Rice University

Distributed Protocols for Signal-Scale Cooperation

by

Christopher R. Hunter

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Dr. Ashutosh Sabharwal, *Chair*
Associate Professor of Electrical and Computer Engineering

Dr. Behnaam Aazhang
J.S. Abercrombie Professor of Electrical and Computer Engineering

---

Dr. Edward W. Knightly
Professor of Electrical and Computer Engineering

Dr. David B. Johnson
Professor of Computer Science

Houston, Texas
May 2012

Abstract

Distributed Protocols for Signal-Scale Cooperation

by

Christopher R. Hunter

Signal-scale cooperation is a class of techniques designed to harness the same gains offered by multi-antenna communication in scenarios where devices are too small to contain an array of antennas. While the potential improvements in reliability at the physical layer are well known, three key challenges must be addressed to harness these gains at the medium access layer: (a) the distributed synchronization and coordination of devices to enable cooperative behavior, (b) the conservation of energy for devices cooperating to help others, and (c) the management of increased inter-device interference caused by multiple spatially separate transmissions in a cooperative network. In this thesis, we offer three contributions that respectively answer the above three challenges.

First, we present two novel cooperative medium access control protocols: Distributed On-demand Cooperation (DOC) and Power-controlled Distributed On-demand Cooperation (PDOC). These protocols utilize negative acknowledgments to synchronize and trigger cooperative relay transmissions in a completely distributed manner. Furthermore, they avoid cooperative transmissions that would likely be unhelpful to the source of the traffic.

Second, we present an energy conservation algorithm known as Distributed Energy-Conserving Cooperation (DECC). DECC allows devices to alter their cooperative behavior based on measured changes to their own energy efficiency. With DECC, devices become *self-aware* of the impact of signal-scale cooperation – they explicitly monitor their own performance and scale the degree to which they cooperate with others accordingly.

Third and finally, we present a series of protocols to combat the challenge of inter-device interference. Whereas energy efficiency can be addressed by a self-aware device monitoring its own performance, inter-device interference requires devices with *network awareness* that understand the impact of their behavior on the devices around them. We investigate and quantify the impact of incomplete network awareness by proposing a modeling approximation to derive relaying policy behaviors. We then map these policies to protocols for wireless channels.

Acknowledgements

Relatively speaking, I have been a student at Rice a long time – from undergraduate study through the completion of the PhD. In my time, I have seen many things change around campus. New buildings, new research directions, and of course, many new faces. One thing that never changed, however, was the unwavering support from the people around me. To begin, I would like to thank my advisor Ashu Sabharwal for all of his help throughout my study. Ashu knows how to give just the right feedback to let his students stay on track but have freedom to explore their work in whatever direction they want. I cannot overstate how lucky I am to have had that guidance.

Next, I would like to thank all of my friends around the department. At times our interactions were entirely technical in nature – hammering out ideas on whiteboards in one anothers' offices was frequent. But I think more importantly, we were able to take those discussions into more social settings. I got some of my best work done musing about research problems over beers at Valhalla. In particular, Gareth Middleton and Patrick Murphy played no small role in my success in this research program. Their insights and abilities to brainstorm my research problems with me were invaluable. Furthermore, their own graduations were inspirational and pushed me to follow through on my work and defend.

Last but certainly not least, I want to thank my family. My parents have been there for me for the last 28 years. At every stage, their uncon-

ditional encouragement has helped me more than I can express. My wife, Juliana, has been by my side since high school. Both of our life goals have evolved a lot since we met all those years ago, but we have supported one another through every twist and turn. I know I would not have been able to persevere without her. Finally, our dogs Chief and Tort. Chief has been with us for 7 years and is very wise – I am pretty sure he could have done my PhD defense for me if he could speak. Tort, on the other hand... at least he means well.

To my family – Juliana, Reg, Jan, Chief, and Tort

Contents

List of Figures

List of Tables

Chapter 1

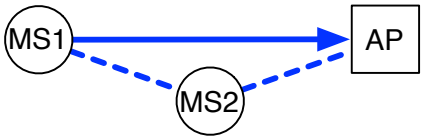
Introduction

As demand for mobile data increases, it is becoming increasingly important to improve network spectral efficiency. The so-called "spectrum crunch" [1] has necessitated a rethinking of the physical layer beyond the traditional standpoint of establishing a reliable connection between only two devices – the transmitter and the receiver. Notably, cooperation between devices at the signal level (see e.g. [2]) can be used to harness the antennas of multiple devices to increase a mix of rate, reliability, and transmission range. Cooperation leverages MIMO-like spatial-diversity benefits with only single-antenna devices and also achieves topological benefits where one device may have a better link quality than its neighbors.

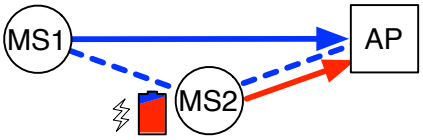
From a physical layer perspective, signal-scale cooperation is well understood. Information theoretic investigations of signal-scale cooperation are abundant and have shown its potential for multifold network capacity improvement [3, and references therein]. Hardware prototypes of these concepts have been presented showing that large improvements in packet error rate can be achieved with cooperation [4–10].

In contrast, the understanding of signal-scale cooperation from a distributed medium access control (MAC) perspective is limited. In a distributed, random access MAC, devices cannot rely on centralized coordination to determine the conditions un-

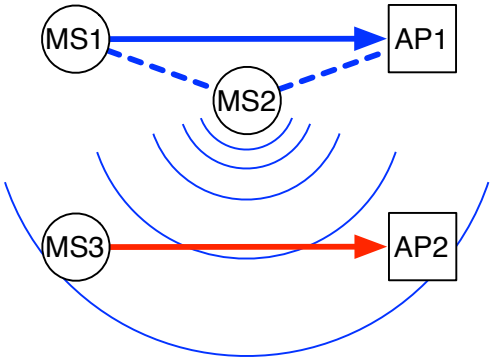
der which cooperation should occur; instead, they must make that decision on their own. Several key challenges must be addressed to realize the gains of signal-scale cooperation in these distributed wireless networks.



(a) Challenge 1: Distributed Synchronization and Coordination



(b) Challenge 2: Energy Efficiency



(c) Challenge 3: Inter-device Interference

Figure 1.1: Challenges for signal-scale cooperative MACs.

Challenge 1: Distributed Synchronization and Coordination

The first challenge of realizing signal-scale cooperation is the synchronization and coordination of devices such that providing cooperative aid to another device actually provides the predicted benefit. A key assumption in communication analyses is that cooperative relay transmissions can be temporally aligned to symbol level accuracy [3] – this is required to ensure that cooperative transmissions are beneficial rather than interfering. In practice, this synchronization is very challenging in devices that operate with independent clock sources. The cyclic prefix of an 802.11 OFDM waveform with 20 MHz of bandwidth is only 0.8 μs – offsets in transmission times of multiple cooperative transmitters will degrade the ability to handle multipath fading.

Furthermore, even if cooperative transmissions can be aligned, knowing *when* to engage cooperation is imperative for there to be any performance gain. Cooperation is inherently a two time-slot operation due to causality – a cooperative relay must receive the information it is to forward before it can do so. If non-cooperative communication can deliver a packet, then using a second time slot to perform cooperative relaying will needlessly increase the amount of time a transmission occupies the wireless medium.

Figure 1.1(a) illustrates this challenge of synchronization and coordination. For mobile station MS2 to provide cooperative assistance to MS1, MS2 must synchronize its transmission with MS1 and also determine the conditions under which acting as a cooperative transmitter would be beneficial.

Challenge 2: Energy Efficiency

Devices acting as cooperative relay transmitters for their neighbors spend energy transmitting data that is not relevant to them, thereby creating a fundamental energy cost to cooperation. In many network scenarios, the energy cost of cooperation can outweigh improvement for particular devices. Figure 1.1(b) illustrates this second challenge. In this scenario, MS2 may be willing to assist MS1, but not if doing so substantially harms MS2's ability to deliver its *own* traffic to the access point AP. If MS2 spends energy assisting MS1 by transmitting on its behalf, cooperation may degrade MS2's battery life. To harness cooperative gains in a network of non-altruistic users who are not selflessly dedicated to helping others, a cooperative protocol operating on a device needs to determine the degree to which cooperation is benefitting or harming the energy efficiency of that device and react accordingly.

Challenge 3: Inter-device Interference

The third challenge of realizing signal-scale cooperation is due to the fact that cooperative relay transmissions can increase interference in a wireless network. The spatial footprint of a cooperative link is larger than that of a non-cooperative link due to the presence of additional transmitters. This has a significant impact on spatial reuse. Figure 1.1(c) illustrates this challenge. Even if MS2 is completely selfless and unconcerned about its own performance, its aid to MS1 may come as a cost to MS3. Wireless transmissions are inherently broadcast – consequently, transmitting on behalf of MS1 can increase MS1's effective spatial footprint and impinge additional interference on MS3's flow of traffic. A cooperative protocol should be cognizant of this negative impact on the rest of the network and only engage cooperation when it is socially responsible to do so.

1.1 Summary of Contributions

In this thesis, we present three primary contributions that each target one of the three aforementioned key challenges. First, we present two novel cooperative MAC protocols known as Distributed On-demand Cooperation (DOC) and Power-controlled Distributed On-demand Cooperation (PDOC) – originally presented in [11] and [12] respectively. These MACs answer the challenge of distributed synchronization and coordination by utilizing negative acknowledgments (NACKs) to trigger cooperative relay transmissions. These NACKs serve two purposes: (a) they give a frame of reference for a source and relay to synchronize their transmissions and (b) they allow the destination to request cooperation only when a transmission has failed and retransmission would be required anyway. Furthermore, PDOC builds upon DOC to allow devices to calculate how much transmission power is required for any given cooperative transmission to succeed, thereby allowing devices to disable cooperation

when it is unlikely to do any good and scale back their power when doing so will not adversely affect the outcome. These cooperative MACs are presented in Chapter 3.

Second, we present an energy conservation algorithm known as Distributed Energy-Conserving Cooperation (DECC). This protocol answers the challenge of energy efficiency by allowing devices to alter their cooperative behavior when signal-scale cooperation causes them more harm than benefit. DECC builds on top of the PDOC protocol and actively measures energy efficiency both with and without cooperation. Using a standard proportional-integral control loop, DECC then dictates a "cooperative resource" to PDOC to indicate whether cooperative transmissions for others should be increased or decreased. PDOC then utilizes its knowledge of required transmission powers for cooperative transmissions to avoid transmitting the highest power (and therefore most costly) packets first. With DECC, devices become *self-aware* of the impact of signal-scale cooperation – they explicitly monitor their own performance and scale the degree to which they cooperate with others accordingly. This energy-conserving protocol is presented in Chapter 4.

Third and finally, we present a series of protocols to combat the challenge of inter-device interference. Whereas energy efficiency can be addressed by a self-aware device monitoring its own performance, inter-device interference requires devices with *network awareness* that understand the impact of their behavior on the devices around them. We investigate and quantify the impact of incomplete network awareness by proposing a modeling approximation to derive relaying policy behaviors. We then map these policies to protocols for wireless channels. This contribution is presented in Chapter 5 and is based off its original presentation in [13].

In sum, these contributions allow devices to intelligently select when and how to act as a relay, tailoring their cooperative efforts based on the broad effects cooperating can have on devices in the network.

Chapter 2

Background

In this chapter, we provide background material foundational to our primary contributions in Chapters 3 through 5. We provide a brief overview of signal-scale cooperation and an overview of the Wireless Open-Access Research Platform (WARP). We then describe our WARPMAC framework contribution (novel to this thesis) that is used by our later implementations as well as by many research groups using WARP around the world.

2.1 Overview of Signal-Scale Cooperation

The first study of cooperative devices was for the information theoretic relay channel [14–16], but the concept flourished in the literature only after the communications research community discovered the tremendous gains in reliability and rate that multiple-input multiple-output (MIMO) communications can provide [17]. MIMO communications hinges on the fact that different, sufficiently separated antennas on a device observe different channel conditions between a transmitter and receiver. In effect, multiple antennas at a transmitter or receiver create multiple independent "spatial streams" for data transmission. MIMO can exploit this fact by either (a)

loading additional symbols into these spatial streams (i.e. multiplexing) or (b) sending redundant symbols across spatial streams to improve reliability (i.e. diversity) [18].

MIMO technology has been adopted into virtually all modern communications standards (e.g. 802.11n, 3GPP LTE-Advanced). However, multiple antennas are generally only found on larger network devices such as access points and base stations. These large devices can fit multiple antennas inside them and still maintain sufficient separation between antennas for the spatial streams to be created. Simply put, portable wireless devices such as smartphones and tablets are physically too small to contain the large antenna arrays required by MIMO communications.

Signal-scale cooperation is a response to this problem. Rather than rely on single devices containing many antennas, cooperation harnesses MIMO-like spatial streams created by multiple single-antenna devices. Pioneering studies in the context of ergodic systems (e.g. CDMA) have been presented [19, 20]. Further work has transitioned to shorter packet duration systems that are delay-constrained (e.g. Wi-Fi) [2]. In that work, the popular "amplify-and-forward" (AF) and "decode-and-forward" (DF) cooperative strategies were analyzed. AF cooperative relays forward previously-saved I and Q samples without performing any kind of decoding on them while DF relays first decode the message and then re-encode it before transmitting.

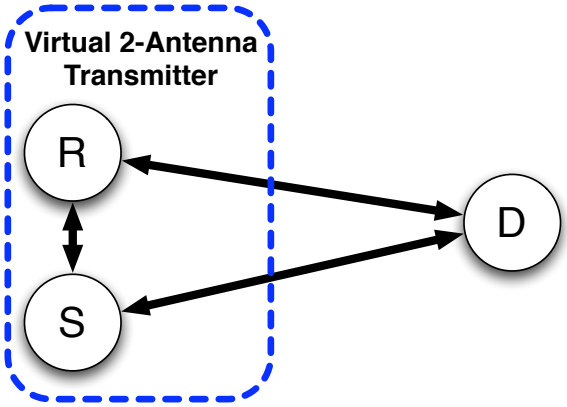


Figure 2.1: Cooperative devices can act as a virtual MIMO transmitter to improve transmission reliability.

Figure 2.1 shows an example illustration where two mobile devices can pool their resources together to act like a single virtual MIMO transmitter. By orthogonalizing source and relay transmissions through the Alamouti 2×1 space-time block code (STBC) [21], a $40\times$ improvement in packet error rate has recently been demonstrated via a study on a hardware testbed [10]. An excellent book-length survey of signal-scale cooperative techniques has been published [3].

2.2 Hardware Platform Overview

Signal-scale cooperation is a challenging subject for empirical investigation because it requires novel techniques at multiple layers of the networking stack – all the way down to waveform construction. Commodity hardware such as Wi-Fi is ill-suited for understanding the effects of cooperation because the PHY cannot be changed and the MAC layer offers very limited design freedom. Since we do not want to be constrained by current Wi-Fi implementation frameworks, we choose Rice University's Wireless Open-Access Research Platform (WARP) [22] for our study as it allows for (a) custom physical layer behavior and (b) real-time operation to preserve the time scales over which multiple devices actually interact in practice. Details of the platform have been presented in [23–25].

2.2.1 Physical Layer Designs

WARP provides a large Xilinx FPGA for performing intensive processing tasks such as a real-time physical layer. The WARP open-source project provides access to a custom amplify-and-forward [4] and decode-and-forward [10] physical layer. In this thesis, we rely heavily on these physical layers for our MAC-level contributions.

2.2.2 Medium Access Layer Designs

To enable the study of real-time MAC layers, we have constructed the WARPMAC framework, which abstracts away many hardware details and allows the MAC designer to focus on programming the custom state machine associated with his or her MAC. We have published the details of this framework in [23] and the latest version of this framework is available on the WARP website [26]. Furthermore, we have developed and continue to maintain an example application of the WARPMAC framework in the form of the WARP OFDM Reference Design [27]. For this reference design, we have implemented a complete Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) MAC layer that is used as the foundation to many implementations by research groups that use WARP.

A number of research projects have partially addressed the need for a platform for novel MAC implementations by overwriting behavior of commercial 802.11 chipsets [28–30]. However, these designs were intended to provide enhancements to existing 802.11 MAC implementations rather than to provide a general MAC development environment. Enhancing these existing projects, the SoftMAC project allows users to modify the format of transmitted packets via custom drivers that exploit reverse-engineered details about the Atheros wireless chipset [31]. While this provides a development environment that generally spans the space of MAC algorithms possible on an 802.11 PHY, it offers limited functionality for research in the larger space of clean-slate MAC-PHY pairs, such as signal-scale cooperation.

WARPMAC provides high-level, low-breadth tools for derivatives of standard random access protocols, and low-level, high-breadth tools for ground-up designs. This structure allows for user-defined abstraction of hardware details and maximum flexibility for a large class of algorithms. WARPMAC provides low and high-level functions to aid in the design of new MAC layers.

Low-Level Functions: These functions provide a low layer of abstraction by incorporating many system driver calls into single functions. For example, a MAC designer has access to a transmission function that takes a packet as an input and returns only when that packet has been sent over the air. Internally, the function places the radio into a transmit mode, loads the payload into the PHY, and starts the PHY processor. Another example of low-level functionality at this layer is callback registration. For example, users can attach functions to the framework such that custom routines will be called upon the reception of a packet that passed a cyclic redundancy check (CRC), the reception of a packet that failed CRC, and the expiration of a timer.

High-Level Functions: At the highest level, the WARPMAC framework provides functions to implement a large class of random access protocols. For example, users can set timers for a timeout or backoff window. For the latter, the function internally manages the binary exponential backoff seen in random access MACs. Additionally, WARPMAC engages a carrier sensing module we have built into the fabric of the FPGA to automatically pause and resume the backoff depending on medium utilization.

The choice of which layer of the WARPMAC organization to use is completely dependent on the requirements of a user's MAC algorithm. For example, because most random access protocols require a binary exponential backoff to deal with medium contention, it is likely that the high-level function capable of such a behavior is reusable in novel random access MACs. However, for MAC algorithms substantially different from basic ALOHA [32] or CSMA/CA, it is likely that low-level parts of the stack will be required.

Figure 2.2 shows how WARPMAC provides the "glue" between the hardware-

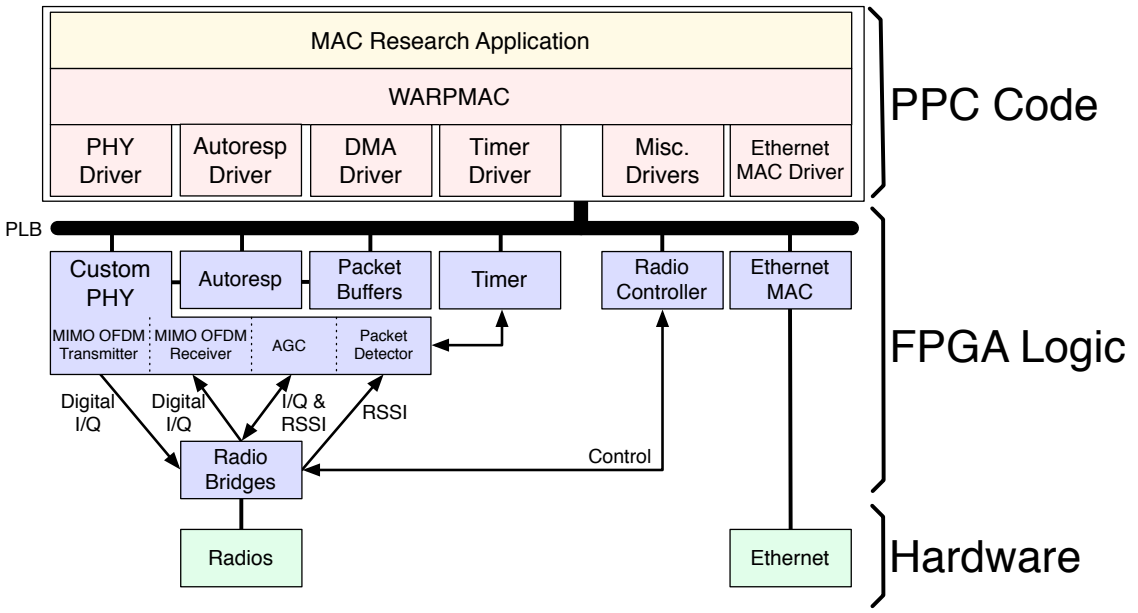


Figure 2.2: WARP real-time designs use a combination of PowerPC and FPGA logic processing resources.

specific implementation details and the top-level MAC research application. A full API is available online [33]. In this thesis, we heavily rely on WARPMAC for the implementation of cooperative protocols.

