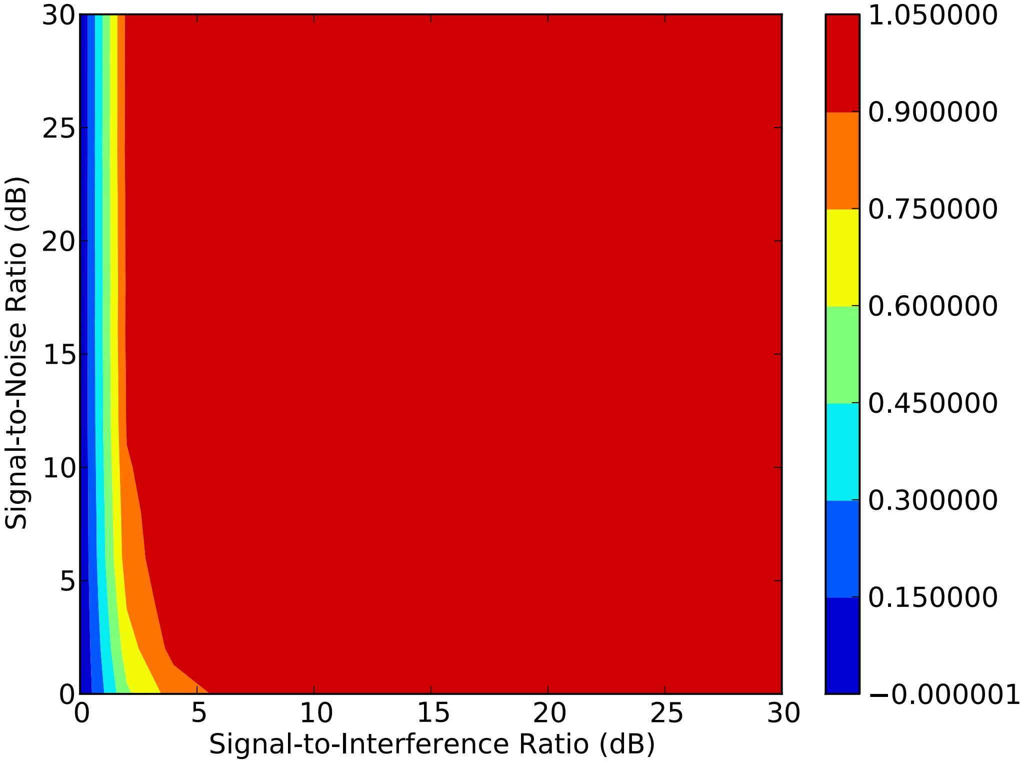


Figure 5.3: Probability of correctly detecting the edge of S_D in channels with different interference and noise energy levels

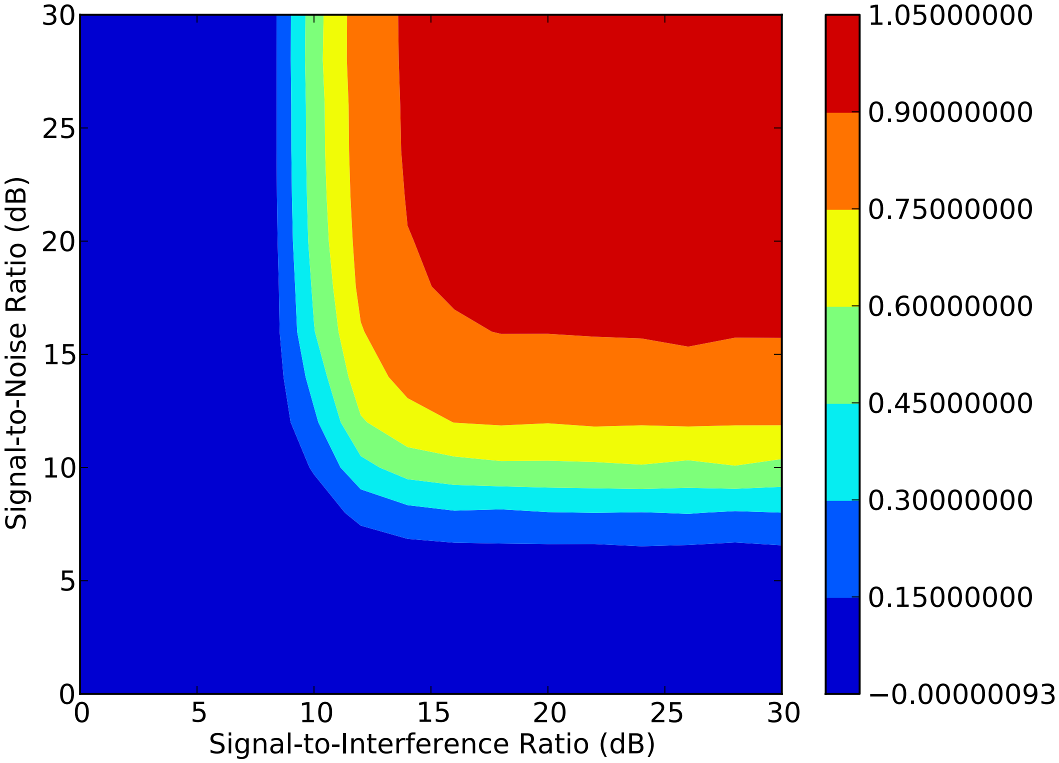


Figure 5.4: Probability of correctly decoding the client ID and queue length information in S_D under different interference and noise energy levels