Nearly all of the MAC state is implemented directly inside the PowerPC (PPC). For time-critical operations, we have also built an "autoresponder" MAC accelerator module into the fabric of the FPGA to handle transmission of packets very quickly and deterministically after the reception of others. The details of this design were originally presented in [34] and are included in Chapter 3. With the autoresponders, we have been able to synchronize source and relay transmissions such that they occur within no more than 100 ns of each other.

Chapter 3

Distributed Synchronization and Coordination

In this chapter we present two novel MAC protocols for leveraging signal-scale cooperation. Included in these presentations are in-depth characterization studies using WARP. We extensively employ these MACs in our contributions presented later in this thesis.

First, in Section 3.1, we present the Distributed On-demand Cooperation (DOC) MAC protocol. DOC uses negative acknowledgment (NACK) feedback from a destination to trigger cooperation from a source and relay. This feedback provides a frame of reference for the source and relay to synchronize their transmissions to within 100 ns of each other – well within the 1.6 μs cyclic prefix of our OFDM implementation. Furthermore, cooperation is enabled "on-demand" via receivers; if cooperation is not needed because direct links have high enough SNR, then DOC gracefully falls back to the IEEE 802.11 DCF-like behavior.

Second, in Section 3.2, we present the Power-controlled Distributed On-demand Cooperation (PDOC) MAC protocol. PDOC considers the problem of conserving energy at cooperative relays by employing relay power control. PDOC uses the NACKs from DOC to piggyback channel information about the source-to-destination link to the relay node, allowing the relay to scale its transmission power accordingly.

3.1 Distributed On-demand Cooperation (DOC)

In this section, we present a real-time testbed implementation of a novel distributed cooperative protocol known as Distributed On-demand Cooperation (DOC). This work was originally presented in [11]. Our implementation allows throughput gains of *at least* 20% from cooperation in many topologies of interest.

One of the key features of DOC is the use of explicit negative acknowledgements (NACKs) to signal the need for physical layer cooperation, making the use of the cooperative mode *on-demand*. A NACK is triggered only on channel-induced errors in the payload. Thus, if the destination node can decode the more-reliable header but not the payload due to a channel fade, then the destination triggers a cooperative mode.

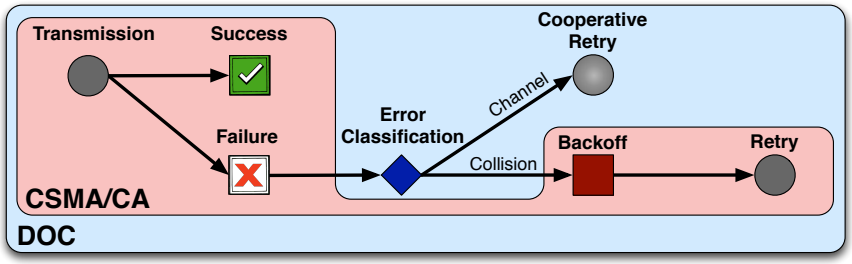


Figure 3.1: DOC encompasses the behavior of CSMA/CA and only engages cooperation when it is needed.

Another important implication of the protocol structure is that it allows nodes to assume the role of a source or relay on per-packet timescales. This ensures that cooperative mode gains are automatically realized when relaying is available, and the system automatically reduces to a traditional non-cooperative CSMA/CA system when relaying is not available without requiring any adjustments at the physical or MAC layers. Figure 3.1 highlights how DOC falls back to the traditional access mechanism when cooperation is unable to help.

The gains are significant, achieving 20+% throughput improvement over non-relay

assisted links. An interesting observation from our testbed evaluation is that gains are near-maximum when the source and relay are close to each other. This observation has a significant usage implication, allowing the *same* user to employ two commonly used personal devices, such as a laptop and a smartphone, to cooperate with each other. Such a usage scenario could also obviate numerous concerns with cooperative communications, such as security, privacy, and incentives for cooperation. The latter concern is discussed at length in Chapter 4.

As part of the overall Rice WARP research project, all implementation source files for the DOC MAC and PHY design are available in the WARP repository [22].

3.1.1 Related Work

The work on cooperative MAC protocols is fairly sparse [8, 35–39]. The work in [35] assumes nodes can be perfectly synchronized at negligible cost. In DOC, we explicitly address the challenges of synchronization at both the PHY and MAC layers. The protocols in [36–39] are designed for distributed systems but rely on explicit RTS/CTS handshakes in order negotiate cooperative transmissions. In contrast, DOC is completely connection-free; relays are not negotiated with on a per-packet basis.

Additionally, we note that there are very few implementations of cooperative protocols [4–7, 10]. In each, the authors implemented cooperative physical layers on software defined radio platforms but focused exclusively on the physical layer and did not consider the MAC or higher network protocols. Further, in [5, 6] the implementation was geared towards narrowband systems with approximately 68 kHz bandwidth. In contrast, our implementation is characterized by end-to-end throughput gains and includes both a custom MAC and PHY. It also operates with an RF bandwidth of 10MHz with straightforward extensions up to 20MHz, the operating regime of most wideband wireless networks. In [7], cooperative diversity is demon-

strated on a hardware platform, but the diversity improvements are analogous to MIMO receiver selection diversity as no waveform-level combinations are employed. In [40], the protocol is specified as both a PHY and a MAC, but only the former is actually implemented due to hardware constraints. By necessity, higher-layer behavior is investigated via simulations and simplified asymptotic analysis.

3.1.2 Protocol Description

In this section, we describe DOC's behaviors and the requirements they impose on a real implementation. Our actual implementation is discussed in Section 3.2.5.

The key mechanism in DOC is the identification of when a cooperative mode is needed. We exploit the fact that packet headers are often much better protected than payloads. For example, in IEEE 802.11a, PHY headers are transmitted at 6Mbps, while payloads are transmitted with a peak data rate of 54Mbps, with significantly higher error probabilities[1]. *Even if* headers and payloads are both transmitted at the slowest rate, payload errors due to channel effects are more probable than header errors since payloads are generally much longer than headers. In DOC, whenever a destination receives a packet with a valid header but with a payload error due to channel fading and noise, it sends a NACK to trigger a cooperative retransmission. Since payload errors could occur due to mid-packet collisions (due to hidden terminals, for example), DOC can utilize a mid-packet collision detector to avoid triggering a cooperative phase if a mid-packet collision was the cause of payload error. Details of this collision detector are presented in Appendix A.

The MAC-level behavior of DOC can be broken up into three basic branches, as shown in Figure 3.2. One branch handles packets passed down from a higher

[1]In IEEE 802.11, the MAC header (e.g. the address and packet type fields) is actually sent at the full payload rate. For our protocol, these fields must be sent at the slower base rate like the PHY header.

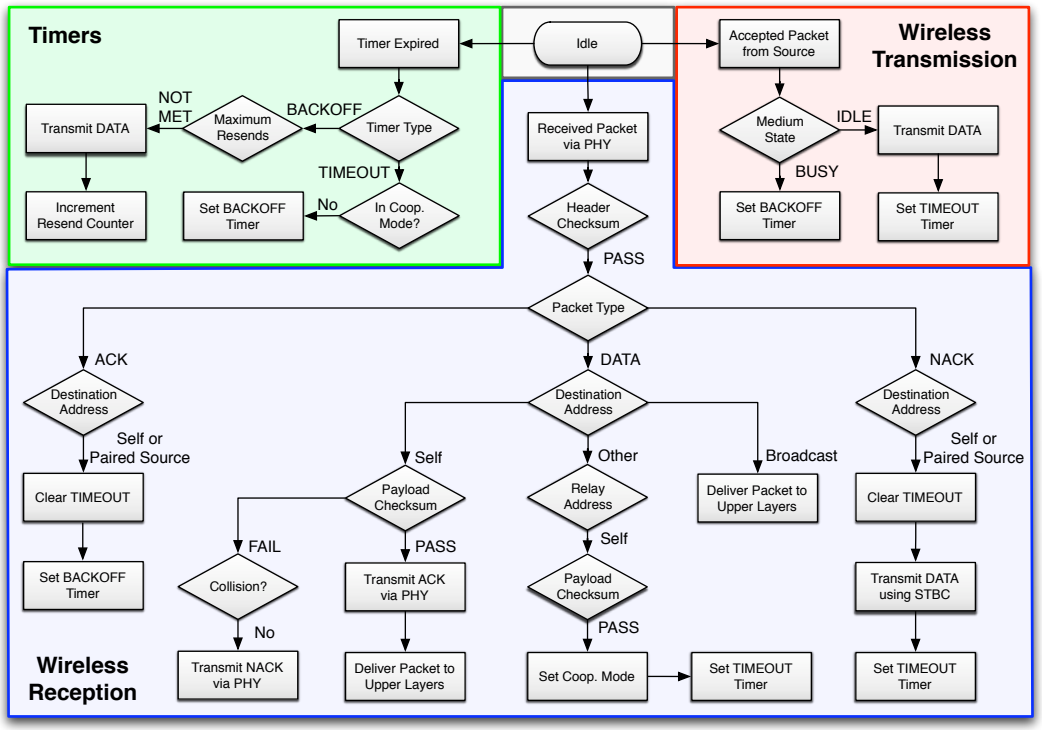


Figure 3.2: DOC state machine.

layer ("Wireless Transmission"), one branch handles the reception of packets from the wireless interface ("Wireless Reception"), and one branch handles timer events ("Timers"). We now describe the behavior of the protocol in each of these branches in detail.

3.1.2.1 Wireless Transmission

This branch handles the wireless transmission of packets from a higher layer. First, the protocol checks the state of the medium. If idle, DOC transmits the packet and starts a timeout timer to wait for an acknowledgment. If the medium is busy, DOC enters a random backoff period before attempting to transmit. After setting the backoff timer, the protocol implicitly returns back to an idle state. For readability, all return-to-idle transitions in Figure 3.2 are implicit in this way. Notice that this behavior is identical to traditional CSMA/CA basic access mechanisms like the IEEE

802.11 DCF.

3.1.2.2 Wireless Reception

In general, this branch handles wireless receptions both for one's self and for another node (in the case one's self is selected as a relay for a cooperative retransmission). The header, which contains the packet's length, rate, and addresses, is protected by its own checksum. If this checksum passes, the MAC assumes it can trust the metadata of the packet, allowing the DOC state machine to potentially recover from packet losses due to channel fades.

3.1.2.3 Timers

This branch handles timer events. Specifically, DOC, like traditional CSMA/CA MACs, contains two types of timers: a timeout and a random backoff. DOC is designed to gracefully degrade back to standard IEEE 802.11-like behavior when errors occur that cooperation cannot help (e.g. collisions). The timer states support these identical behaviors that are shared between the protocols.

3.1.3 FPGA Implementation

In order to realize a complete real-time cooperative MAC and PHY, we use WARP. The real-time physical layer we employ is the custom cooperative physical layer presented in our earlier publication [34]. In this section, we use our PHY's amplify-and-forward mode for simultaneous source and relay transmissions within the same band. The implementation makes use of the WARPMAC framework's MAC accelerators that allow for source-relay synchronization within a single sample period, also presented in our earlier publication [34].

The primary goal of our design is to enable cooperation in random access networks, which requires that nodes be able to trigger cooperative transmissions in response to receiving data or control packets from other nodes in the network. This requirement imposes very strict synchronization tolerances on the latency between receiving a wireless packet and transmitting one in response. In our previous WARP designs, the Rx-Tx turnaround was controlled entirely from the MAC software running in the FPGA's PowerPC core, and the turnaround latency was approximately $24 \pm 0.8 \mu$s.

This 1.6μs window is the same length as the guard interval used in our PHY design to protect OFDM symbols from inter-symbol interference. If this level of jitter were present in the start times of cooperative transmissions, a significant fraction of the OFDM cyclic prefix would be consumed by the synchronization uncertainty, leaving too little guard interval to protect against multipath in the propagation environment.

To combat this problem, we designed a new FPGA subsystem to manage all Tx-Rx and Rx-Tx transitions. This subsystem has two major functions. First, it controls the pins that enable the transmit and receive paths through the WARP hardware's radio transceiver. Second, it contains logic that can automatically initiate a packet transmission in response to a packet reception. The conditions for triggering the transmission and the contents of the response packet are programmed by the MAC software and can be changed on per-packet timescales. This subsystem essentially functions as a MAC "accelerator," allowing protocol behaviors to be specified in C code but executed by dedicated hardware resources at fast and deterministic timescales.

The transmitted packet's header is constructed on-the-fly from both static values provided by user code and values pulled from the header of a received packet. The packet templates are defined by user code and can be updated at run-time. This design enables the implementation of high-performance MAC protocols without having

to implement the protocol itself in the FPGA fabric.

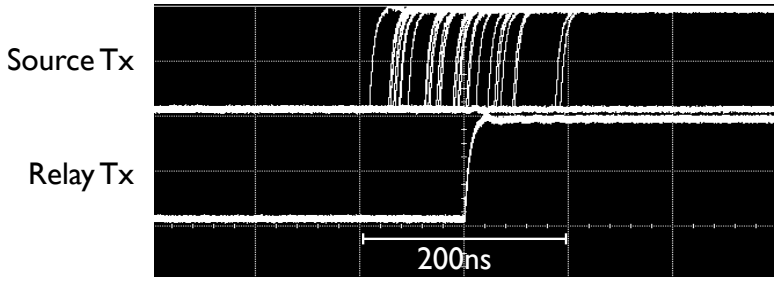


Figure 3.3: Real-time capture of source transmission start relative to relay transmission start.

Figure 3.3 illustrates the jitter between the source and relay transmissions as measured in real-time. The oscilloscope here was triggered by the relay transmission and displays the start of multiple corresponding source transmissions. The difference in transmit start times is clearly bounded to ± 1 sample period (100ns). This bound on transmit offsets is expected, given that nodes operate with independent sampling clocks and use correlation against incoming preambles to define the timing of the receiver's processing.

In the case of overlapping transmissions, the first transmission to arrive at the destination will trigger reception, thus limiting the offset to one sample period (plus any propagation time differences) for a given reception. The issue of propagation times is also tractable in the kinds of networks where we envision the cooperative transceiver could be employed. In our implementation, a single sample of cyclic prefix provides tolerance for propagation distance offsets of 100+ feet (wireless signals propagate at $\approx$1 ft/ns). This issue would require more detailed consideration in systems operating over longer distances or with more aggressive cyclic prefix lengths. For example, if nodes were aware of their relative distances, cooperating nodes could delay their transmissions to align their arrivals at the destination. Our automatic response block provides a user-programmable delay to accommodate exactly such a scheme.

Table 3.1 lists the key parameters of our DOC implementation. Table 3.2 lists

the FPGA resource usage for the full DOC MAC/PHY design as implemented in the WARP FPGA Board's Xilinx XC2VP70 FPGA.

Table 3.1: MAC/PHY Parameters for DOC Implementation

Carrier Frequency	2427MHz
Transmit Power	10dBm
RF Bandwidth	10MHz
OFDM Symbol	64 subcarriers
OFDM Cyclic Prefix	1.6μs
Header Length	24 bytes
Header Rate	BPSK at 6Mbps
Payload Length	1470 bytes
Payload Rate	QPSK at 12Mbps
DATA Packet Duration	1.06ms
ACK/NACK Packet Duration	80μs
DATA-ACK Turnaround Time	17μs

Table 3.2: FPGA Resource Usage

FPGA Resource	**Utilization**
Logic Slices	23283 of 33088 (70%)
18x18 Multipliers	161 of 328 (49%)
18kb Block RAMs	304 of 328 (92%)

3.1.4 Experimental Design

Our experimental setup consists of three WARP nodes, each consisting of a WARP FPGA and radio board. Every node is configured with the same DOC MAC/PHY design, implementing the full MAC protocol and OFDM transceiver. For each trial, the source node generates random packets with 24 byte headers and 1470 byte payloads. The source is fully backlogged, immediately initiating a new transmission whenever the previous one succeeds (i.e. received ACK + backoff) or times out (timeout + backoff). Each data point represents a trial in which at least 60MB of data is transmitted by the source node, spanning a time sufficiently long to experience thousands of random channel coefficients.

In order to realize a repeatable propagation environment, we use the Azimuth ACE 400WB wireless channel emulator [41, 42]. This emulator is designed explicitly for testing high-performance MIMO systems and is widely used in industry for wireless systems characterization and standards compliance testing. The wireless interface of each WARP node is connected to the emulator, as shown in Figure 3.4. A custom Tcl script is used to control the WARP nodes (to gather performance statistics) and the Azimuth Director-II API (to configure the emulator).

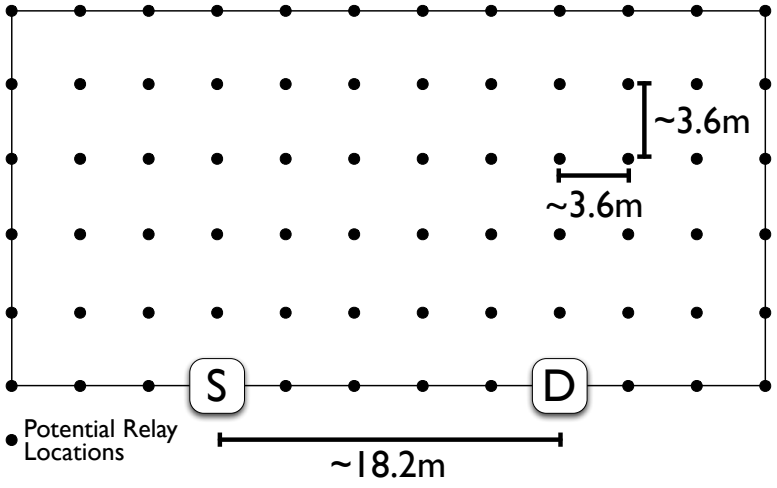
We choose the channel model for our tests from the TGn family of models proposed for the IEEE 802.11n standard [43]. Specifically, we use TGn model B, which models a channel with nine taps and delay spread of 80 ns. This model realistically captures the scaling, phase, and dispersion effects of scattering in an indoor environment. It specifies a Doppler spread of 2.6 Hz, which we adopt in our experiments.

For every test, the three wireless links (S-D, S-R, and R-D) use the same model and have independent instantaneous fading coefficients. The emulator allows each link to be configured with an average SNR spanning a 40dB range. By sweeping various average SNRs, we are able to emulate a large number of physical topologies.

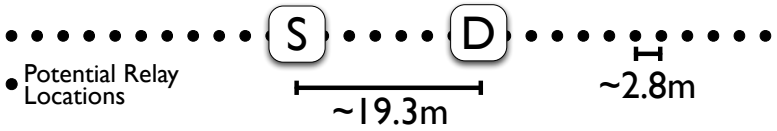
Figure 3.4: A channel emulator is used to control average SNR between each node as well as apply random fading elements to each link.

3.1.4.1 Topologies

The channel emulator enables our experiments to exercise arbitrary topologies by choosing various average SNRs between each node. We choose a source-destination separation of approximately 20 m, yielding a regime where many packets are being lost due to channel effects. With this distance (and thus SNR) fixed, we run experiments that sweep the position of the relay in a space around the source and destination nodes. As shown in Figures 3.5(a) and 3.5(b), these experiments take two forms. In the first topology, we test 72 relay locations uniformly distributed in the 2-D space surrounding the source and destination. We test only those relay positions "north" of the source-destination line, recognizing the results will be symmetric about this axis. In the second topology, we constrain the relay to locations along the line connecting the source and destination and conduct 30 trials at equally spaced points.