Number of APs	5, 10, 15, 20
Bandwidth of each backhaul link	1Mbps
Bandwidth of wireless channel	54Mbps
Wired backhaul cross traffic rate	0.1 to 1Mbps, in increments of 0.1Mbps
Cross traffic model	Exponential, Pareto
Average On/Off duration	1s On, 2s Off
Simulation duration	250s
Number of repetitions per experiment	10

Table 5.2: Parameters used in the evaluation of Sidekick MAC in ns-2

5.6 Evaluation of the Sidekick MAC

We implemented the Sidekick MAC on ns-2 and evaluated its performance under a myriad of conditions. Table 5.2 lists the parameter values used in the simulation evaluation of Sidekick. We consider a scenario with multiple APs and one or more clients. Each AP has a single backhaul link that is connected to the Internet.

In this section, we will evaluate the Sidekick MAC with schedulers `Sidekick-ILP` and `Sidekick-Greedy`. For brevity, we will refer to these two Sidekick configurations as `Sidekick-ILP` and `Sidekick-Greedy` directly. The performance of these Sidekick configurations will be compared to that of FatVAP, which is a notable multi-AP aggregation protocol. FatVAP does suffer from an inability to receive out-of-band queue or bandwidth information from candidate APs — it must first connect to an AP before it can measure traffic statistics that are necessary for constructing a connection schedule. We will demonstrate th performance gains that come from the partially-overlapping signaling capability of Sidekick.

5.6.1 Performance Under Static Conditions

We first evaluate Sidekick under static network conditions: all backhaul links, along with the associated cross traffic, are active at the start of the simulation and only one client is present. We compare the ability of Sidekick and FatVAP to efficiently select the best subset of APs to use.

Fig. 5.5 shows the total download by the single client over the entire 250s simulation run. Here, the APs are configured such that each AP has a degree of two: the channel used by each AP overlaps with exactly two other randomly-selected APs. Observe that the total amount of data downloaded by `Sidekick-ILP` and `Sidekick-Greedy` are relatively independent of contending traffic on the backhaul link, with `Sidekick-ILP` outperforming `Sidekick-Greedy` by a margin of less than 10%. FatVAP, on the other hand, outperforms Sidekick when the backhaul links are lightly loaded — with the cross traffic throughput is under 400kbps,

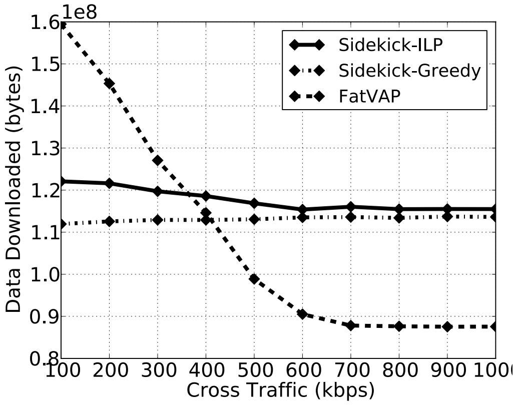


Figure 5.5: Total data downloaded by a single client from 10 APs over the 250s simulation run with different cross traffic speeds on the backhaul link.

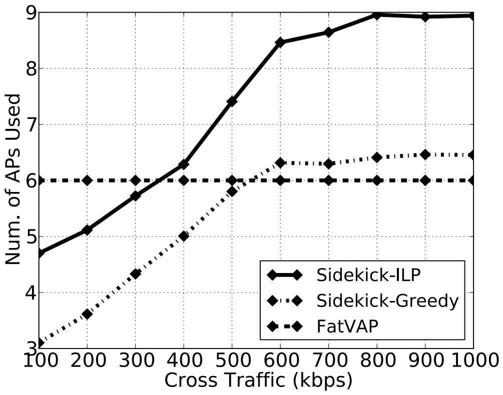


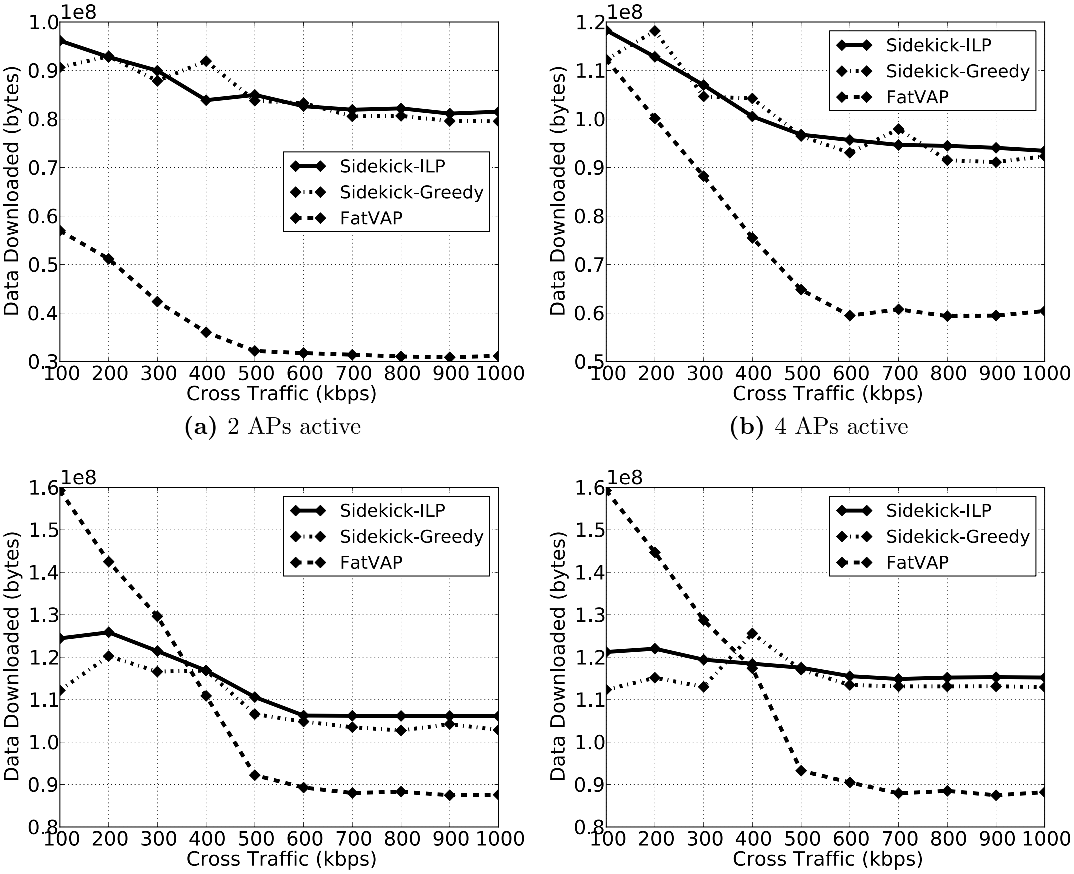
Figure 5.6: Mean number of APs active in a connection schedule under different cross traffic rates. A total of APs are present and the channel of Sidekick AP partially overlaps with that of exactly one other randomly-selected AP.

FatVAP can download up to 45% more data than `Sidekick-Greedy`. However, when the backhaul links are heavily-loaded, both `Sidekick-Greedy` and `Sidekick-ILP` download at least 30% more data than FatVAP.

The reason for this behavior lies in the number of APs that are selected by Sidekick and FatVAP as part of the connection schedule. Fig. 5.6 shows the number of APs that are active in the schedule computed by `Sidekick-ILP`, `Sidekick-Greedy` and FatVAP under varying cross traffic throughput. FatVAP selects its set of APs based on the average wireless and wired throughput measured over a 2-second window. This minimizes the impact of that short term variations, due to the on-off nature of the cross traffic and the bursty nature of typical TCP flows, will have on the resulting schedule. Hence, it maintains a constant set of 6 APs that are active in every connection schedule.

The number of APs used by Sidekick does not increase further with increasing overlapping degrees of each AP. Hence, in the rest of this section, we will consider only APs with overlapping degrees of two.

The Sidekick client, on the other hand, receives real-time information on the actual length of the queue at each AP and is therefore more sensitive to variations in the queue lengths over short time scales. The burstiness of the packet arrival increases as the throughput of the cross traffic increases and during time quantums in which many APs have short queues, Sidekick adapts by increasing the number active APs in its schedules. The use of queue lengths (Sidekick) instead of transmission rate (FatVAP) in constructing the connection



(a) 2 APs active

(b) 4 APs active

(c) 6 APS active

(d) 8 APs active

Figure 5.7: Average total data downloaded over 20 simulation runs under different cross traffic throughput and number of active APs at the start of the simulation.

schedule also ensures that the client will eventually connect to slow APs when the queues on those APs have grown to be sufficiently large. This enables Sidekick to maintain its data transfer rate in the face of many low bandwidth backhaul links.