(a) 2-D Topology



(b) 1-D Topology

Figure 3.5: We consider a variety of relay locations in our studies.

In the following figures, the independent variable is presented as distance in me-

ters. The actual experimental variable, as configured in the channel emulator, is the average SNR along each wireless channel. The mapping of path loss to distance requires selection of a path loss exponent representative of the propagation environment [44]. We choose a nominal exponent of 2.1, representing an indoor setting with moderate scattering and mapping to intuitive distances for indoor wireless networking. The exponent experienced by over-the-air transmissions will be heavily dependent on the physical environment. The gains we identify at various path losses would occur even if the mappings to distances were adjusted.

3.1.4.2 Metrics

We use end-to-end throughput as the metric for our experiments. While bit error rate is a common metric for analyses of cooperative physical layer designs, throughput depends on both the BER and MAC overhead, and thus better captures the impact of both physical and MAC layer effects on performance of the full system. Our measurement of throughput is calculated using the number of bytes successfully delivered to the destination. Our design does not count duplicate receptions (i.e. when an ACK packet is lost) towards throughput by means of sequence numbers in every packet header. We also track how often the relay participates in a given test. Our current implementation prevents the relay from transmitting a given DATA packet multiple times. Thus, we calculate the probability of cooperation as the ratio of packets transmitted by the relay to the number of unique packets transmitted by the source.

3.1.4.3 Real-time Observations

In order to monitor real-time node interactions, we use an oscilloscope to capture digital control signals driven by each node's FPGA. The signals are driven by FPGA logic and allow monitoring of MAC/PHY state transitions in real-time without inter-

fering with the protocol itself. Figure 3.6 illustrates some key interactions. The figure shows four signals: source transmission, relay transmission, destination transmission, and a flag indicating an error-free packet reception at the destination. Transmissions of ACK and NACK can be disambiguated by whether the destination flagged the preceding DATA packet as good. The duration of each transmission signal in these plots is identical to the duration of the corresponding RF transmission.

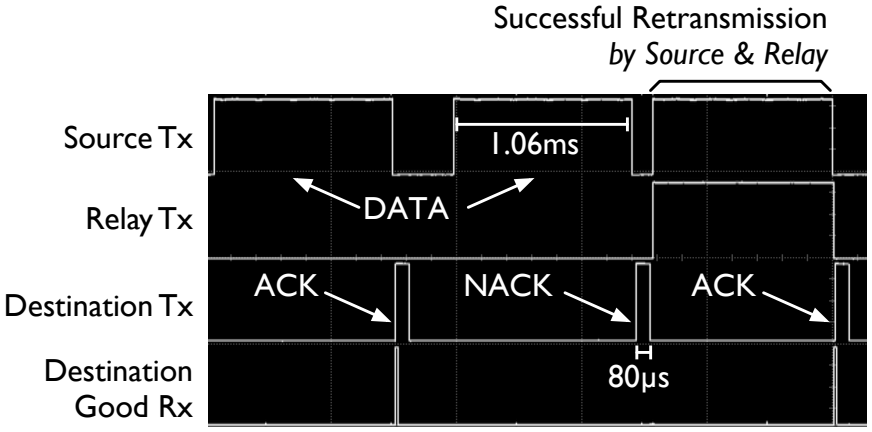


Figure 3.6: Oscilloscope capture showing real-time transmission and reception events with a successful cooperative retransmission.

Figure 3.6 shows two distinct packet exchanges. The first is a successful DATA-ACK exchange between the source and destination. The second exchange demonstrates the cooperative retransmission. Here, the destination sends a NACK in response to the first DATA transmission, indicating a packet error likely due to fading. Both the source and relay receive the NACK and immediately cooperate in re-transmitting the DATA packet. The destination receives this transmission successfully and sends an ACK in response.

Figure 3.7 uses the same four signals as the previous figures, viewed over a much longer time scale (approximately 325 ms). At this scale, it is possible to visualize channel variations and the resulting node transmissions. The node behaviors in Figure 3.7 are the real-time reactions of the cooperative MAC/PHY to the random

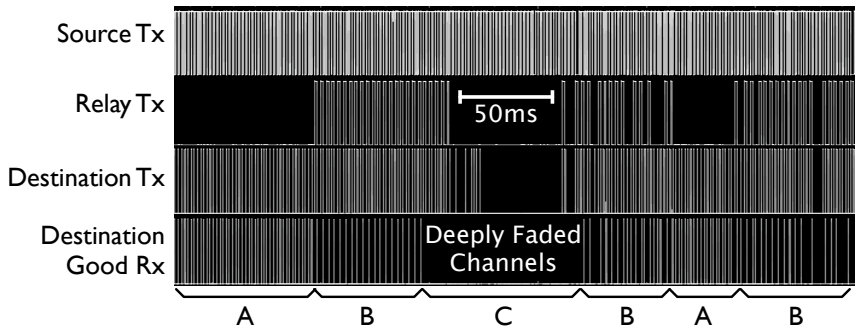


Figure 3.7: Real-time Tx/Rx events showing on-demand cooperation initiated by packet losses due to channel fades. The labels highlight various channel conditions, where the source-destination link does not require help from the relay (A), where the S-D channel degrades sufficiently that the relay actively cooperates (B), and where the S-D link degrades to the point that it cannot sustain any communication (C).

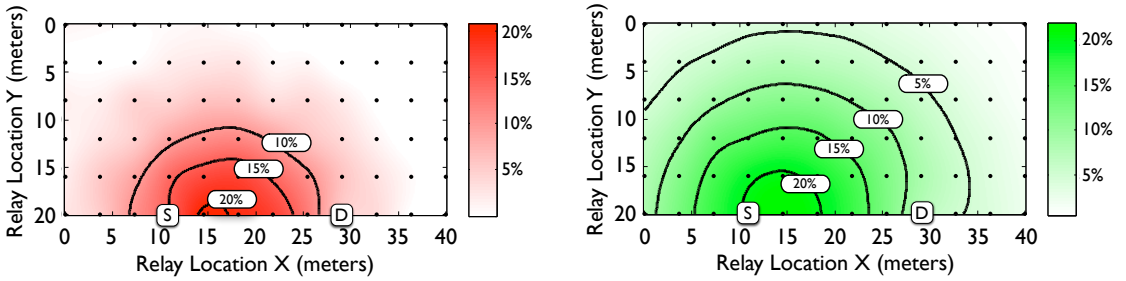
channel coefficients imposed by the emulator.

3.1.5 Experimental Evaluation

Figure 3.8 presents experimental results gathered at 72 locations for the relay with fixed source and destination locations. Each dot in the contour plots represents a relay location, with the contour lines tracing constant paths through values interpolated between relay locations. Cumulatively, the data presented in Figure 3.8 represents 72 minutes of experimental time, or the transmission of over 3.5 million packets.

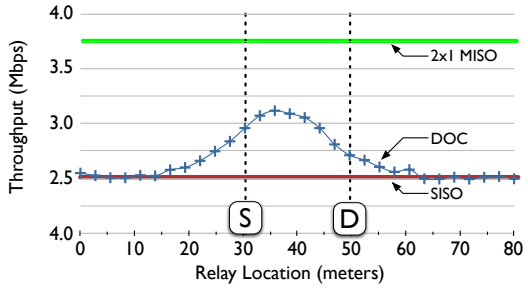
Figure 3.8(a) shows throughput improvement of a relay-aided cooperative link over a non-coopeartive SISO link, while 3.8(b) shows the probability of the relay participating in a given packet exchange at each location. It is clear from these plots that the relay provides the most benefit when located near the source node. The peak performance improvement is significant, exceeding 20%, even though the relay was engaged in fewer than 25% of packet exchanges.

Figure 3.8(c) presents results of throughput for 30 relay locations along the line connecting the source and destination. The throughput plot also presents measurements of two non-cooperative schemes. The SISO line corresponds to a source-



(a) Throughput improvement over SISO.

(b) Probability of the relay cooperating.



(c) Throughput improvement relative to MISO

Figure 3.8: Experimental results for DOC implementation.

destination link running the DOC protocol in the absence of a relay. The 2×1 MISO line shows the throughput of a non-cooperative link where the source uses the same MAC protocol but the PHY operates in Alamouti MISO mode, transmitting simultaneously from two antennas at the source. The multiplexing gain of a 2×1 MISO system is the same as a SISO system, which implies that the asymptotic growth of capacity for MISO and SISO have the same slope [17]. However, at the finite SNRs of interest, 2×1 MISO reduces packet losses due to an added diversity branch [45], which leads to fewer retransmissions and hence increased end-to-end throughput.

Thus, the 2×1 MISO line represents an upper bound to cooperative performance, as a true MISO link realizes full diversity with every transmission, whereas no cooperative scheme can. Most importantly, our cooperative implementation strictly outperforms the SISO link and achieves a significant fraction of the performance gain possible with true MISO.

3.2 Power-controlled Distributed On-demand Cooperation (PDOC)

In Section 3.1, we presented DOC as a mechanism for realizing signal-scale cooperative gains with a random access MAC. In this section, we present and evaluate Power-controlled Distributed On-demand Cooperation (PDOC). This cooperative MAC protocol shows substantial energy savings over DOC while maintaining the same throughput performance. These energy savings stem from two key components: (i) PDOC lets a cooperative relay disable a transmission if that transmission is not going to be successful regardless and (ii) PDOC lets cooperative relays transmit at less-than-maximum transmission power as long as the likelihood of success is not significantly perturbed. PDOC was originally presented in [12].

3.2.1 Related Work

Power-controlled relaying has seen a number of studies with an information theoretic focus [46, 47]. These studies show considerable performance gains for devices that employ power-control, but they rely on substantial amounts of channel state information to be available at all participating nodes. In our work, we focus on the protocol implications of delivering this channel state information to the relevant devices in the network.

A similar research area is the study of power control for Mobile Ad Hoc Networks (MANETs) [48–50]. In all of these works, power control is embedded into the MAC by hijacking the RTS/CTS mechanism established by the IEEE 802.11 DCF. Since power control requires feedback for the transmitter to know how much power to use, RTS/CTS handshakes are performed at maximum power and a lower power is selected for the DATA/ACK exchange. Perhaps most related to our work, the strategy

proposed in [51] applies this RTS/CTS-based approach to cooperative relaying. In all commercial 802.11 devices, however, RTS/CTS behavior is disabled by default. This is due to the fact that the overhead required by using RTS/CTS handshakes in front of every data transmission is too large for normal use[1]. Since RTS/CTS are disabled, any form of power control that relies on them would also be disabled by default. By contrast, our work requires no explicit handshake before attempting to transmit data.

3.2.2 System Model and Analysis

In this section, we provide an information theoretic basis for the study of energy efficient cooperative communications. Specifically, we highlight regimes where traditional cooperative communication protocols create energy waste. We then target these regimes in our protocol design in Section 3.2.3.

We employ an information theoretic outage model to describe link performance in the presence of fading. Formally, let

$$P_{out} = Pr\{\log\left(1 + \mathsf{SNR} \cdot |h|^2\right) < R\} \tag{3.1}$$

represent the probability that a transmission of rate R fails to be decoded, where h represents a multiplicative factor corresponding to an instantaneous channel fade. We use this expression to describe packet loss rate in the absence of any interference. In [11] and Section 3.1, the DOC protocol innovated over the state-of-the-art by ensuring that cooperation occurs only when direct transmission fails. Let the labels S, R, D represent source, relay, and destination respectively. DOC attempts to cooperate

[1]RTS/CTS can still be enabled by users whose networks need it (e.g. extreme hidden terminal problems).

only when

$$\log\left(1+\mathsf{SNR}_{\mathsf{SD}}\cdot|h_{\mathsf{SD}}|^2\right)<R \tag{3.2}$$

$$\log\left(1+\frac{T_{\mathsf{S}}L_{\mathsf{SD}}}{N_0}|h_{\mathsf{SD}}|^2\right)<R, \tag{3.3}$$

where T_{S} represents the transmission power of the source in watts, L_{SD} represents the pathloss between source and destination as a unitless linear scale factor, and N_0 represents the power of a thermal noise floor at the receiver. Assuming a Rayleigh fading channel, let $H_{\mathsf{SD}}=|h_{\mathsf{SD}}|^2$ be drawn as an exponential random variable with unit parameter. However, this density describes all possible channel conditions between source and destination. Since cooperation occurs only on a subset of these conditions, we can manipulate Equation 3.3 in order to redefine the density on H_{SD} as an exponential that is clipped to finite support. Formally, the probability density function for H_{SD} given relay transmission under the DOC protocol is

$$f_{H_{\mathsf{SD}}}(x)=\begin{cases}\frac{\exp(-x)}{1-\exp\left(\frac{-N_0(2^{R_{\mathsf{SD}}}-1)}{T_{\mathsf{S}}L_{\mathsf{SD}}}\right)} & x\in[0,\frac{N_0(2^{R_{\mathsf{SD}}}-1)}{T_{\mathsf{S}}L_{\mathsf{SD}}}]\\ 0 & \text{otherwise.}\end{cases}$$

Borrowing from [2], the rate of a link using decode-and-forward cooperative signaling is

$$R=\log\left(1+\mathsf{SNR}_{\mathsf{SD}}|h_{\mathsf{SD}}|^2+\mathsf{SNR}_{\mathsf{RD}}|h_{\mathsf{RD}}|^2\right) \tag{3.4}$$

$$R=\log\left(1+\frac{T_{\mathsf{S}}L_{\mathsf{SD}}}{N_0}|h_{\mathsf{SD}}|^2+\frac{T_{\mathsf{R}}L_{\mathsf{RD}}}{N_0}|h_{\mathsf{RD}}|^2\right), \tag{3.5}$$

where all variables are defined similarly to before[1]. Solving for the relay transmit

[1]It is worth noting that the $|h_{\mathsf{RD}}|^2$ channel power is modeled as a standard exponential random variable, whereas $|h_{\mathsf{SD}}|^2$ follows the density of the clipped exponential defined earlier.

power T_{R} in Equation 3.5,

$$T_{\mathsf{R}} = \frac{N_0 \left(2^R - \frac{T_{\mathsf{S}} L_{\mathsf{SD}}}{N_0} |h_{\mathsf{SD}}|^2 - 1\right)}{L_{\mathsf{RD}} |h_{\mathsf{RD}}|^2}. \tag{3.6}$$

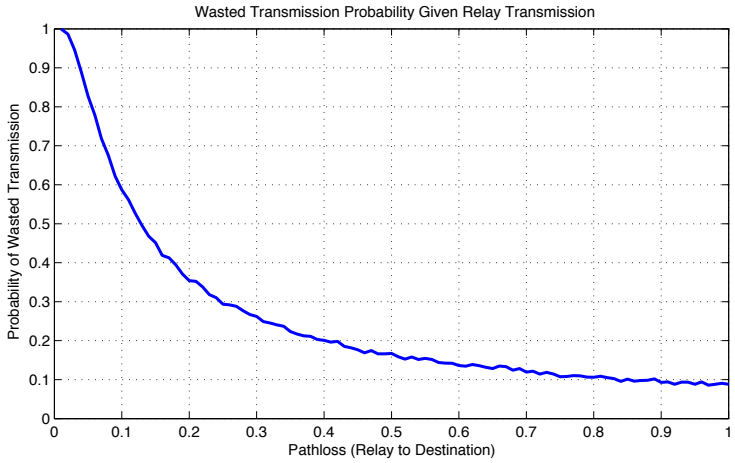
Depending on the instantaneous channel realizations between source/destination and relay/destination, T_{R} can take on a range of values. When very small, this expression tells us that the relay need not transmit at large power in order to ensure that it assists the source on that particular packet. When very large, the converse is true: the relay would require an extraordinarily large amount of transmission power in order to be beneficial.

Suppose T_{S} represents the maximum possible transmission power that each radio can produce. For all cases of $T_{\mathsf{R}} > T_{\mathsf{S}}$, it is impossible for cooperation to assist without violating this constraint. As such, a relay transmitting at maximum power in these regimes produces *wasted* transmissions that only serve to draw power from the relay, meanwhile providing no cooperative benefit. Calculating $Pr\{T_{\mathsf{R}} > T_{\mathsf{S}}\}$ in closed-form is highly involved due to the simultaneous interplay of a clipped exponential random variable along with a standard exponential random variable, so we turn to simulation of these expressions to evaluate this probability.

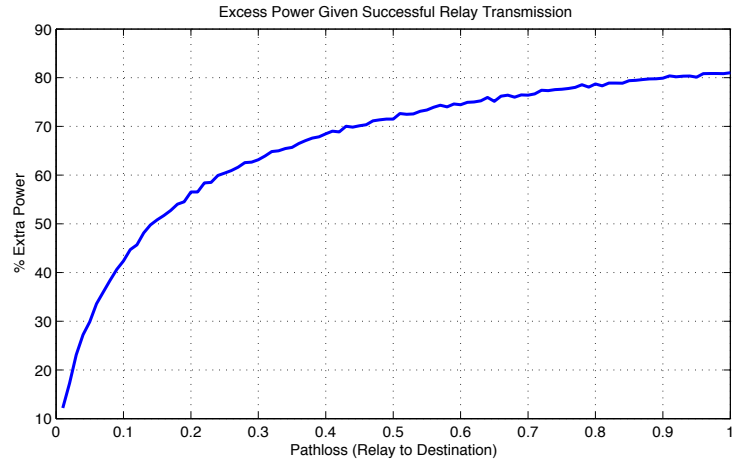
Figure 3.9(a) shows the likelihood of wasted transmissions as a function of L_{RD}. When there is a significant distance between relay and destination (i.e. small L_{RD}) there is a large chance that the power required by the relay exceeds the maximum power constraint. Thus, the DOC relay transmits in these regimes even though it simply cannot help.

In Figure 3.9(b), we plot the following metric:

$$\frac{E[T_{\mathsf{S}} - T_{\mathsf{R}}]}{T_{\mathsf{S}}} \; \forall \; T_{\mathsf{R}} \leq T_{\mathsf{S}}. \tag{3.7}$$



(a)



(b)

Figure 3.9: There are two regimes of waste for the relay: far from the destination and close to the destination. An energy efficient protocol should be able to find maximal savings in these regimes.

This metric captures all the cases where relay transmission would help and then evaluates how much excess power a full power transmission at the relay draws as compared to the minimum that would be needed. Figure 3.9(b) shows that as the relay gets close to the destination (i.e. large L_{RD}), the DOC relay spends upwards of 80% more power than would be strictly required in order to have reliable communication.

From this analysis we find two guiding protocol design goals:

- When a relay cannot help, it should avoid wasted transmissions and simply not

transmit.

- When a relay can help, it should transmit with only *just enough* power to ensure the reliable delivery of the packet.

We use these findings to motivate the structure of our proposed protocol in Section 3.2.3.

3.2.3 Protocol Description

In this section, we describe the Power-controlled DOC protocol (PDOC). The protocol exhibits two key features that are in line with the observations made in Section 3.2.2:

- PDOC only transmits when doing so will likely result in a successfully delivered packet.
- In addition, PDOC will transmit at a power level that is only just sufficient for successful decoding.

PDOC is able to make these determinations by exploiting feedback from the destination node in the event of packet losses.

Figure 3.10 illustrates a timeline of events that occur with the PDOC protocol. The figure shows a failed direct packet exchange followed by a successful cooperative packet exchange. The numbered circles correspond to particular points in time where PDOC behaves differently than the original DOC protocol. These events correspond to:

1. The destination estimates the channel $T_{\mathsf{S}} L_{\mathsf{SD}} |h_{\mathsf{SD}}|^2$ when receiving the packet. This estimation occurs anyway in any coherent communication system with channel-state-information-at-the-receiver (CSIR). Rather than throw away the

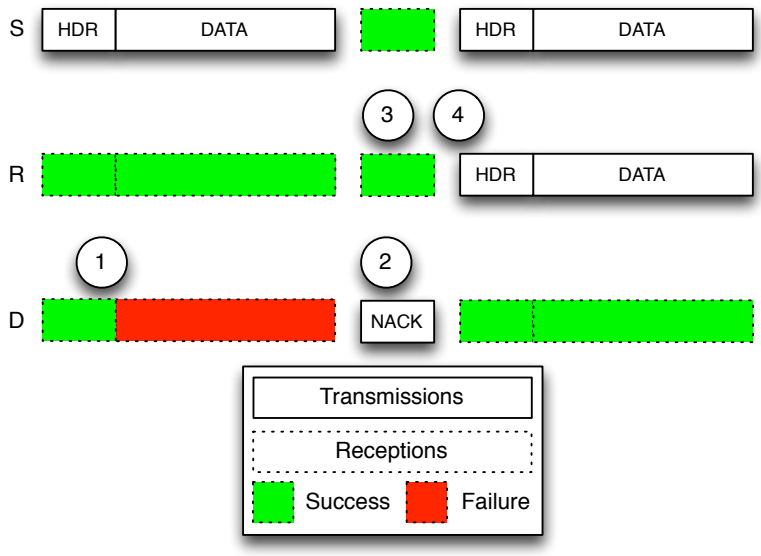


Figure 3.10: PDOC embeds source-to-destination feedback inside NACK transmissions.

estimate after using it to attempt to detect and decode the packet, PDOC receivers save the value into memory.

2. In DOC, a failure event in decoding the payload of the source's transmission causes the destination to generate a negative acknowledgment (NACK) and broadcast it to both the source and relay. PDOC enhances this behavior by including the $T_{\mathsf{S}} L_{\mathsf{SD}} |h_{\mathsf{SD}}|^2$ estimate inside the NACK packet. While this extra information increases the overhead in the protocol over DOC, it is very likely this additional overhead can be negligible. For example, in our WARP implementation of PDOC in Section 3.2.5, received signal strength (RSSI) is a 10-bit value. The only overhead PDOC incurs over DOC is the inclusion of this 10-bit value in the NACK packet. If this is too much, a further quantized version of the value could be included instead at the cost of accuracy in the estimate of $T_{\mathsf{S}} L_{\mathsf{SD}} |h_{\mathsf{SD}}|^2$.

3. Upon receiving the NACK, the relay directly estimates the channel $T_{\mathsf{S}} L_{\mathsf{RD}} |h_{\mathsf{RD}}|^2$. Additionally, it reads $T_{\mathsf{S}} L_{\mathsf{SD}} |h_{\mathsf{SD}}|^2$ out of the NACK itself, giving it access to both the SD and RD channels[1].

4. Finally, the relay decides whether or not to transmit and at what power level. This determination is made as a function of the SD and RD channel values.

The relay determines whether or not its transmission would be helpful by calculating

$$X_{\mathsf{DEC}} = \left[\frac{T_{\mathsf{S}} L_{\mathsf{SD}} |h_{\mathsf{SD}}|^2 + T_{\mathsf{S}} L_{\mathsf{RD}} |h_{\mathsf{RD}}|^2}{N_0} \geq Th \right], \tag{3.8}$$

where $[\cdot]$ is the Iverson bracket and Th is an *a priori* known threshold calibrated to the performance of the physical layer. $X_{\mathsf{DEC}} = 1$ if the relay's transmission will allow the destination to decode the source's message and $X_{\mathsf{DEC}} = 0$ otherwise.

Given that the relay's transmission can be helpful, the PDOC protocol can determine a scale factor on its transmission power such that $X_{\mathsf{DEC}} = 1$ is *still* ensured.

$$T_{\mathsf{excess}} = \min \left(1, \frac{ThN_0 - T_{\mathsf{S}} L_{\mathsf{SD}} |h_{\mathsf{SD}}|^2}{T_{\mathsf{S}} L_{\mathsf{RD}} |h_{\mathsf{RD}}|^2} \right). \tag{3.9}$$

Note that $T_{\mathsf{excess}} \in [0, 1]$ represents how much less T_{R} can be than the maximum transmission power T_{S} while still allowing decoding. Thus, the transmission power the relay chooses is simply $T_{\mathsf{R}} = T_{\mathsf{S}} \cdot T_{\mathsf{excess}}$.

3.2.4 Simulation Evaluation

We first evaluate the PDOC protocol using the ns-2 simulator (version 2.34). Specifically, we base our protocol implementation on the recent 802.11Ext extension [52].

[1] This assumes that the destination would transmit the NACK at full power. Since the source transmits the data at full power as well, $T_{\mathsf{S}} = T_{\mathsf{D}}$.

We have added a number of features to the simulator in order to properly evaluate the PDOC mechanisms.

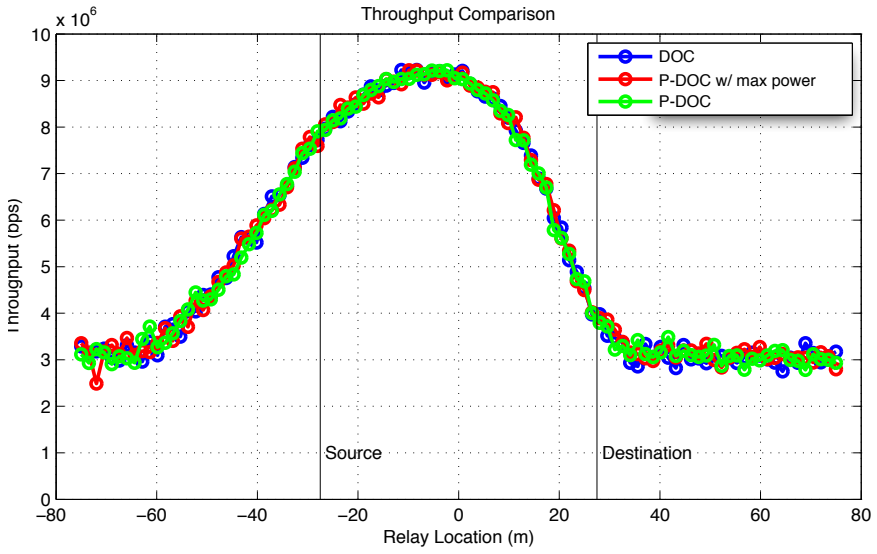
- **Cooperative PHY:** The physical layer is able to distinguish between simultaneous cooperative transmissions and simultaneous interfering transmissions. The former are subject to an increase in received signal power equal to the sum of each individual received signal power. This mimics transmissions employing a space-time block code (STBC) such as Alamouti.

- **Correlated Rayleigh channel model:** By default, the Nakagami fading model present in the ns-2 mobility library uses independent and identically distributed (i.i.d.) block fading where every use of the channel is completely independent of every other use. Since PDOC requires coherence across packets (e.g. to prevent the estimate of h_{SD} from being stale at the relay), we have built a far more realistic channel model where channel powers are correlated over time. Packet transmissions that occur in rapid succession are highly likely to see the same channel.

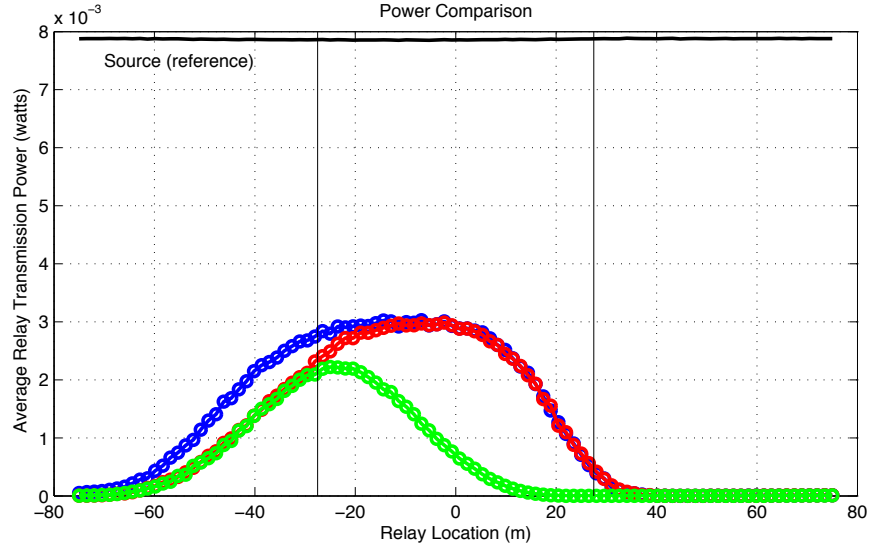
- **Per-packet power control:** By default, all nodes in the network use a single parameter as their transmission power. We extend this model to allow per-packet control over transmission power to allow relays to choose the power level of their transmissions.

After building these extensions to the simulator itself, we implemented the PDOC protocol described in Section 3.2.3. To evaluate the protocol, we simulated a linear topology with a fixed source and destination location and swept the relay along the line connecting the two.

Figure 3.11(a) shows the throughput performance of DOC, PDOC with maximum transmission power, and the full PDOC. The proposed protocol is able to achieve



(a)



(b)

Figure 3.11: The PDOC protocol successfully saves energy in regimes both close to and far away from the destination.

precisely the same performance as the original DOC protocol, which means that its energy savings do not come at the cost of any lost performance.

Figure 3.11(b) shows the average measure of power that the relay spends transmitting for each of the protocols. Note that PDOC shows savings over DOC when the relay is to the left of the source. This is true even when the relay is forced to al-

ways transmit at maximum power. This confirms our predictions from Figure 3.9(a). When far away from the destination, the relay is less likely to provide any benefit at all. PDOC chooses not to transmit in these regimes and takes no degradation in performance since those transmissions would not be successful anyway. In regimes where the relay is closer to the destination, PDOC performs many orders-of-magnitude better than DOC. This confirms our predictions from Figure 3.9(b). As the relay gets close to the destination, it can afford to lower its transmission power since little power is needed to traverse such a powerful channel.

3.2.5 FPGA Implementation

In this section we choose to focus solely on one particular challenge associated with the implementation of PDOC. In Equations 3.8 and 3.9, relay transmission power is directly computed from the formal definition of an information theoretic outage event in Equation 3.3. This computation is straightforward due to an infinite-length block code that can *guarantee* the success or failure of a transmission for any given relay transmission power. In practice, any finite length code cannot make this guarantee. Instead of directly calculating the required relay transmission power like in Equations 3.8 and 3.9, we experimentally determine the mapping between relay power and successful decoding probability for our physical layer implementation.

In this experiment, we use the same Azimuth ACE 400WB emulator [41] as from Section 3.1 to control the source-to-destination (SD) and relay-to-destination (RD) channel qualities. We then artificially trigger node transmissions and measure three outputs from the network: (i) the SD link power (RSSI) of a source-only transmission at the destination, (ii) the RD RSSI of a destination-only NACK transmission at the relay, and (iii) the packet error rate (PER) of a cooperative $\mathsf{S}+\mathsf{R}$ transmission as a function of the first two measurements.

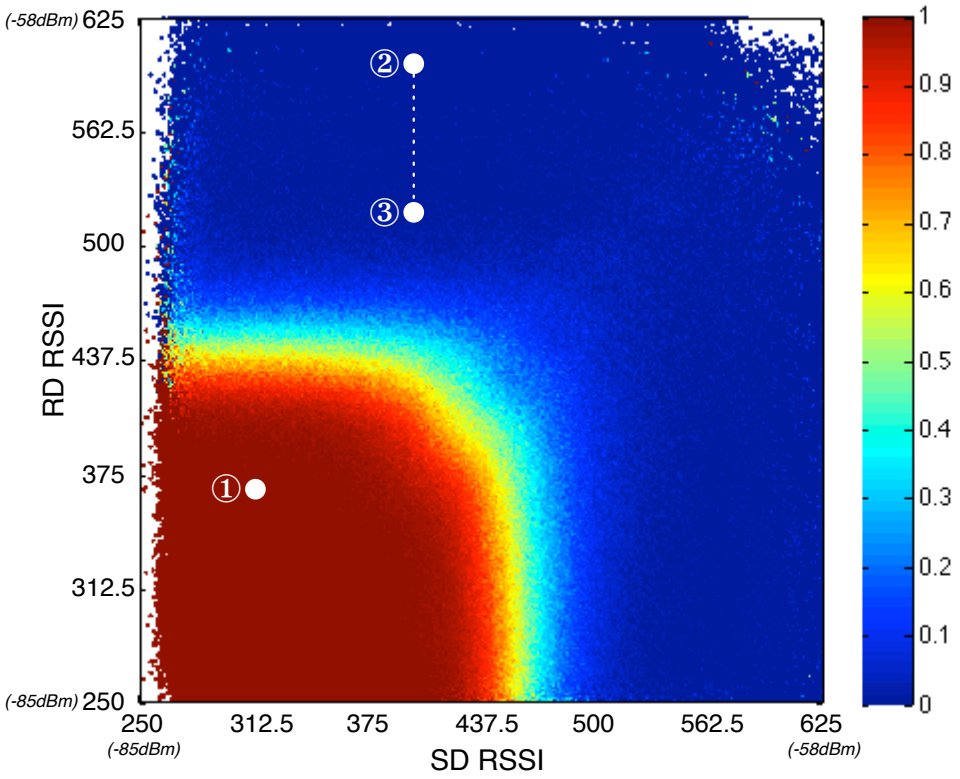


Figure 3.12: Packet error rate of cooperative transmission.

Figure 3.12 shows the packet error rate of our cooperative physical layer as a function of the SD and RD channel qualities[1]. The regions in red correspond to cooperative transmissions that are nearly guaranteed to ultimately fail despite the presence of the relay. Regions in blue, however, show scenarios where cooperative transmissions are very likely to succeed[2].

A useful interpretation of this figure is that random fading in the network amounts to a random dart throw in this RSSI space. As an example, if this dart happens to land at the point labelled ①, the relay knows that *even at maximum transmission*

[1]For this experiment, we use a variant of our physical layer that employes no channel coding. The same characterization can be applied to the different code rates thereby allowing PDOC to coexist with automatic rate adaptation schemes.

[2]This experiment uses our emulator with a single channel tap and is therefore frequency flat. In frequency selective channels, PDOC may be extended by exchanging RSSI for a per-subcarrier power measurement.

power, the cooperative packet is still very likely to fail. Therefore, the relay simply avoids transmission. However, if the random fading dart happens to land at the point ②, then the relay knows it can scale back its transmission power such that the final operating point will be point ③ without much harm to the error rate. In our implementation, we have enforced that any relay transmission where the probability of error is larger than 90% is abandoned. Furthermore, a relay can scale down its transmission power such that no more than 10% probability of error is incurred. Of course, these values can be modified by future implementations and were only selected as a baseline for evaluation.

3.2.6 Experimental Evaluation

To evaluate and characterize the performance of the full PDOC implementation we use the channel emulator to effectively mimic a 2D topology with fixed source and destination locations by controlling the output attenuation on each path. The emulator itself controls the small-scale Rayleigh fading applied to each link. We then sweep the relay along points in the 2D space and linearly interpolate the results.

Figure 3.13(a) shows the throughput performance of our real-time DOC implementation. When the relay is in the region between the source and destination, the throughput gains are quite significant (from just over 3Mbps to nearly 7Mbps)[1]. Figure 3.13(b) shows the throughput performance from the new PDOC implementation. The proposed protocol is able to achieve the same performance as the original DOC protocol, which means that energy savings do not come at the cost of any lost performance.

Figure 3.13(c) shows the average amount of power that the DOC relay spends

[1]This performance increase actually exceeds the performance increase reported in Section 3.1. In between the original DOC publication and the evaluation of PDOC, we had completed the shift away from an amplify-and-forward physical layer to a decode-and-forward one.

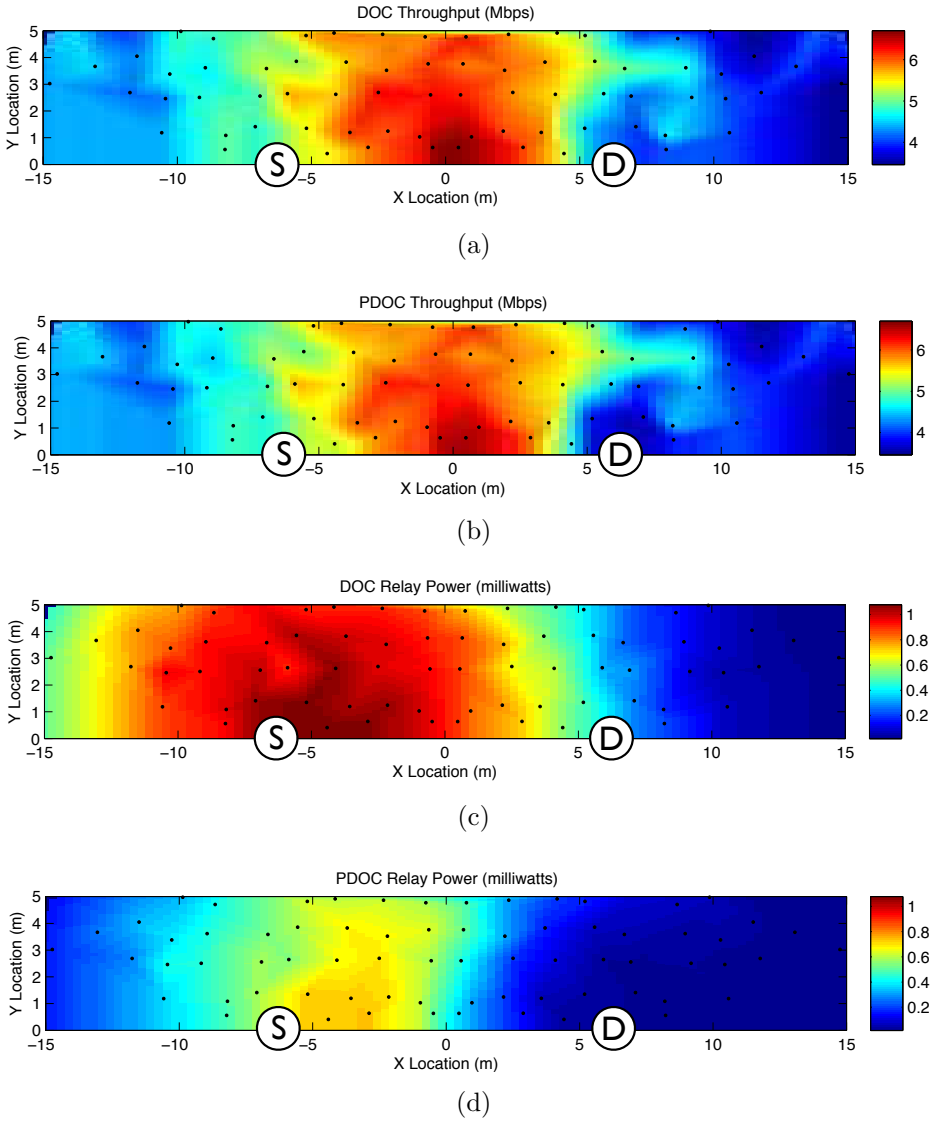


Figure 3.13: The PDOC protocol successfully saves energy in regimes both close to and far away from the destination.

transmitting while Figure 3.13(d) shows the same information for the PDOC implementation using the same color scale. There are significant power savings throughout the topology. Furthermore, two key regimes for savings are apparent:

- **Far-from-Destination Relay:** When the relay is far from the destination (e.g. the relay is to the left of the source in Figure 3.13), there are many occurrences of channel fades that lead to a scenario in which cooperative transmission still

fails. PDOC recognizes these conditions and appropriately disables relay action to save significant amounts of energy without adversely affecting performance.

- **Close-to-Destination Relay:** When the relay is close to the destination, there are many occurrences of channel fades in which the relay need only transmit at a small fraction of maximum transmission power to still be successful.

These two observations confirm predictions from Figures 3.9(a) and 3.9(b).