5.6.2 Adapting to Significant Bandwidth Changes

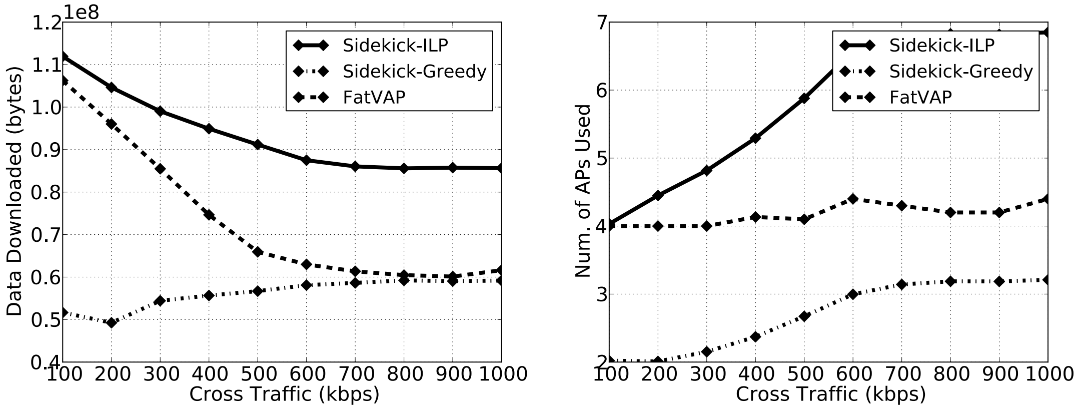
Sidekick is able to search for new APs while simultaneously connecting to the currently active set of APs in is connection schedule due to the use of in-band Aileron signaling. Here, we evaluate the efficacy of this AP discovery mechanism. This proceeds as follows: we run the ns-2 simulation with a total of 10 APs as before, but only a fraction of these APs are active at the start of the simulation. After 100s, the remaining non-active APs are brought online and begin to advertise bandwidth availability to the Sidekick client.

Fig. 5.7 shows the total data downloaded over the 250s simulation run with different numbers of active APs at the start of the simulation. When only 2 APs are active at the start of the simulation, FatVAP can only achieve a maximum download of 60MB, while both `Sidekick-ILP` and `Sidekick-Greedy` can download at least 80MB in 250s. Similar behavior can be observed when 4 APs are active at the beginning of the simulation. This stark difference in performance between FatVAP and Sidekick is due to the fact that Sidekick can quickly detect the new APs at the 100s mark and add these APs to the connection schedule; FatVAP, on the other hand, does not probe for additional transmission opportunities and therefore cannot take advantage of the bandwidth offered by the newly active APs. When 6 and 8 APs are active at the beginning of the simulation, FatVAP does achieve its maximum performance as seen earlier in Fig. 5.5 because it only uses a maximum of 6 APs in its schedule.

5.6.3 Performance with Wireless Contention

In this section, we evaluate the performance of Sidekick in the presence of channel contention from other wireless clients. We model the effect of wireless interference by randomly varying the bandwidth of each channel from an AP to the client between 22 and 54Mbps. This setup succinctly captures the effects of channel interference from both WLAN nodes and other noise sources while enabling us to focus on the behavior of Sidekick. The bandwidth of each wireless channel is fixed at the start of the simulation and each experiment is repeated 20 times. We run the simulation with 10 APs so that both Sidekick and FatVAP will not be constrained by the available backhaul bandwidth; all APs are active at the start of the simulation.

Fig. 5.8 shows the performance of Sidekick and FatVAP in this scenario. Observe that `Sidekick-ILP` outperforms both `Sidekick-Greedy` and FatVAP. The improvement of `Sidekick-ILP` over FatVAP comes from its access to real-time information on the queue length and wireless rate. `Sidekick-Greedy`, on the other hand, only takes the queue length information into account and hence cannot determine the optimal order of APs in its connection schedule when faced with wireless links of significantly varying throughput. This is also evident by the fact that `Sidekick-Greedy` uses significantly fewer APs in its connection schedule, as compared to `Sidekick-ILP` and FatVAP — `Sidekick-Greedy` often gets “stuck” on APs with long queue sizes and low wireless throughput.



(a) Total data downloaded

(b) Mean #APs per schedule

Figure 5.8: Total data downloaded under different cross traffic data rates with wireless contention

Chapter 6

Conclusion

Thesis Statement: *Next-generation networks that incorporate software-defined programmability, PHY coordination, spectrum and protocol agility is novel and absolutely necessary to meet the capacity and coverage demands of future wireless networks.*

In the face of growing wireless protocol complexity and increasing demand for ubiquitous connectivity, disparate, fixed-function wireless network architectures can no longer keep up with the required adapability and dynamisim required. Instead, we need a new flexible and programmable software-defined wireless architecture that supports antenna coordination, programmable RF frontends and centralized processing of wireless protocols.

This dissertation fulfills the thesis statement by introducing key technological advances that (a) enhances current off-the-shelf devices with spectrum agility and integrates them into future wireless networks, (b) enables the deployment of next-generation networks over low-cost, commodity backhaul networks and (c) facilities low overhead coordination and communication over spectrum-agile networks. Each of these advances provides clear and demonstrable benefits over legacy wireless networks and serves as building blocks for next-generation wireless architectures.

This dissertation studies the design of next-generation, software-defined wireless network architecture and analyzes the key components required to build such a network. Towards that goal, this dissertation covers four key pieces of work:

Rodin demonstrates an approach to integrate existing wireless devices into the new wireless architecture by bringing spectrum agility to COTS devices;

Aileron is a novel approach to spectrum coordination that is necessary for fast and efficient cognitive spectrum access;

Sidekick efficiently and effectively aggregates disparate blocks of spectrum from different wireless APs; and

Spiro illustrates a novel backhaul management and compression technique to enable the transport and processing of coordinated multipoint RF data over commodity Ethernet networks.

The mechanisms and techniques presented in this dissertation serve as the fundamental building blocks for future wireless networks.

Impact of Future Technological Advances

We present four key technologies necessary for future networks. However, the development of future new and existing technologies will have an effect on the advances presented in this thesis.