3.3 Discussion

In this chapter, we have presented two novel MAC protocols that can substantially reduce unnecessary relay transmissions as well as the energy usage of cooperative relays. Furthermore, these protocols use NACKs to trigger, synchronize, and coordinate cooperation between devices thereby answering the first challenge laid out in Chapter 1. The next two challenges are considered in Chapters 4 and 5, respectively. First, why should a device agree to be a cooperative relay in the first place, given that acting as such will use energy even under the best conditions? And second, even when a device is dedicated to acting as a relay, how can it avoid creating inter-device interference as a result?

Chapter 4

Energy Efficiency

In Chapter 3, we presented DOC and PDOC as MAC layers that can leverage signal-scale cooperative resources. In this chapter, we consider the broader implications of using such cooperative protocols on the energy efficiency of devices in a wireless network. Specifically, in Chapter 3, all MAC evaluations considered the case of relay devices dedicated to being cooperative relays in the network; they never had their own traffic to send or receive. In this chapter, we generalize our study to consider an arbitrary collection of devices that may have their own traffic to send and may cooperate with one another.

A key barrier to the practical deployment of cooperation in mobile devices remains unsolved: when two mobile devices cooperate, the performance gain from cooperation may be uneven. In fact, if one device acts as a cooperative relay for another far more than the reverse case, it may well suffer in both throughput and energy efficiency. Thus, a key question for every energy-constrained mobile device is "why should I help?".

We answer the above question by designing a completely device-localized protocol called Distributed Energy-Conserving Cooperation (DECC). DECC has three components. First, DECC measures the extent of benefit that cooperation provides by

measuring an energy efficiency reference when periodically disabling its cooperative mode. This allows the device to measure the amount of benefit it derives by cooperating with other devices. This measurement is performed completely asynchronously, without any assistance or packet exchanges with the AP or other devices.

Second, based on the measured benefit, the devices scale their cooperative "effort," a dimensionless quantity to control their level of cooperation. This parameter is adjusted by a proportional-integral control loop that tracks the ongoing energy efficiency calculations.

Third, the effort parameter is used to modify the MAC layer behavior by allowing the MAC to make a decision on a per-packet basis if it will cooperatively transmit the last overheard packet or not. The determination to cooperate on a packet also uses information about the instantaneous channel quality estimates of the relevant links. These estimates are derived from the last received packet via each link so no additional overhead in channel estimation is incurred.

A key feature of DECC is that it is "association-free." That is, devices do not care whose packet they are helping but simply adapt their cooperative behavior to manage their own total energy efficiency loss or gain. This is an important aspect as it allows the device behavior to automatically adapt to topology changes due to device mobility and traffic changes.

We use our PDOC implementation from Chapter 3 to investigate and determine two crucial findings. First, we show that there exist network situations where signal-scale cooperation is beneficial for *all* devices in the network. In these situations, even selfish devices will cooperate. Second, we show that in many cases, a little sacrifice by a helping device may produce disproportionally large gain for the device being helped. Even a little altruism can lead to significant gain for others. Inspired by these observations, DECC allows each device to constrain its level of altruism to a

user-selectable bound.

DECC has been implemented and evaluated on WARP. We have additionally developed a custom simulator to show that DECC also scales to arbitrary network sizes and topologies. With DECC, devices can bound their maximum acceptable loss in energy efficiency to, for example, only 5%. But more importantly, DECC allows the devices to derive large energy efficiency benefits when they are available with completely localized decision making.

We envision many uses of DECC in actual devices, all of which may have the following philosophy. Wireless users are already growing accustomed to the "spectrum crunch." So they could be inspired to enable a bounded altruism mode (e.g. they contribute no more than 5% energy efficiency) in the hope that they will be helped in the future by other devices as they move in the network and encounter poor connectivity locations. Since devices control their individual level of altruism, this amount can be changed by each user based on their specific use case scenario. If no one cooperates, DECC simply reduces to the base MAC and PHY (e.g. Wi-Fi).

DECC allows devices to become *self-aware*; they explicitly monitor the impact of cooperating with others on their own performance. In Chapter 5, we present techniques to allow devices to become *network-aware* so devices consider the impact of cooperating with others *on* others in the network.

4.1 Related Work

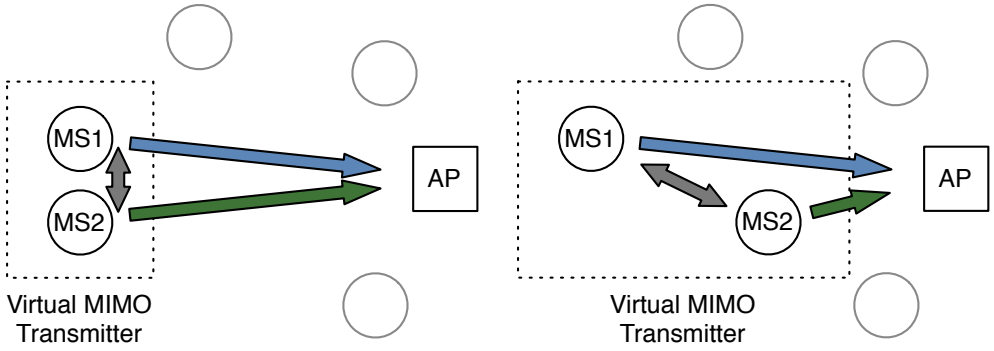
Methods to induce cooperation between nodes in a network have been extensively studied in multiple contexts. For example, in the context of ad hoc networks, there has been extensive work to promote cooperation so that all nodes participate in a fair manner to increase either the network capacity [53] or improve energy efficiency [54]. The key mechanisms in this large body of work are inspired by game theoretic princi-

ples, where each node is considered rational and optimizes its utility function [55, and references therein]. The implicit assumption in game-theoretic development of cooperative (or competitive) networks is each node is fully aware of whom it is playing this game against – which is akin to each node knowing about all the nodes in the network, their utility function and what policy they are using to maximize the utility. This information is critical for proving properties of the game (e.g. Pareto optimality, Nash equilibria). However, that level of information is not available in Wi-Fi networks, since the potential overhead of collecting, managing, and sending this information to all nodes would be very high and hence not scalable. In contrast, we focus on the design of network protocols where each node is completely agnostic to the network topology and has no access to other nodes' utilities or strategies.

4.2 Energy Efficiency using Signal-Scale Cooperation

By design, signal-scale cooperation is intended for small devices (e.g. smartphones or tablets) that are unable to directly achieve spatial diversity or multiplexing through the use of antenna arrays. These small devices share another fundamental trait: they are typically also constrained by limited battery life. As such, the *energy efficiency* of a device is extremely important. When this is taken into account, signal-scale cooperation exposes a clear tradeoff – in order to allow devices in the network to realize the significant rate and reliability performance gains that cooperation can provide, other devices must spend additional wireless transceiver energy to act as their cooperative transmitters. To date, it is unknown whether such additional energy costs are prohibitively high when offset against cooperative gains.

This tradeoff between increased energy consumption and better performance is



(a) Symmetric Topology (b) Asymmetric Topology

Figure 4.1: Asymmetries in topology yield asymmetries in the performance/energy tradeoff.

exacerbated by the fact that not everyone in a wireless network has the same amount to gain from cooperation nor do they have the same potential energy costs. Figure 4.1 illustrates a common uplink scenario that highlights this problem. In Figure 4.1(a), mobile stations MS1 and MS2 are equidistant from the access point AP and see similar distributions of fluctuations in the wireless channel. Because of this symmetry, the gains and costs of cooperation will be comparable for both devices. In Figure 4.1(b), MS2 is much closer to the AP than MS1 and therefore has a much higher SNR link to the AP on average. This yields two important effects:

- MS2 has less to gain from cooperation than MS1 since it can rely more heavily on its direct link to the AP.
- MS2 has more energy to lose from cooperation since it can provide more substantial and more frequent cooperative assistance to MS1.

As this simple example demonstrates, the "winners" and "losers" from cooperation are topologically dependent. In Section 4.3, we perform an in-depth WARP characterization of this topological dependence to demonstrate how severe the performance/energy tradeoff can be.

4.2.1 Key Metric

Throughout this chapter, we use "energy efficiency" as our fundamental metric. Specifically, we define energy efficiency to be

$$\gamma = \frac{\text{\# of bits transmitted + received}}{\text{total energy spent}}, \tag{4.1}$$

where γ has a unit of bits/joule. Practically speaking, energy efficiency captures the amount of data that comes into and goes out of a device for a certain amount of battery capacity spent. Note, the numerator of this metric does *not* include any receptions and transmission made on behalf of others; it counts only the bits that are relevant to the device.

4.3 Experimental Characterization

To quantify the energy efficiency challenge for signal-scale cooperation, we study small, representative topologies in detail using the PDOC WARP implementation from Section 3.2. In this systematic characterization, we make two key findings:

- Cooperation can be mutually beneficial to devices, but the regimes where this occurs are topologically small and therefore unlikely in general.
- In regimes where one device suffers energy efficiency loss for the gain of another device, the loss to one is usually smaller than the gain to the other.

These findings provide the motivation for our protocol innovations in Section 4.4.

For the experiments in this chapter, we have extended the PDOC design to implement the performance of maximal-ratio-combining between the first source trans-

mission and relay-only transmission. Details of this extension are available in Appendix B. Throughout our study, we take an *event-driven* approach to power measurements. Offline, we calculate the energy costs of transmission and reception events and save these values to the board. Then, the real-time FPGA implementation on WARP uses these values to calculate on-the-fly how much power is being drawn. The details of the offline power measurements that enable this study are available in Appendix C.

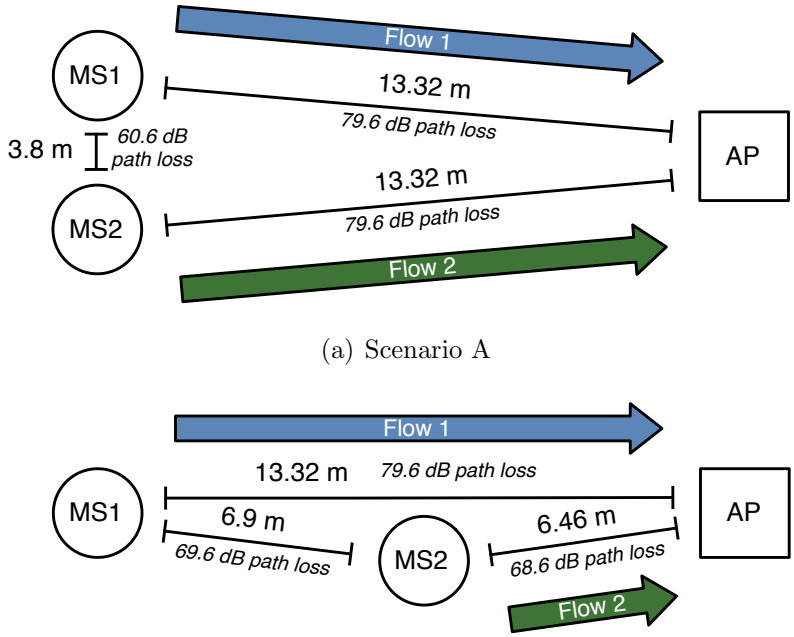
4.3.1 Experimental Results

Using our WARP implementations, we measure the achieved throughput and energy efficiency of the cooperative PDOC protocol alongside a non-cooperative CSMA/CA protocol.

4.3.1.1 Topologies Studied

To study the impact of topological asymmetry on the gains and costs of signal-scale cooperation, we first study the atomic case of two flows in a three-node system, where mobile stations with backlogged queues are uplinking to a common access point. For all of the hardware experiments in this chapter, we use the Azimuth Systems ACE 400WB channel emulator. All links undergo a single-tap Rayleigh fading channel with a velocity of 1.2km/hr to emulate channel coherence times consistent with walking-speed mobility.

We use the channel emulator to construct the two topologies shown in Figure 4.2. Using a path loss model, we convert the distances shown in the figure to average path losses using a path-loss exponent of 3.5. This is consistent with the urban environments in which non-line-of-sight Rayleigh fading occurs. We configure the emulator with these path loss values and let the emulator then superimpose the



(a) Scenario A

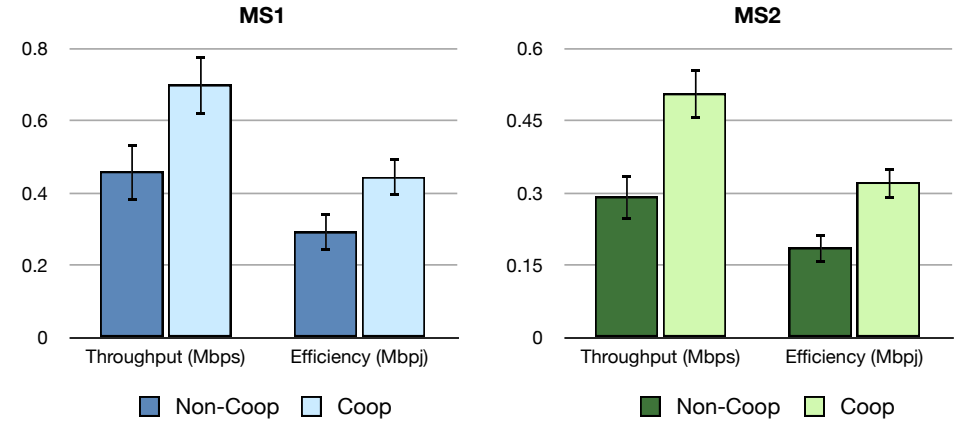
(b) Scenario B

Figure 4.2: Topologies under study.

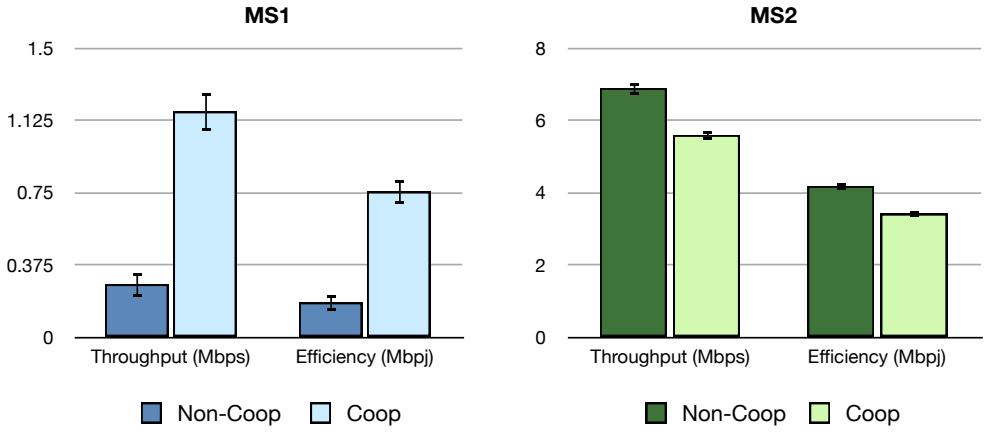
Rayleigh channel fading on top of those values.

In Scenario A, the mobile stations are equidistant from the AP to create a symmetric topology. In Scenario B, MS2 is placed approximately halfway between MS1 and the AP. This introduces an asymmetry in the network where the flow sourced by MS2 undergoes less severe fading events.

All results in this section come from 15 separate 120 second trials where all random number generators (at each node and the channel emulator) are reseeded. Error bars represent the standard deviation across those trials. All data transmissions are 1300 byte packets modulated at 16-QAM in 10MHz of bandwidth. The data presented here represent the cumulative successful reception of over 28 billion bits, or 2.7 million data packets.



(a) Scenario A



(b) Scenario B

Figure 4.3: Experimental results for Scenarios A and B.

4.3.1.2 Scenario A: Symmetric Topology

In the symmetric topology, both flows are in equally harsh fading environments. Figure 4.3(a) shows that cooperation is mutually beneficial to both flows. There are many cases where the instantaneous fades between each MS and the AP create link outages. By cooperating and achieving spatial diversity, these links can often be repaired. This results in an energy efficiency improvement for each flow in excess of 50%.

4.3.1.3 Scenario B: Asymmetric Topology

In the asymmetric topology, Flow 2 is considerably closer to the AP than Flow 1. As such, Flow 2 incurs fewer transmission failures due to adverse channel conditions. Figure 4.3(b) shows a stark result: while Flow 1 achieves a dramatic improvement in energy efficiency performance (+320%), that benefit comes at the cost of Flow 2 (−18%). In the asymmetric topology, MS1 relies on MS2 to act as its relay far more often than the reverse case. MS2 spends considerable resources to help MS1 without reciprocation.

4.3.2 Establishing the Trend

While the study of Scenarios A and B in Sections 4.3.1.2 and 4.3.1.3 establish the severity of improvement or degradation in energy efficiency, they do not establish how likely these two scenarios are. We examine the trends connecting these two regimes with another empirical study considering a linear topology.

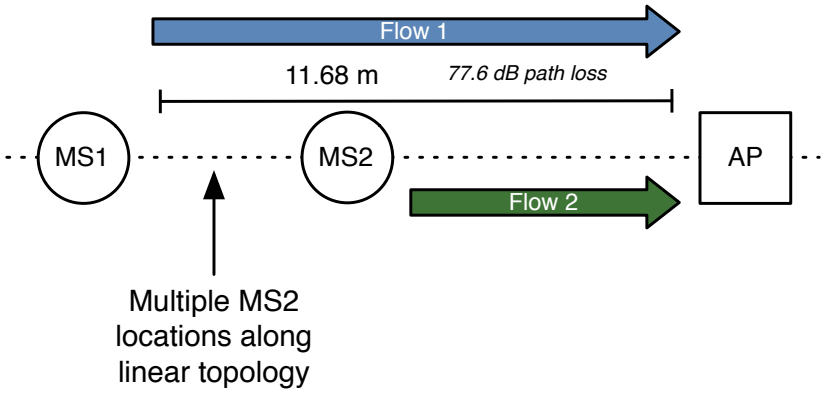


Figure 4.4: Linear topology.

In the topology in Figure 4.4, MS2 is placed at a number of locations along the linear topology that connects MS1 and the AP. In this way, MS2 can be "swept" through the mutually beneficial and one-sided beneficial regimes that we previously established.

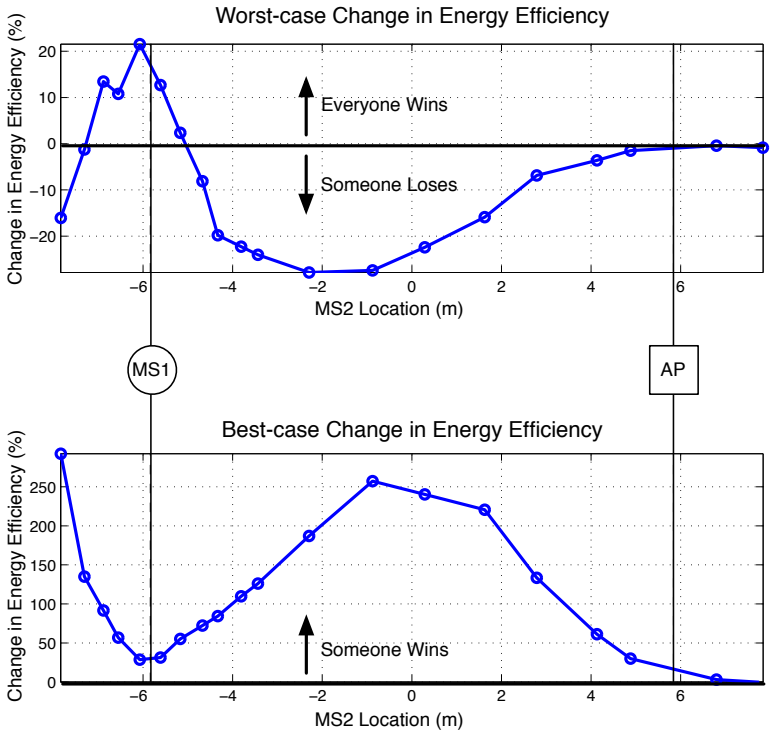


Figure 4.5: Experiment results for linear topology.