Rodin facilities interoperability between current fixed-function networks into future programmable, software-defined wireless networks. However, the need for such an interoperable platform will gradually diminish as the number of deployed next-generation devices increases. Even so, the success of this evolution into next-generation networks depends critically on the existence of such hybrid devices.

Aileron and Sidekick enables low-overhead coordination across heterogenous spectrum agile devices. Future developments into separate low-power hardware and communication channels will reduce the need for a non-coherent signaling channel.

Spiro manages and compresses the I/Q samples over the backhaul network. Future developments in high bandwidth and low latency network devices (e.g. fiber or microwave backhauls) will have limited impact on the necessity for Spiro as such developments mainly affect the scale of the network. With a higher backhaul capacity, we can obviously support a larger number of RRUs. However, the aim of Spiro is to maximize the number of RRUs that can be supported. Hence, the usefulness of Spiro will only be reduced if developments in backhaul capacity far outstrips the demand from the wireless RRUs.

Future Directions

While this dissertation covers individual components necessary for next-generation networks, it leaves four key questions that still need to be answered before such a network can be realized.

What is the optimal transmission policy in this next-generation network?

These individual components, while significant, must operate as part of a larger, cohesive network in order to be effective in addressing the spectrum scarcity problem. To that end, a coherent and effective policy that defines the communication paradigm of next generation devices must still be defined and evaluated before further progress can be made.

How do we handle centralized processing of RF signals?

Each of these components brings its own computational overhead and tradeoffs to the centralized processing resource. For example, while the use of Aileron may speedup the exchange of control information, non-coherent demodulation of control messages is computationally expensive. Appropriate CPU and power management policies have to be developed to account for this overhead, while ensuring that the timing demands of wireless protocols are met. In light of this, a model of the computational complexity and energy requirements of centralized processing of RF signals must be developed to address the unique demands of next generation networks.

What computing models are necessary for future software-defined wireless networks?

Future software-defined wireless networks are envisioned to make widespread use of commodity general-purpose computing hardware for PHY processing. Such platforms provide a scalable, yet cost-effective solution for the centralized processing of PHY-layer information. However, these shared systems are typically not designed to meet the hard realtime constraints of current PHY protocols. Hence, new PHY protocols that are adaptive to backend computational capabilities may be needed. Furthermore, current general-purpose platforms may need to be extended with a selective and specialized set of hardware resources to meet PHY processing demands.

How do we integrate software-defined PHYs with other network services?

General purpose platforms can execute both the software-defined PHYs and other related network services together. However, this heterogeneous software environment may require new realtime resource scheduling algorithms that can balance both the throughput and latency demands, and the programmable flexibility of the software-defined networking stack.

Bibliography

[1] S. Kim, X. Wang, H. Kim, T. Kwon, and Y. Choi, "CRAWDAD trace snu/bittorrent/tcpdump/static (v. 2011-01-25)." Downloaded from http://crawdad.cs.dartmouth.edu/snu/bittorrent/tcpdump/static, Jan. 2011.

[2] "Cisco visual networking index: Global mobile data traffic forecast update, 2011–2016," *http://www.cisco.com*.

[3] "Mobile broadband: The benefits of additional spectrum," *http://www.fcc.gov/document/mobile-broadband-benefits-additional-spectrum*, October 2010.

[4] A. Gember, A. Akella, J. Pang, A. Varshavsky, and R. Caceres, "Obtaining in-context measurements of cellular network performance," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, IMC '12, (New York, NY, USA), pp. 287–300, ACM, 2012.

[5] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "Achieving high data rates in a distributed mimo system," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, Mobicom '12, (New York, NY, USA), pp. 41–52, ACM, 2012.

[6] S. Bhaumik and S. Chandrabose, "Cloudiq: a framework for processing base stations in a data center," in *Mobicom*, 2012.

[7] K. Tan, J. Fang, Y. Zhang, S. Chen, L. Shi, and J. Zhang, "Fine-grained channel access in wireless LAN," in *SIGCOMM*, 2010.

[8] M. Alizadeh, A. Kabbani, T. Edsall, B. Prabhakar, A. Vahdat, and M. Yasuda, "Less is more: trading a little bandwidth for ultra-low latency in the data center," in *NSDI*, 2012.

[9] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl, "A case for adapting channel width in wireless networks," *SIGCOMM*, 2008.

[10] S. Rayanchu, V. Shrivastava, S. Banerjee, and R. Chandra, “FLUID: Improving throughputs in enterprise wireless LANs through flexible channelization,” in *MOBICOM*, 2011.

[11] H. S. Rahul, S. Kumar, and D. Katabi, “Jmb: scaling wireless capacity with user demands,” in *SIGCOMM*, 2012.

[12] Q. Xu, J. Huang, Z. Wang, F. Qian, A. Gerber, and Z. M. Mao, “Cellular data network infrastructure characterization and implication on mobile content placement,” in *SIGMETRICS*, 2011.

[13] S. Deb, K. Nagaraj, and V. Srinivasan, “Mota: engineering an operator agnostic mobile service,” in *Mobicom*, 2011.