In Figure 4.5, we first examine the worst-case change in energy efficiency. At every experiment point, we plot either the change in energy efficiency of MS1 or MS2 (whichever is worse). When the worst-case change is larger than 0%, this describes a regime where *everyone wins*. Both MS1 and MS2 are better off with cooperation than without it. While this situation is clearly possible, it is not likely in a general network – there is only a span of around two meters near MS1 where mutually beneficial cooperation occurs. Thus, our first experimental finding is that cooperation can be mutually beneficial to devices, but the regimes where this occurs are topologically small and therefore unlikely in general. Note that this observation ties back to our observation in Section 3.1, in that it suggests that one relatively simple usage scenario for the implementation of cooperation is where the same user allows two personal devices (such as a laptop and a smartphone) to cooperate with each other.

We next examine the best-case change in energy efficiency. In the regimes where

"someone loses," the other device does dramatically better. The sacrifice from one device yields tremendous improvements in energy efficiency to the other. Accordingly, our second experimental finding is that in regimes where one device suffers energy efficiency loss for the gain of another device, the loss to one is usually smaller than the gain to the other.

These two findings motivate the need to allow for *some* degree of altruism among devices. Not only are mutually beneficial regimes rare, but the gains that can be had by allowing one device to suffer even a small amount can be very large for another. In this chapter, we present the Distributed Energy-Conserving Cooperation (DECC) system that can allow a device to provided a bounded amount of altruism in the cooperative aid of others in a network.

4.4 Protocol Description

In this section, we present the details of DECC. This design provides a completely distributed mechanism for devices to enforce *bounded altruism* in cooperative networks. With DECC, devices realize substantial improvements in energy efficiency or, at worst, they suffer a small and well-bounded amount of degradation.

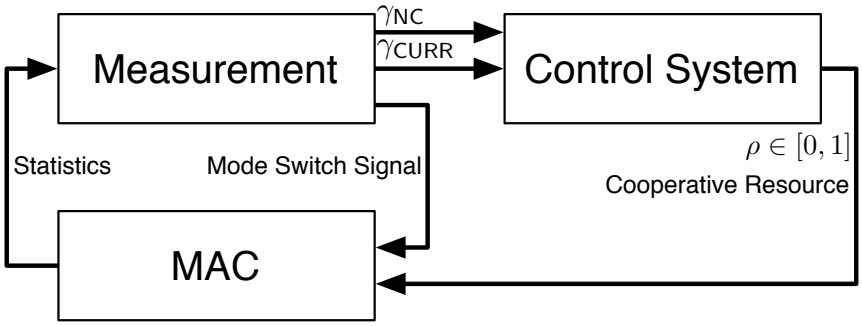


Figure 4.6: DECC block diagram.

Figure 4.6 illustrates the block diagram for DECC. It is important to emphasize that this system is implemented at *each node in the network* and is completely

distributed. There are three main subsystems of DECC: the measurement module, control system module, and the MAC.

4.4.1 Measurement

The measurement module is responsible for determining whether or not cooperation helps or hurts the energy efficiency of the device. It actively switches a device to and from a mode where cooperation is not allowed. During these mode switches, it directly measures the energy efficiency of the device. The measurement module produces two outputs:

γ_{CURR}: the current energy efficiency of the device. This value can be directly measured by periodic polling of the statistics from the MAC. In a unit of time (e.g. once per second), the measurement block interrogates the MAC for the number of bits it has successfully transmitted or received. It also interrogates the kernel to determine the amount of energy spent (in joules) over the same time. The ratio of these two values make up the current energy efficiency of the node.

γ_{NC}: the energy efficiency of the device with its cooperation disabled. If cooperation is currently being employed, this cannot be directly computed.

The method for measuring non-cooperative energy efficiency requires each device to occasionally disable cooperation for a period of time when it senses changes to incoming or outgoing traffic load. During this explicit non-cooperative period, $\gamma_{\mathsf{NC}} = \gamma_{\mathsf{CURR}}$. The device can perform this switch by simply refusing to send any NACKs and refusing to act as a relay for any other device in the network. Additionally, the device will include a single bit flag inside any outgoing transmissions to

notify any destinations that they should not send any NACK packets for that transmission. This will ensure that the device neither acts as a cooperative relay for others nor is aided by others on its transmissions. The frequency and duration of the non-cooperative periods are parameters that are exposed to higher layers. When a device senses changes to traffic patterns or mobility, these parameters can be adjusted at runtime accordingly. Each device asynchronously switches modes with no coordination with others. This allows each device to set their own measurement parameters. In Section 4.5.2.3, we empirically study these parameters in an example network to investigate the length of time that is sufficient for accurate measurements.

4.4.2 Control System

The control system is responsible for scaling the effort the device dedicates to acting as a cooperative relay for others such that it is sufficiently offset by the performance gain from cooperation. The control system uses the outputs of the measurement module as inputs to a feedback loop that controls a normalized value corresponding to cooperative effort.

The output of the control system is a cooperative resource $\rho \in [0, 1]$. $\rho = 0$ represents the case where a device never provides cooperative assistance to others and $\rho = 1$ represents the case where it provides as much assistance as it is able to.

In the control system, we introduce a new parameter α that represents an *altruism factor*.

$$\alpha \in [0 \text{ —— } 1 \text{ ———— } \infty]$$
Selfless Selfish

When $\alpha \to \infty$, the control system will disallow cooperation with others at all times. When $\alpha = 1$, the control system will disallow cooperation if there is *any* loss on energy efficiency. When $\alpha = 0$, the control system is effectively disabled and will

not attempt to enforce a minimum energy efficiency. As such, the device is completely selfless and willing to cooperate with others as much as possible. This factor allows a user to scale how altruistic his or her device is.

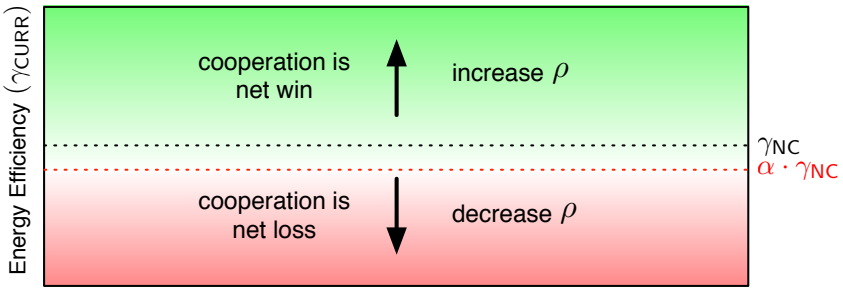


Figure 4.7: Relationship between cooperative resource ρ and energy efficiency γ.

Figure 4.7 illustrates the goal of the control system. When cooperation is a "net win" ($\gamma_{CURR} \geq \alpha \cdot \gamma_{NC}$), cooperative resources are increased. When cooperation is a "net loss" ($\gamma_{CURR} < \alpha \cdot \gamma_{NC}$), cooperative resources are decreased. The idea is that, if a device is performing better with cooperation than without, the only reason that is occurring is because other devices in the network are giving it cooperative resources. If devices start rolling back cooperative resources when they are "winning," this gain can disappear. It is only when a device knows that cooperation is actively harming its energy efficiency that it should lower the amount that it is willing to help others.

In determining a value for α, setting $\alpha = 1$ will not ensure that everyone is helped by cooperation; it will only ensure that no-one is hurt. If every device in a network operates under the rule that it will tolerate *no* negative deviation from non-cooperative performance, then the steady-state operation of that network can easily devolve to no device ever cooperating. As an example, imagine a general network where all cooperation is currently disabled. In other words, $\gamma_{CURR} = \gamma_{NC}$. If the devices in the network employ an altruism parameter $\alpha = 1$, then cooperation stays disabled *even if* some users in the network could benefit at low cost to others.

The α altruism parameter, when set below one even by a slight amount, can serve to bias the network in the direction of providing cooperative resources in case there is mutual benefit to be had. Returning to the previous example, if $\alpha = 1 - \epsilon$ (where ϵ is a very small number), then the cooperative resources of devices will start to ramp up among devices whom cooperation is helping. As devices start to see gains from cooperation they will correspondingly cooperate more, resulting in further gains. For our implementation in Section 4.5, we evaluate a value for α that is *prima facie* tolerable to everyone ($\alpha = .95$).

As a control system, we employ a classic design known as a proportional-integral (PI) controller. These controllers date back to the 1890s [56] and are used heavily to this day.

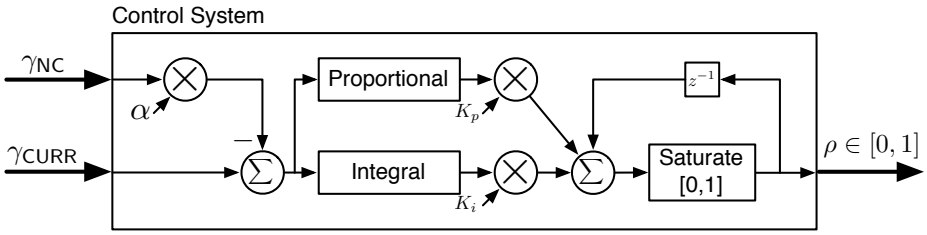


Figure 4.8: Proportional-integral control loop.

Figure 4.8 shows the PI controller in DECC. The controller reacts to two terms:

- **Proportional:** The proportional branch of the controller reacts to a current measurement of error. This error describes how far the system is away from some established baseline at this very moment.

- **Integral:** The integral branch of the controller reacts to the history of errors.

The two filter parameters K_p and K_i can be adjusted to set how quickly the control loop reacts to changes in its inputs as well as the damping response of the filter.

4.4.3 MAC

Given ρ, the MAC should scale its *effort* as acting as a cooperative relay. We use PDOC from Chapter 3 such that, when DECC lowers ρ, the MAC avoids cooperating on the most expensive packets *first*. This recognizes the fact that not all transmissions cost the same amount of energy.

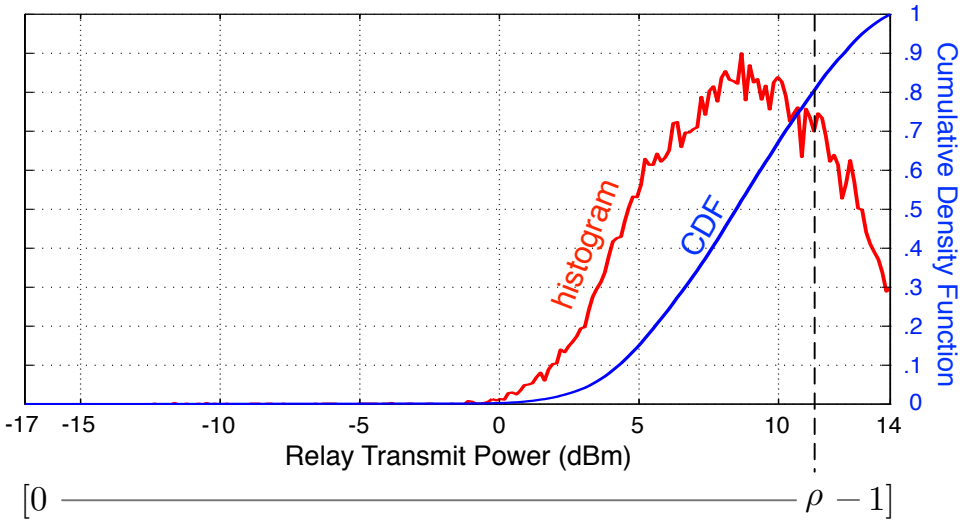


Figure 4.9: Using ρ to cap relay transmission power.

Figure 4.9 shows data from the MS2 device from the experiment in Section 4.3.1.3. This data shows the PDOC protocol in action. Over the range of possible transmission powers (-17 to $+14$ dBm), MS2 selects a transmission power such that the cooperative transmission is likely to succeed. Notice that the majority of transmissions do not occur at maximum transmission power. Because of this fact, by using the ρ input from DECC, the MAC caps its maximum relay transmission power and ensures that the cooperative exchanges that are skipped are the ones that draw the most power from the device. As illustrated in Figure 4.9, a $\rho \approx .79$ would eliminate 20% of relay transmissions. Those eliminated relay transmissions would be the highest power and therefore the most costly in terms of energy efficiency.

In this way, the MAC can react to *instantaneous* changes in channels. Even with

a small ρ, the MAC may still opt to act as a cooperative relay if the necessary power for that action is sufficiently small.

4.5 Realization and Evaluation

In this section, we evaluate DECC by presenting a custom real-time prototype on the WARP testbed. We then revisit the two topologies from Sections 4.3.1.2 and 4.3.1.3 and characterize DECC's performance. Finally, we use an implementation of DECC in a custom simulator to evaluate a larger network scenario.

4.5.1 Implementation Details

We have implemented DECC on WARP entirely in the built-in PowerPC without any changes to the FPGA fabric.

The MAC keeps a cumulative counter of the number of bits successfully transmitted (i.e. ACKed bits) and the number of bits successfully received. The MAC also keeps a cumulative measurement of the amount of energy spent according to the description provided in Appendix B. These values are provided to the DECC software module via a periodic 1 Hz timer. Every second, the DECC software module calculates the energy efficiency of the previous second of traffic and feeds this into the proportional path of the PI control system. Additionally, it updates a cumulative energy efficiency over the entire runtime of the experiment. This cumulative energy efficiency forms the integral path of the PI control system. The ρ cooperative resource is updated at the conclusion of each timer callback.

As described in Section 4.4.3, the MAC layer uses the value of ρ to decide whether or not any given potential relay packet should actually be transmitted. Specifically, ρ is used as a scale factor on the maximum allowed transmission power. With the

WARP radios and OFDM PHY, our maximum possible transmission power is 14 dBm, so the maximum allowed transmission power is simply

$$P_T \leq \rho \cdot 14 \text{ dBm}. \tag{4.2}$$

Any potential relayed packet that exceeds this power threshold is cancelled and nothing is transmitted. We use our WARPnet Measurement Framework to allow each device to send packet traces over Ethernet to a host computer for analysis [57–59].

For simplicity of visualization we have first opted for a "one-shot" non-cooperative mode performed at the beginning of the experiment for the devices to measure γ_{NC}, the energy efficiency of a non-cooperative link. We then demonstrate the full asynchronous mode-switching functionality of DECC's measurement module in Section 4.5.2.4.

Table 4.1: DECC Parameters

K_p	10^{-8}
K_i	10^{-7}
α	$0, .95$
Direct Mode Duration	45 s
Trial Duration	500 s

Table 4.1 shows the DECC parameters used for the experiments in this section. We do not claim optimality in the selection of these parameters. In fact, we have intentionally left the control loop slightly underdamped as it makes the visualization of the input-output relationship of the DECC control system easier to discuss.

Figure 4.10 shows an example timeline of events taken from an oscilloscope capture of a trial. One can see MS1 and MS2 transmitting their own data packets as well as a cooperative transmission from MS2 to assist MS1 for a packet that was NACKed. In order for this cooperative transmission to have occurred, the required

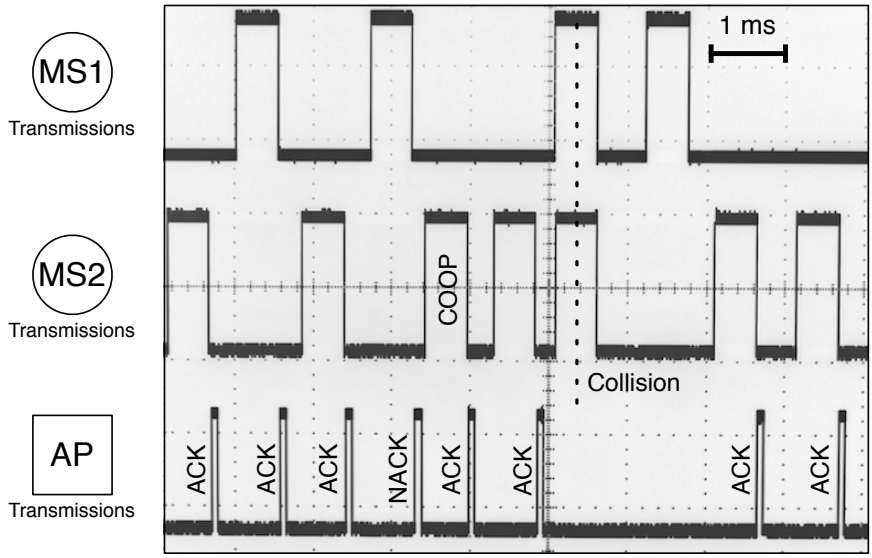


Figure 4.10: Oscilloscope-captured timeline of events.

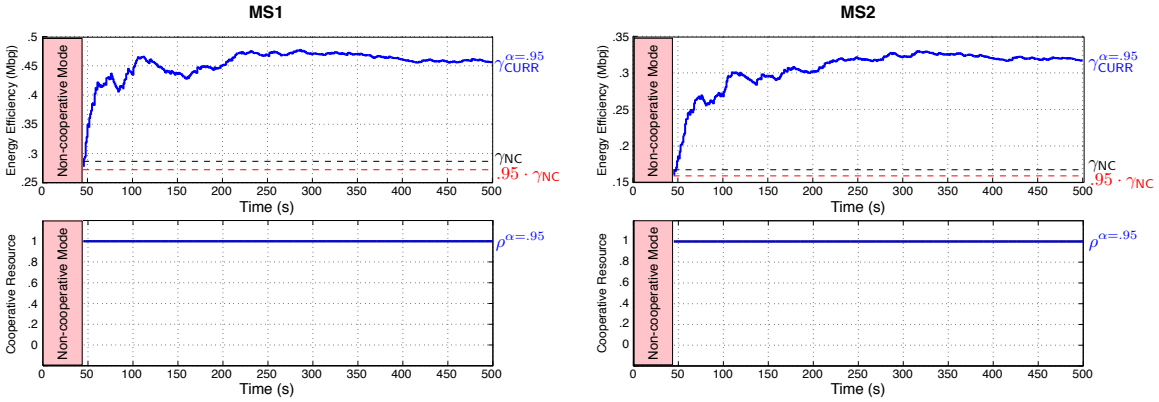
relay transmission power from MS2 was less than $\rho \cdot 14$ dBm.

4.5.2 Experimental Evaluation

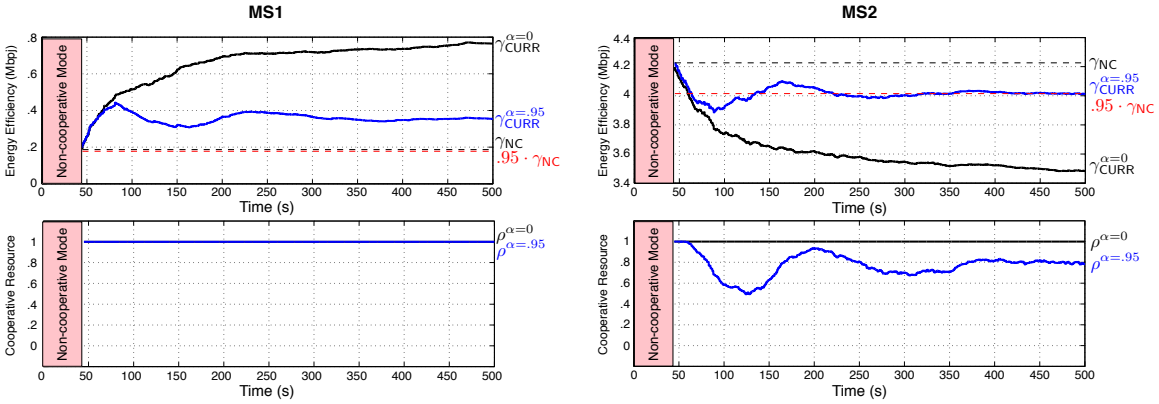
In Section 4.3, we demonstrated that the gains or losses in energy efficiency due to signal-scale cooperation are severely topologically dependent. First, in Sections 4.5.2.1 and 4.5.2.2, we revisit two of the same topologies and show that DECC can (a) preserve mutual cooperative energy efficiency gains when they occur and (b) protect a device from losses in energy efficiency due to cooperation.

Second, in Section 4.5.2.3, we investigate the DECC parameters that yielded the experimental results from the prior sections. Specifically, we empirically study the amount of time DECC needs to spend in a non-cooperative mode such that the measurement of γ_{NC} in the non-cooperative phase is representative of what γ_{NC} would be during the cooperative phase.

Finally, in Section 4.5.2.4, we demonstrate the full asynchronous mode-switching capability of DECC.



(a) Scenario A: Symmetric Topology



(b) Scenario B: Asymmetric Topology

Figure 4.11: Energy efficiency with DECC.

4.5.2.1 Revisiting Scenario A: Symmetric Topology

In Section 4.3.1.2, we observed that devices in a symmetric topology can provide equal amounts of cooperative assistance to one another and, therefore, provide mutual benefit. With DECC activated, devices should be able to recognize that they are each being aided by cooperation and do nothing to reduce cooperative resources; DECC should set $\rho = 1$ indefinitely.

Figure 4.11(a) shows a 500 second timeline of a trial for both devices (MS1 and MS2). During the first 45 seconds, both MS1 and MS2 refuse to act as cooperative relays for one another. During this period, γ_{NC} (and therefore, $\alpha \cdot \gamma_{\mathsf{NC}}$) are measured. For this trial we set $\alpha = .95$ so that devices would be willing to tolerate as much as a

5% reduction in energy efficiency. Because of the symmetric nature of the topology, both devices perform significantly better in the cooperative versus the non-cooperative mode. Throughout the entire experiment, DECC leaves the cooperative resource $\rho = 1$ because both devices are better off with cooperation than without it. The 50%+ improvement in energy efficiency that was measured in Section 4.3.1.2 is retained.

4.5.2.2 Revisiting Scenario B: Asymmetric Topology

In Section 4.3.1.3, the asymmetric topology caused MS1's energy efficiency to improve by over 320% and MS2's energy efficiency to degrade by 18%. With DECC activated, we expect MS2 to reduce the amount of effort it puts into cooperation in order to halt this loss.

Figure 4.11(b) shows a timeline for both devices in the asymmetric topology. For reference, we have included a trial for $\alpha = 0$ where the DECC control loop is effectively disabled. In this case, we can see that MS2 maintains a $\rho = 1$ despite performing significantly worse with cooperation than without; MS2 is completely selfless for $\alpha = 0$.

For $\alpha = .95$, however, we can see that MS2 lowers the cooperative resource ρ once γ_{CURR} falls below $.95 \cdot \gamma_{\mathsf{NC}}$. When this happens, γ_{CURR} at MS2 begins to increase back towards the non-cooperative performance while γ_{CURR} at MS1 begins to decrease since fewer cooperative resources are being provided by MS2. After some underdamped oscillations around the $.95 \cdot \gamma_{\mathsf{NC}}$ line, DECC's selection of ρ ensures that MS2 suffers no more than $1 - \alpha = 5\%$ loss of energy efficiency rather than the 18% loss suffered without DECC.

Despite the reduction in cooperative services provided by MS2 when DECC is activated, MS1 still sees nearly a $2\times$ improvement in energy efficiency. This shows that substantial cooperative gains can be achieved in return for small, bounded losses

in efficiency for a subset of nodes even in severely uneven scenarios.

4.5.2.3 Measurement Deviation Analysis

In the previous experiment, we chose a non-cooperative duration of 45 seconds and witnessed that the DECC control system at MS2 converged on a 5% loss in energy efficiency. In this section, we investigate this selection of duration.

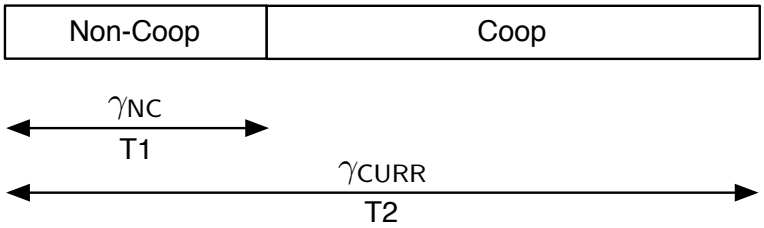


Figure 4.12: Ideally, the measurement of γ_{NC} during the T1 period should be representative of what γ_{NC} would be if measured throughout the T2 period.

Figure 4.12 illustrates a mode switch performed by the measurement module of DECC. During the T1 period, DECC measures the energy efficiency of a non-cooperative mode, γ_{NC}. When cooperation is allowed during the rest of T2, DECC implicitly relies on the measurement of γ_{NC} during T1 being representative of what γ_{NC} would be if it were measured throughout T2. To investigate the durations of T1 and T2 required for such reliance to be appropriate, we construct an experiment to measure the percentage change in energy efficiency measurements between periods T1 and T2. Formally, we measure the deviation[1]

$$d = \frac{|\gamma_{T1} - \gamma_{T2}|}{\gamma_{T1}}. \tag{4.3}$$

The goal of this experiment is to determine the deviation between the two periods if non-cooperative links were used in both. As such, we use our CSMA/CA imple-

[1]We note that the metric d in Equation 4.3 can be dominated by outliers if γ_{T1} is very small or zero. As such, we provide the median across all trials as well as the mean and standard deviation error bars.

mentation on WARP. We test the asymmetric topology described in Section 4.3.1.3. Furthermore, for ease of visualization, we consider the case where T2 = 4 · T1. In other words, we consider the $\frac{1}{4}$ duty cycle case where, for example, a measurement in one second is expected to be valid for the next three seconds.

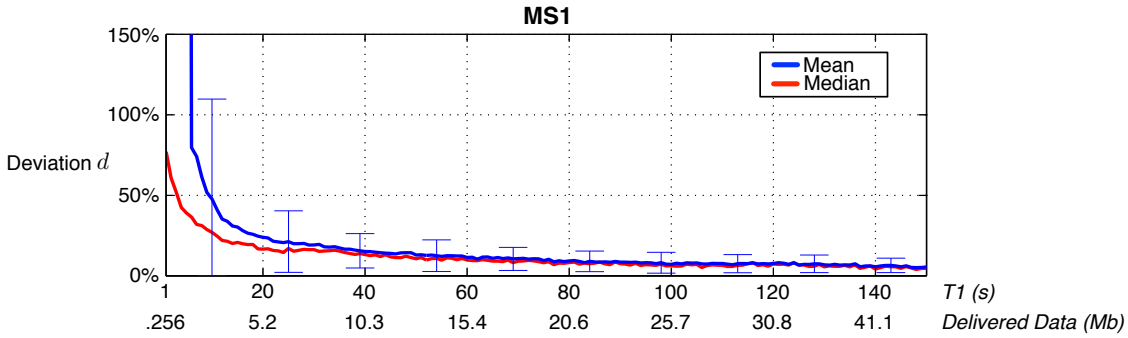


Figure 4.13: Because the energy efficiency of MS1 is so small, the percentage deviation can be quite large absent substantial measurement time.

Figure 4.13 shows the percent deviation between T1 and T2 for MS1. Recall that MS1 is the device that is a long distance away from the AP. In any given second, the achieved throughput of the device only ensures that, on average, 256 kilobits are successfully delivered. With the 1300 byte payloads being sent, this is an average of just over only 24 successful packet transmissions per second. As such, the measurement from one second to the next can deviate significantly. The 45 seconds of non-cooperative measurement in Section 4.5.2.2 resulted in a deviation of approximately 20% on average, which is far below the nearly 100% improvement measured with DECC set to $\alpha = .95$ and even further below the 320% improvement seen with $\alpha = 0$. As such, DECC is unlikely to confuse cooperative gain for cooperative loss despite the difficulty in measuring energy efficiency.

Figure 4.14 shows the percent deviation between T1 and T2 for MS2. Recall that MS2 was the device where the 5% loss (for $\alpha = .95$) was actually a binding constraint. As such, MS2's ability to measure γ_{NC} to well within 5% is paramount. With the 45 seconds of non-cooperative measurements, MS2 measures γ_{NC} to within

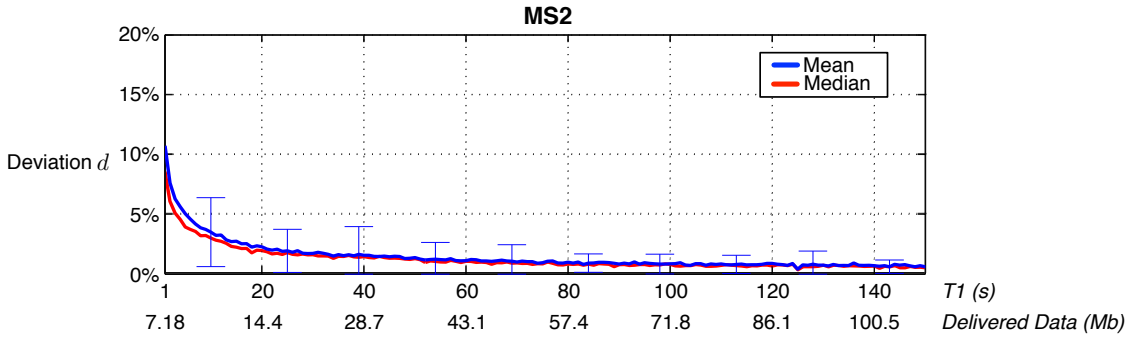


Figure 4.14: Because MS2 delivers far more data per unit time than MS1, its percentage deviation between T1 and T2 is far superior to that of MS1.

an approximately 2% deviation. Of course, if MS2 is less strict about how bounded its loss is (e.g. $\alpha = .85$ or lower), then even shorter non-cooperative durations can be tolerated since the deviation between measurements becomes small relative to the bound.

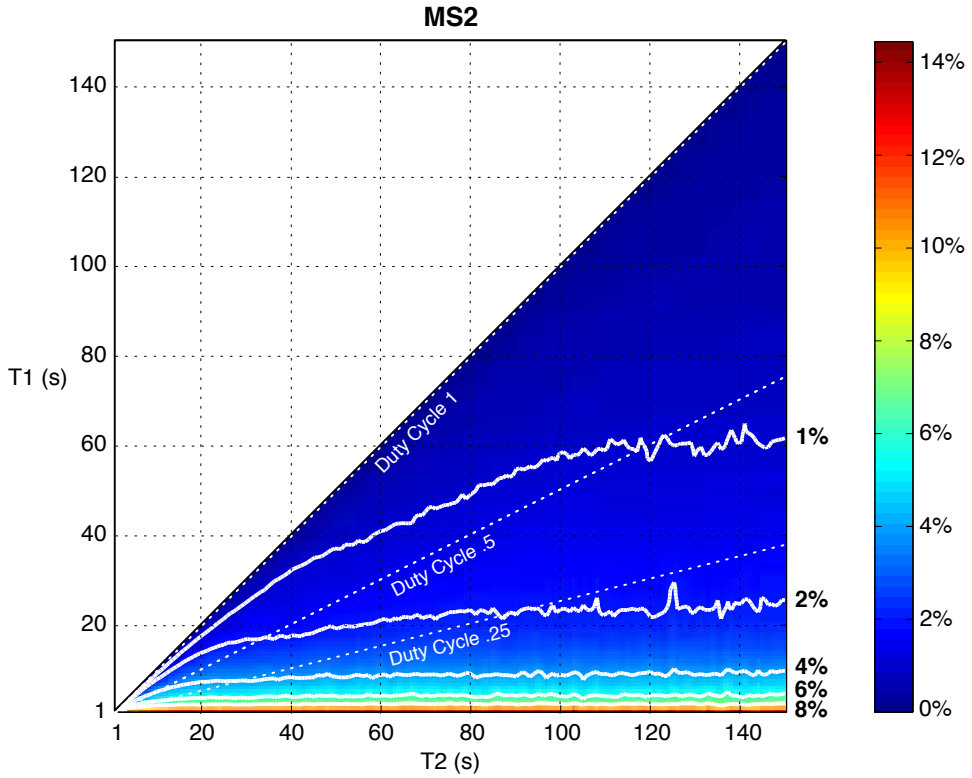


Figure 4.15: MS2 measurement deviation is represented in color and in contour lines.

In the previous analysis, a duty cycle of $\frac{1}{4}$ was assumed. To characterize the impact of duty cycle selection on energy efficiency measurements, we generalize our study in Figure 4.15. In this figure, measurement deviation between T1 and T2 are represented by color and contour lines along the [1%, 2%, 4%, 6%, 8%] ridges. When T1 = T2, the measurement deviation between T1 and T2 is zero because the measurements occur over exactly the same regions of time. Of course, a 100% duty cycle also means that cooperation can never be enabled to take advantage of the perfectly accurate measurement. The data in Figure 4.14 is, in fact, a cross section of Figure 4.15 along the line labelled "Duty Cycle .25."

We make no claims as to the universality of these specific T1 and T2 durations for general networks. The deviation from one measurement to the next is highly dependent on channel coherence times, traffic patterns, device mobility, and network size. However, the implementation presented in this thesis can be used for a future user study where large, mobile networks with actual user-generated traffic patterns affect these measurements. In this experiment, MS1 observed considerably higher measurement deviation between T1 and T2 than MS2 because MS1 delivers far fewer bits than MS2 in the same amount of time. For DECC, this means that devices that stand to lose energy efficiency due to cooperation will tend to be able to measure non-cooperative energy efficiency more easily. This is well-aligned with DECC's structure – by design, DECC treats *any* improvement in efficiency with cooperation as a reason to cooperate more. It is only when the device's energy efficiency is degraded by cooperation that accurate measurement of non-cooperative energy efficiency is very important. Said another way, the higher the deviation between measurements, the less such deviation matters for the operation of DECC.

4.5.2.4 Asynchronous Switching

As noted earlier, the previous DECC evaluation hinged on a single beginning-of-time non-cooperative measurement period of 45 seconds. In this section, we demonstrate that the asynchronous switching functionality of DECC specified in Section 4.4.1 holds. This functionality allows a device to effectively spread non-cooperative training over a longer window to allow the control system to begin adapting even if the underlying measurements are noisy. In this experiment, we consider a 1/4 duty cycle where, on average, one second of non-cooperative measurement supports the next three seconds where cooperation is allowed. To demonstrate that these mode switches are not coordinated between MS1 and MS2, we allow each device to enter a one second non-cooperative mode followed by a random cooperative duration chosen with a uniform random variable between zero and six seconds. In this way, MS1 and MS2 separately decide to disable cooperation and those decisions need not align.

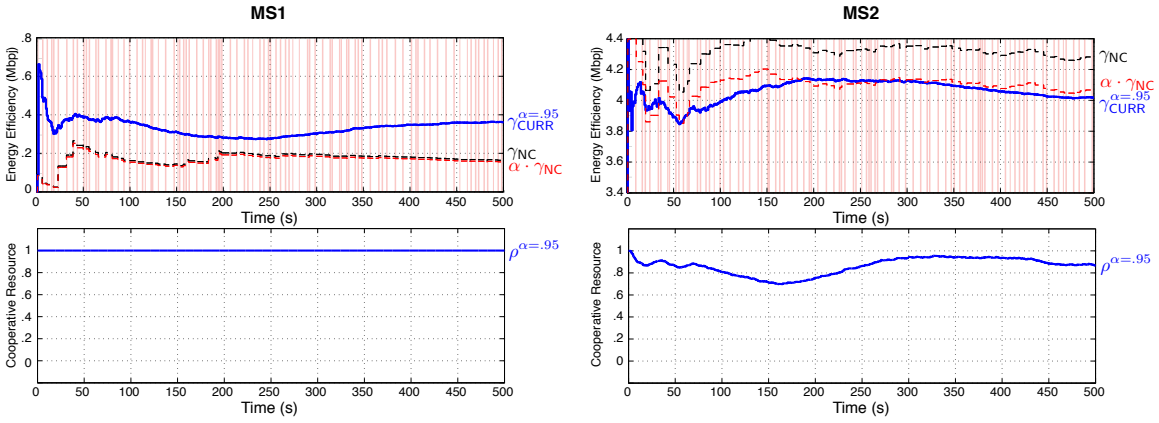


Figure 4.16: With DECC, each device can asynchronously switch between cooperative and non-cooperative modes.

In Figure 4.16, vertical red bars represent non-cooperative periods for MS1 and MS2. We can see that, over time, γ_{NC} is refined as cumulatively more time is spent in a non-cooperative mode. By 180 seconds into the experiment, MS2 has spent approximately $\frac{180}{4} = 45$ seconds measuring non-cooperative energy efficiency – equivalent

to the "one-shot" experiments in Sections 4.5.2.1 and 4.5.2.2. We can see that the steady-state behavior of the asynchronously switched DECC is similar to the behavior observed in Figure 4.11(b). The only difference is the DECC control system must track a moving target instead of the static measurement of γ_{NC}.

4.5.3 Larger Network Simulation

The previously described experiments present a targeted investigation of atomic topological relationships. In this section, we evaluate the scalability of DECC by considering a larger network. In this experiment, each device randomly chooses to enter its non-cooperative test mode independently from the other devices in the network. Furthermore, to demonstrate the distributed nature of DECC, we allow any device in the network to provide cooperative relaying assistance to any other device. Specifically, if any device (a) decodes the original transmission from the source and (b) decodes the NACK packet from the AP, it can act as a cooperative relay on that particular packet. If more than one relay transmits on a given packet, we assume they will not interfere with one another as they can employ conjugate beamforming to align the phase of their transmission to the channel estimate garnered from the reception of the NACK[1].

The channel emulator we have employed in earlier sections of this thesis is a 4×4 MIMO channel emulator and lacks a sufficient number of emulated paths to test networks larger than three devices. Instead, we have opted to implement DECC in the well-known ns-2 network simulator [61]. This implementation builds on the ns-2 implementation of PDOC presented in Section 3.2.4.

[1]We recognize that this kind of beamforming relies on channel reciprocity between the reception of the NACK and transmission of the cooperative packet. Until recently, phase reciprocity was assumed to be impossible in Wi-Fi like radios that switch between transmission and reception modes. However, modern (and ongoing) work has demonstrated that offsets between transmissions and receptions can be calibrated away [60].

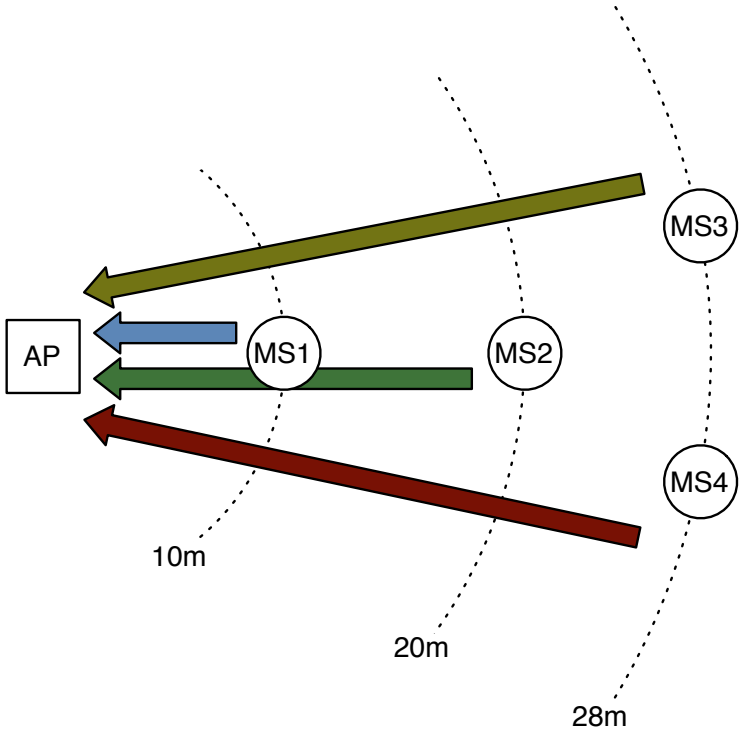


Figure 4.17: Network topology.