[14] S. Gollakota, F. Adib, and D. Katabi, “Clearing the RF smog: Making 802.11 robust to cross-technology interference,” *SIGCOMM*, 2011.

[15] C.-T. Chou, N. Sai Shankar, H. Kim, and K. G. Shin, “What and how much to gain by spectrum agility?,” *JSAC*, vol. 25, Apr. 2007.

[16] V. Shrivastava, S. Rayanchu, J. Yoon, and S. Banerjee, “802.11n Under the Microscope,” *IMC*, 2008.

[17] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, “The impact of channel bonding on 802.11n network management,” in *CoNEXT*, 2011.

[18] M. Heusse and F. Rousseau, “Performance anomaly of 802.11b,” in *INFOCOM*, 2003.

[19] M. Loiacono, J. Rosca, and W. Trappe, “The snowball effect: Detailing performance anomalies of 802.11 rate adaptation,” in *GLOBECOM*, 2007.

[20] V. Kone, L. Yang, X. Yang, B. Y. Zhao, and H. Zheng, “On the feasibility of effective opportunistic spectrum access,” *IMC*, 2010.

[21] R. Gummadi, D. Wetherall, and B. Greenstein, “Understanding and mitigating the impact of RF interference on 802.11 networks,” *SIGCOMM*, 2007.

[22] T. Lin and Y. Tseng, “Collision analysis for a multi-Bluetooth picocells environment,” *IEEE Communications Letters*, vol. 7, no. 10, 2003.

[23] M. Wellens and P. Mähönen, “Lessons learned from an extensive spectrum occupancy measurement campaign and a stochastic duty cycle model,” *Mobile networks and applications*, 2010.

[24] USRP. http://www.ettus.com.

[25] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker, "Sora: High performance software radio using general purpose multi-core," in *NSDI*, 2009.

[26] WARP. http://mangocomm.com.

[27] Y. Yuan, P. Bahl, R. Chandra, and P. Chou, "Knows: Kognitiv networking over white spaces," *DySPAN*, 2007.

[28] J. So and N. Vaidya, "Multi-channel MAC for ad hoc networks: handling multi-channel hidden terminals using a single transceiver," in *MOBICOM*, 2004.

[29] S. Lakshmanan, J. Lee, R. Etkin, S.-J. Lee, and R. Sivakumar, "Realizing high performance multi-radio 802.11n wireless networks," in *SECON*, 2011.

[30] L. Yang, W. Hou, L. Cao, B. Zhao, and H. Zheng, "Supporting demanding wireless applications with frequency-agile radios," *NSDI*, 2010.

[31] K. Tan, H. Shen, J. Zhang, and Y. Zhang, "Enable flexible spectrum access with spectrum virtualization," *DySpan*, 2012.

[32] X. Zhang and K. G. Shin, "Adaptive subcarrier nulling: Enabling partial spectrum sharing in wireless LANs," *ICNP*, 2011.

[33] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," *MOBICOM*, 2008.

[34] I. Selesnick, M. Lang, and C. Burrus, "Constrained least square design of FIR filters without specified transition bands," *IEEE Trans. on Signal Processing*, 1996.

[35] D. Chu, "Polyphase codes with good periodic correlation properties," *IEEE Trans. on Information Theory*, 1972.

[36] "Linux ath9k driver." http://wireless.kernel.org/en/users/Drivers/ath9k.

[37] S. Hong and S. Katti, "DOF : A Local Wireless Information Plane," in *SIGCOMM*, 2011.

[38] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with Wi-Fi like connectivity," *SIGCOMM*, 2009.

[39] S. Hong, J. Mehlman, and S. Katti, "Picasso: Flexible RF and Spectrum Slicing," *Sigcomm*, 2012.

[40] "Proposed TGac Draft Amendment (Draft 3.1) for IEEE 802.11 Wireless LANs," Aug. 2012.

[41] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat, "Learning to share: narrowband-friendly wideband networks," *SIGCOMM*, vol. 38, no. 4, pp. 147–158, 2008.

[42] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, "Practical implementation of blind synchronization in NC-OFDM based cognitive radio networks," in *CoRoNet*, 2010.

[43] X. Liu, A. Sheth, M. Kaminsky, K. Papagiannaki, S. Seshan, and P. Steenkiste, "Dirc: increasing indoor wireless capacity using directional antennas," in *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, (New York, NY, USA), pp. 171–182, ACM, 2009.

[44] L. E. Li, Z. M. Mao, and J. Rexford, "Toward software-defined cellular networks," in *EWSDN*, 2012.

[45] S. Aditya and S. Katti, "Flexcast: graceful wireless video streaming," in *Mobicom*, 2011.

[46] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H. P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Comms Mag*, Feb 2011.

[47] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu, "Multi-cell mimo cooperative networks: A new look at interference," *JSAC*, Dec 2010.

[48] "Mobile access ve." `http://www.corning.com/cablesystems/nafta/en/mobileaccess/products/mobileaccess_ve.aspx`.

[49] "Crown castle das solutions." `http://www.crowncastle.com/das/solutions.aspx`.

[50] Z. Ma, M. G. Zierdt, J. Pastalan, A. B. Siegel, T. Sizer, A. J. de Lind van Wijngaarden, P. R. Kasireddy, and D. Samardzija, "Radiostar: Providing wireless coverage over gigabit ethernet," *Bell Labs Technical Journal*, vol. 14, no. 1, pp. 7–24, 2009.

[51] D. Samardzija, J. Pastalan, M. MacDonald, S. Walker, and R. Valenzuela, "Compressed transport of baseband signals in radio access networks," *Trans. Wireless Comms*, Sept 2012.

[52] D. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[53] K. C.-J. Lin, S. Gollakota, and D. Katabi, "Random access heterogeneous mimo networks," in *SIGCOMM*, 2011.

[54] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A first look at traffic on smartphones," in *IMC*, 2010.

[55] H. Ballani and P. Costa, "Towards predictable datacenter networks," *ACM SIGCOMM CCR*, 2011.

[56] S. Guha and J. Chandrashekar, "How healthy are today's enterprise networks," *IMC*, 2008.

[57] G. Wang and T. S. E. Ng, "The impact of virtualization on network performance of amazon ec2 data center," in *INFOCOM*, 2010.

[58] A. Gorokhov, "Antenna selection algorithms for mea transmission systems," in *ICASSP*, 2002.

[59] D. Tse and P. Viswanath, *Fundamentals of wireless communication.*

[60] D. Hostetler and Y. Xie, "Adaptive power management in software radios using resolution adaptive analog to digital converters," in *IEEE Computer Society Annual Symposium on VLSI*, 2005.

[61] N. Farrington, G. Porter, P.-C. Sun, A. Forencich, J. Ford, Y. Fainman, G. Papen, and A. Vahdat, "A demonstration of ultra-low-latency data center optical circuit switching," in *SIGCOMM*, 2012.

[62] D. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, 1952.

[63] K. Balachandran, J. Kang, K. Karakayali, and K. Rege, "Nice: A network interference cancellation engine for opportunistic uplink cooperation in wireless networks," *Wireless Communications, IEEE Transactions on*, vol. 10, no. 2, pp. 540–549, 2011.

[64] J. S. Vitter, “Design and analysis of dynamic huffman coding,” in *Foundations of Computer Science*, 1985.

[65] S. Gollakota, S. D. Perli, and D. Katabi, “Interference alignment and cancellation,” in *SIGCOMM*, 2009.

[66] P. Marsch and G. Fettweis, “Uplink comp under a constrained backhaul and imperfect channel knowledge,” *Trans. on Wireless Comms*, 2011.

[67] P. Baracca, S. Tomasin, and N. Benvenuto, “Constellation quantization in constrained backhaul downlink network mimo,” *Trans on Comms*, March 2012.

[68] A. del Coso and S. Simoens, “Distributed compression for mimo coordinated networks with a backhaul constraint,” *Wireless Communications, IEEE Transactions on*, Sept 2009.

[69] E. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *Trans on Information Theory*, Feb 2006.

[70] J. Paredes, G. Arce, and Z. Wang, “Ultra-wideband compressed sensing: Channel estimation,” *Selected Topics in Signal Proc*, 2007.

[71] P. Zhang, Z. Hu, R. Qiu, and B. Sadler, “A compressed sensing based ultra-wideband communication system,” in *ICC*, 2009.

[72] S. Kandula, S. Sengupta, and A. Greenberg, “The nature of data center traffic: measurements & analysis,” in *IMC*, 2009.

[73] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, “Vl2: a scalable and flexible data center network,” in *SIGCOMM*, 2009.

[74] Y. Chen, R. Griffith, J. Liu, R. H. Katz, and A. D. Joseph, “Understanding tcp incast throughput collapse in datacenter networks,” in *WREN*, 2009.

[75] E. Chai and K. G. Shin, “Sidekick: AP Aggregation Over Partially Overlapping Channels,” in *ICNP*, 2011.

[76] Y. R. Li, Li Erran and Tan, Kun and Viswanathan, Harish and Xu, Ying and Yang, “Retransmission Repeat: Simple Retransmission Permutation Can Resolve Overlapping Channel Collisions,” in *MobiCom*, 2010.

[77] A. Schulman, D. Levin, and N. Spring, "CRAWDAD data set umd/sigcomm2008 (v. 2009-03-02)." Downloaded from http://crawdad.cs.dartmouth.edu/umd/sigcomm2008, Mar. 2009.

[78] H. Rahul, F. Edalat, D. Katabi, and C. Sodini, "Frequency-aware rate adaptation and MAC protocols," in *Mobicom*, 2009.

[79] M. Vutukuru, K. Jamieson, and H. Balakrishnan, "Harnessing exposed terminals in wireless networks," in *NSDI*, 2008.

[80] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *Communications, IET*, vol. 1, no. 2, 2007.

[81] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Transactions on Communications*, vol. 45, no. 12, 1997.

[82] J. van de Beek, M. Sandell, and P. Borjesson, "ML estimation of time and frequency offset in OFDM systems," *IEEE Transactions on Signal Processing*, vol. 45, July 1997.

[83] M. Naik, A. Mahanta, R. Bhattacharjee, and HB, "An Automatic Blind Modulation Recognition Algorithm for M-PSK Signals Based on MSE Criterion," *E-business and Telecommunication Networks, Communications in Computer and Information Science*, vol. 3, pp. 257–266, 2007.

[84] "Technical report on rf channel characterization and system deployment modeling," Tech. Rep. JTC(AIR)/94.09.23-065R6, JTC (Air) Standards Contribution, Sept 1994.

[85] Y.-J. Chang, F.-T. Chien, and C. Kuo, "Opportunistic Access with Random Subchannel Backoff (OARSB) for OFDMA Uplink," in *GLOBECOM*, 2007.

[86] B. Roman, F. Stajano, I. Wassell, and D. Cottingham, "Multi-Carrier Burst Contention (MCBC): Scalable Medium Access Control for Wireless Networks," in *WCNC*, 2008.

[87] B. Roman and I. Chatzigeorgiou, "Evaluation of Multi-Carrier Burst Contention and IEEE 802.11 with fading during channel sensing," in *PIMRC*, 2009.

[88] M. Jain, J. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, P. Levis, S. Katti, and P. Sinha, "Practical, Real-time, Full Duplex Wireless," *MobiCom*, 2011.

[89] J. Proakis, *Digital Communications*. McGraw-Hill, 4 ed.

[90] E. Magistretti, K. Kant, B. Radunovic, and R. Ramjee, “WiFi-Nano: reclaiming WiFi efficiency through 800 ns slots,” *Mobicom*, 2011.

[91] A. Woo and D. E. Culler, “A transmission control scheme for media access in sensor networks,” *MOBICOM*, 2001.

[92] L. G. Roberts, “ALOHA packet system with and without slots and capture,” *ACM SIGCOMM CCR*, Apr. 1975.

[93] K. H. Kim and K. G. Shin, “On accurate and asymmetry-aware measurement of link quality in wireless mesh networks,” *IEEE/ACM Transactions on Networking*, vol. 17, Aug. 2009.

[94] K. Wu, H. Tan, Y. Liu, J. Zhang, Q. Zhang, and L. Ni, “Side channel: bits over interference,” in *MOBICOM*, 2010.

[95] K. Chebrolu and A. Dhekne, “Esense: communication through energy sensing,” in *MOBICOM*, 2009.

[96] A. Dutta, D. Saha, D. Grunwald, and D. Sicker, “SMACK: a SMart ACKnowledgment scheme for broadcast messages in wireless networks,” *ACM SIGCOMM CCR*, Oct 2009.

[97] J. Wang, Y. Fang, and D. Wu, “A Power-Saving Multi-Radio Multi-Channel MAC Protocol for Wireless Local Area Networks,” *INFOCOM*, 2006.

[98] A. Swami and B. Sadler, “Hierarchical digital modulation classification using cumulants,” *IEEE Transactions on Communications*, vol. 48, Mar. 2000.

[99] K. Ho, W. Prokopiw, and Y. Chan, “Modulation identification of digital signals by the wavelet transform,” *Radar, Sonar and Navigation, IEE Proceedings*, vol. 147, no. 4, 2000.

[100] B. Ramkumar, “Automatic modulation classification for cognitive radios using cyclic feature detection,” *IEEE Circuits and Systems Magazine*, 2009.

[101] “Fon.” Website. `http://corp.fon.com/en`.

[102] “Meraki.” Website. `http://meraki.com`.

[103] T. Dasilva, K. Eustice, and P. Reiher, “Johnny Appleseed: wardriving to reduce interference in chaotic wireless deployments,” in *MSWiM*, 2008.

[104] S. Kandula, K. Lin, T. Badirkhanli, and D. Katabi, "FatVAP: aggregating AP backhaul capacity to maximize throughput," in *NSDI*, 2008.

[105] D. Giustiniano, E. Goma, A. Lopez Toledo, I. Dangerfield, J. Morillo, and P. Rodriguez, "Fair WLAN backhaul aggregation," in *Mobicom*, 2010.

[106] E. Chai and K. Shin, "Low Overhead Control Channels in Wireless Networks," Tech. Rep. CSE-TR-574-11, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, 2011.

[107] A. Mishra, V. Shrivastava, and S. Banerjee, "Partially overlapped channels not considered harmful," in *IMC*, 2006.

[108] R. Chandra, P. Bahl, and P. Bahl, "MultiNet: connecting to multiple IEEE 802.11 networks using a single wireless card," in *INFOCOM*, 2004.

[109] a.J. Nicholson, S. Wolchok, and B. Noble, "Juggler: Virtual networks for fun and profit," *IEEE Transactions on Mobile Computing*, vol. 9, Jan. 2009.

[110] X. Xing, S. Mishra, and X. Liu, "ARBOR: hang together rather than hang separately in 802.11 wifi networks," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, Mar. 2010.

[111] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan, "Interactive wifi connectivity for moving vehicles," in *SIGCOMM*, 2008.

[112] P. Gilbert, E. Cuervo, and L. P. Cox, "Experimenting in mobile social contexts using JellyNets," in *HotMobile*, 2009.