Figure 4.17 shows the topology under consideration. In Section 4.3.2, we established regimes of mutual benefit and one-sided benefit. In this simulation we consider a variety of devices at different distances away from an AP to sample points in these various regimes. The purpose of this topology selection is to showcase scenarios where devices may be simultaneously helped by some devices in the network and harmed by others. For example, from our study so far we expect that MS2 will be able to derive benefit from MS1. However, the existence of MS3 and MS4 on the fringes of the network will impact those gains in until-now unknown ways. In this simulation, we perform the same asynchronous mode switch of one second of non-cooperative mode to an average of three seconds of cooperation. Furthermore, to allow devices' measurements of γ_{NC} to stabilize, we have performed the simulation over a very long time of 2000 seconds. By the end of this simulation timeline, virtually no deviation in the cumulative measurement of γ_{NC} from one update to the next is observed.

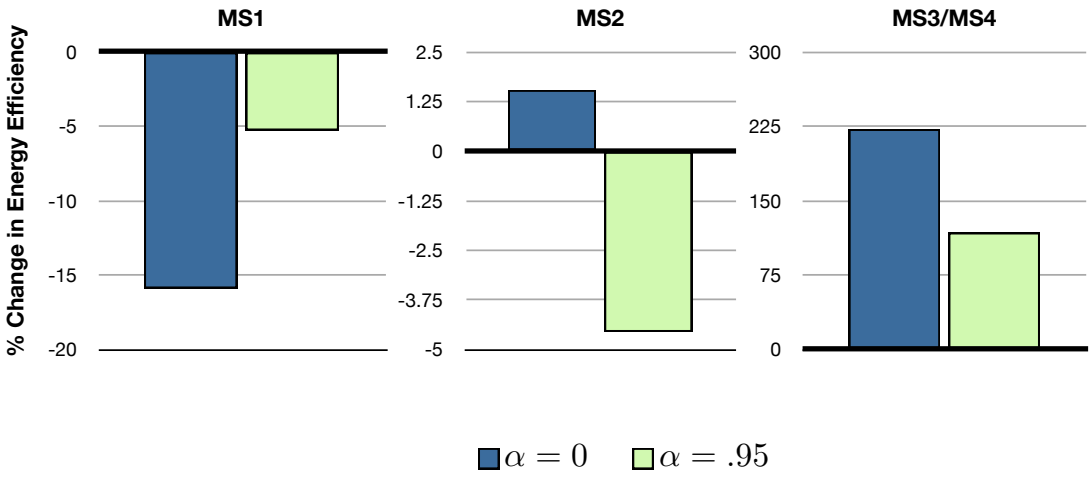


Figure 4.18: Results of larger network simulation simulation.

Figure 4.18 shows the percentage change of energy efficiency for each device in the network. Specifically, we plot $(\gamma_{\mathsf{CURR}} - \gamma_{\mathsf{NC}})/\gamma_{\mathsf{NC}}$ for every device. Furthermore, we consider the case of completely selfless devices ($\alpha = 0$) and the case of devices employing DECC for bounded altruism ($\alpha = .95$).

When $\alpha = 0$, MS1 acts as a cooperative relay completely selflessly. Since it is in a location that can benefit all of the other nodes, MS1 sees a significant degradation in energy efficiency of 15%. By enabling DECC with $\alpha = .95$, that loss is reduced to the specified 5%. This is consistent with our earlier investigations.

MS2, on the other hand, shows a surprising implication of DECC in large networks. When every device acts completely selflessly with $\alpha = 0$, the gains to MS2 from MS1 are enough to offset the costs of helping MS3 and MS4. Overall, MS2 sees a modest increase in energy efficiency. However, when $\alpha = .95$, the amount of gain that MS1 provides MS2 is reduced accordingly. This decrease in improvement for MS2 means that MS2 is now more burdened with acting as a cooperative relay; MS2 starts to suffer an energy efficiency loss. However, since MS2 is also implementing DECC, DECC ensures that MS2 does not suffer more than 5%. In this way, DECC shifts the burden of cooperation off of one user and distributes it to others.

The net effect for MS3 and MS4 is that DECC will lower the amount of improvement they see, but the gains are still considerable (2× the energy efficiency) considering the amelioration of harm on the other devices in the network.

4.6 Discussion

In a general network, the mobility of devices means that sometimes they may be close to an AP and subject to potential losses in energy efficiency due to cooperation. Other times, the devices will be on the outskirts of the network and enjoy tremendous improvements in energy efficiency. With DECC, we have created a way for devices to bound their potential losses to a tolerable level and still allow the achievement of large gains elsewhere in the network. With DECC, the answer to the question of "why should I help?" is simply "why not?". Devices have only a known, set amount of energy efficiency they could possibly lose and everything to possibly gain.

Chapter 5

Inter-device Interference

In Chapter 4, DECC allowed devices to become self-aware. Only when devices are directly benefitted by cooperation do they provide the maximum amount of cooperative relaying opportunities for others. In this chapter, we consider mechanisms that allow devices to become *network-aware* and change their relaying behavior based upon the interference it will create for other devices in the wireless network. This work was originally presented in [13].

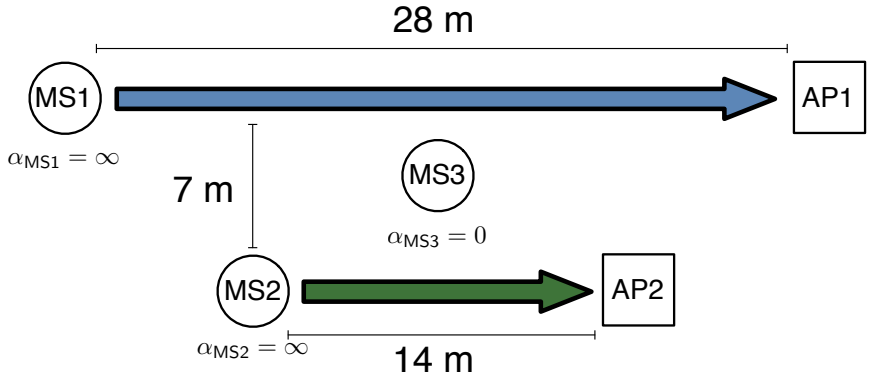


Figure 5.1: MS3 is completely altruistic and willing to assist MS1 and MS2 as much as it can. MS1 and MS2 are completely selfish and never act as cooperative relays.

Consider the topology in Figure 5.1. MS1 and MS2 source two independent flows to their own destination APs. MS3 is an idle device in the network that is not

actively sourcing its own traffic. Furthermore, each device implements DECC (and also, therefore, PDOC). For simplicity of discussion, assume that MS1 and MS2 are completely selfish with $\alpha = \infty$; they never will act as a cooperative transmitter. MS3, on the other hand, is completely altruistic with $\alpha = 0$. As such, MS3 is willing to cooperate with either MS1 or MS2 on any given packet.

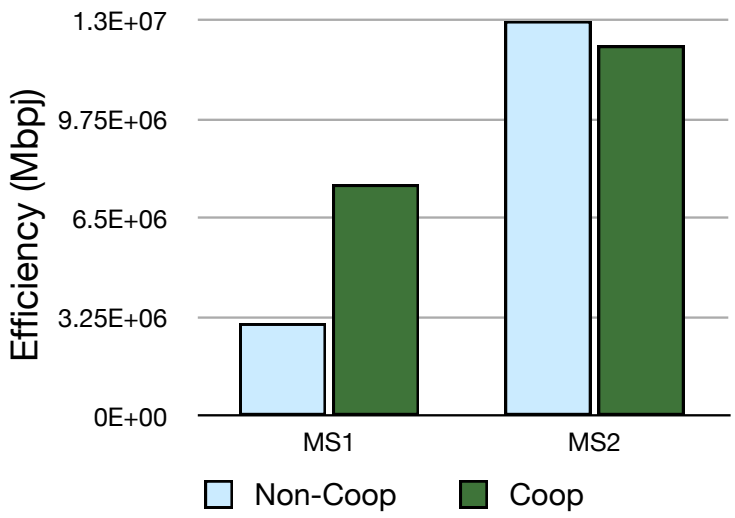


Figure 5.2: MS3 helps MS1 considerably but causes some harm to MS2.

Figure 5.2 shows the results of a simulation of this network. Despite MS3's best intentions to act completely selflessly in the assistance of others, we observe that its cooperative transmissions help the energy efficiency of MS1 only – and actually hurt that of MS2. In other words, even discounting the potential loss in energy efficiency of MS3 itself that DECC would mitigate (i.e. letting MS3 be completely altruistic in its actions), its cooperative assistance to the rest of the network is not universal. Because of the MS1-AP1 and MS2-AP2 separation differences, MS3 generally provides far more assistance to MS1 than it does to MS2. As such, on the whole, MS2 witnesses an increase in interference when MS3 transmits, without much corresponding assistance for its own flow.

To mitigate this, a "socially responsible" MS3 might scale back its cooperative assistance to MS1 so as to soften the losses to MS2. However, from MS3's perspective, network state information is necessary in order to identify this situation and respond appropriately to it. In distributed networks, assuming *full* network state information is not generally realistic, so MS3 needs some policy for determining when to cooperate given the incomplete network state information available to it. Development of such policies is the goal of this chapter.

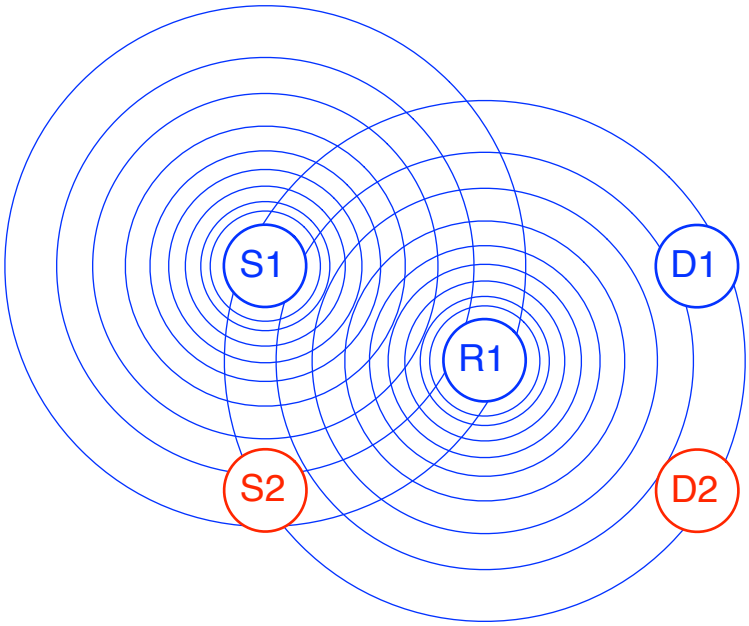


Figure 5.3: Relays can decrease spatial reuse by adding interference.

We focus on this interference management problem of cooperative relays by considering a simplified version of the earlier network. Specifically, we consider the five-device, two-flow network where a dedicated relay assists one flow but not the other, shown in Figure 5.3. From the perspective of S1, the relay can only provide improvement to its performance. From the perspective of S2, the relay can only increase the spatial footprint of the other flow and thus can only reduce the opportunities for spatial re-use (i.e. the simultaneous usage of the medium by both flows).

Our core contribution is a technique for protocol development for cooperative relays that addresses interference management. This technique applies for *any* subset of full network state information, which constitutes both the channel states of all links and device states. This contribution is framed by three key results. First, we propose a binary approximation for the network, simplifying all random variables to two-state variables. The binary approximation is then used to create relay access policies where the relay has zero, one, or two hops of channel information about the rest of the network. For each case of channel state information (zero, one, and two hops), we also consider the impact of whether the relay adopts a conservative or aggressive viewpoint about the *unknown* network state information. These access policies serve as guidelines to design cooperative protocols for actual wireless channels.

Second, we compare the partial information policies to the policy which has full information to quantify the performance impact of each piece of network state information. The six protocols ({aggressive or conservative} relay $\times$ {zero, one, or two} hops of channel knowledge) quantify an intuitive result. If the relay is aggressive and assumes the best case scenario about what is not known about the rest of the network, then gains can be significant for the cooperative flow but they come at the expense of significant loss for other flows in some topologies. In other words, aggression is generally not a socially responsible strategy. In contrast, a conservative relay, which aims to cause no harm to other flows, requires a substantial amount of network information to provide any significant cooperative gains. In short, a relay can be both helpful and socially responsible only if it has significant information about the state of the network. Otherwise, a conservative relay will end up staying silent most of the time, avoiding causing any harm to a neighboring flow but also avoiding providing much cooperative assistance at all.

Lastly, we close the loop by translating the relay access policies from the binary

approximation to SINR-based protocols and study their performance using network simulations. This last step is possible for all but the single-hop knowledge policies because the binary collision model turns out to be too crude to predict the behavior in this case. For the other four protocols, our simulations reveal that the trends predicted by the binary model hold for the fading channels.

The proposed computational mechanism for protocol design is inspired by the fact that designing medium access protocols with provable performance is often analytically and/or computationally intractable due to large state space. As a result, medium access protocols are often designed on a case-by-case basis with different amounts of network state information. As a notable exception, the authors in [62, and references therein] reverse engineer the exponential backoff structure of many random-access MAC protocols as a solution to a non-cooperative game. Similarly, the authors in [63] present an optimization-based framework for automated protocol design that solves an example scheduling problem. These works pursue a different methodology to a similar high-level goal: the construction of protocols in a procedural fashion.

5.1 Related Work

Our methodology is similar in spirit to recent work on deterministic approximation information theoretic analyses [64], where deterministic network models provide an insight into the design of Gaussian network models in many, but not all, cases – for example, the deterministic model in [64] is not a useful approximation of the MIMO channel.

As described in Chapter 2, information theoretic analyses of cooperative communication have a sizable body of literature [2, 3, 65, and references therein]. These works generally assume perfect network knowledge and centralized coordination to establish performance bounds on cooperative networks. In practice, global network

knowledge at every device is generally infeasible and/or not scalable as the size of the network increases. Accordingly, in contrast to these works, we study distributed cooperative protocols. The work in [66] studies the effects of interference and cooperative networks from the opposite perspective of our work. Whereas [66] studies the implications of interference on the performance of a cooperative flow, we design protocols that address the implications of increased interference caused by cooperative flows on the rest of the network.

While not specifically targeting cooperation applications, there exists a sizable body of literature on managing interference in ad hoc networks. These varied strategies range from altering carrier-sense thresholds according to network dynamics [67], to modifying the NAV structure of 802.11 to be less conservative [49, 68], and finally to using out-of-band busy tones to enhance channel reservations [69]. To address the main challenge in managing relay-induced interference, we have chosen to base our protocol design on standard CSMA/CA access mechanisms like the IEEE 802.11 DCF. Conceptually, we believe that the prior literature can be leveraged in the context of cooperative interference management by using the proposed framework.

5.2 System Model

In this section, we describe our signal model, decoding model, and carrier-sensing model. We then discuss physical layer relaying schemes and define the desired relaying policies.

5.2.1 Signal Model

We assume a slow fading model on the propagation of wireless signals. The reception of a transmission from a source device S at a destination device D in the presence of

interferers is represented by

$$y_{\mathsf{D}} = h_{\mathsf{SD}} x_{\mathsf{S}} + \sum_{i \in \mathcal{I}} h_{i\mathsf{D}} x_i + z_{\mathsf{D}}, \tag{5.1}$$

where y_{D} represents the received signal at D and x_{S} represents the transmitted signal from S. The multiplicative fade h_{ij} between devices i and j remains constant for *at least* the duration of x[1]. The additive noise z_{D} is assumed to be circularly symmetric complex Gaussian random variable that is drawn i.i.d. for every sample of x_i. The set $\mathcal{I}$ contains all other simultaneously transmitting sources in the network that act as interferers to S.

5.2.2 Decoding Model

We further assume an SINR-based decoding model that allows D to correctly decode a packet from S if and only if

$$\frac{H_{\mathsf{SD}} E[|x_{\mathsf{S}}|^2]}{\sum_{i \in \mathcal{I}} H_{i\mathsf{D}} E[|x_i|^2] + E[|z_{\mathsf{D}}|^2]} \geq \gamma_{\mathsf{DEC}}, \tag{5.2}$$

where $H_{ij} = |h_{ij}|^2 = |h_{ji}|^2$ represents the instantaneous, path-symmetric power of the fading channel, $E[\cdot]$ represents an expected value over the duration of the transmission x_{S}, and γ_{DEC} is an SINR detection threshold.

5.2.3 Carrier-sensing Model

When a device S is backlogged with packets to send and is currently receiving, it will pause the state of its backoff counter when the total received energy exceeds a

[1]In this formulation, we make no assumptions on the distribution from which h_{ij} is drawn. In Section 5.4 we will evaluate the protocols in a Rayleigh fading environment, but our proposed methodology applies to other channel distributions.

threshold, or

$$\sum_{i \in \mathcal{I}} H_{i\mathsf{S}} E[|x_i|^2] + E[|z_{\mathsf{D}}|^2] \geq \gamma_{\mathsf{CS}}, \tag{5.3}$$

where γ_{CS} is a carrier-sensing power threshold.

5.2.4 NACK-based Relaying Protocols

Many schemes for cooperative signaling have been proposed. For example, the two most common methods for signaling that can improve diversity in reception over direct transmission are the Amplify-and-Forward (AF) and Decode-and-Forward (DF) schemes [2]. In Chapter 3, we designed and implemented NACK-based cooperative MAC layers [11, 12] in WARP. We use this implementation as the basis for the MAC layer protocol development in this work.

5.2.5 Relaying Policies

We refer to the instantaneous snapshot of network dynamics as network state information (NSI). Consider a network of devices represented by the set $\mathcal{N}$. Two key components frame NSI,

$$\mathsf{NSI} := \left\{ \mathcal{H}^{\frac{|\mathcal{N}|(|\mathcal{N}|-1)}{2}}, \mathcal{X}^{|\mathcal{N}|} \right\}, \tag{5.4}$$

where

$$\mathcal{H} := \{H_{ij} | \forall i, j \in \mathcal{N}, i \neq j\} \tag{5.5}$$

$$\mathcal{X} := \{X_i | \forall i \in \mathcal{N}\} \tag{5.6}$$

represent the sets of channel states and transmission states in the network respectively. Note that the cardinality of $\mathcal{H}$ is $\frac{|\mathcal{N}|(|\mathcal{N}|-1)}{2}$ if self-channels are disallowed and channel gains are assumed to be path symmetric. Since the cardinality of $\mathcal{X}$ is $|\mathcal{N}|$, the

cardinality of NSI grows with the cube of the number of devices in the network, or $O(|\mathcal{N}|^3)$. Given a half-duplex constraint, a device i can either be transmitting or receiving at any given point in time[1], or $X_i \in \{\mathsf{Tx}, \mathsf{Rx}\}$. Let $\mathsf{R} \in \mathcal{N}$ represent a device in the network that is capable of acting as a cooperative relay for a flow of traffic in the same network and X_{R} represent its instantaneous transmission state. Additionally, let $\widehat{\mathsf{NSI}} \subset \mathsf{NSI}$ represent a subset of network state information available to the relay and $\psi \in \widehat{\mathsf{NSI}}$ represent a particular network state from the perspective of the relay. We define a *relaying policy* as the mapping of a known NSI state at R onto the transmission state of the relay, or

$$X_{\mathsf{R}} := f(\psi), \tag{5.7}$$

where $f : \widehat{\mathsf{NSI}} \to \{\mathsf{Tx}, \mathsf{Rx}\}$. We distill the task of cooperative policy design down to determining this functional mapping for a particular objective: to maximize the rate of a cooperative flow while minimizing any rate degradations in noncooperative flows of the network. In other words, we aim to minimize the spatial reuse degradation that can be caused by cooperative relays by eliminating relay transmissions in cases where doing so would harm a nearby flow.

5.3 Binary Approximation and Policy Design

In the model described in Section 5.2, NSI contains channel fades that are supported over a continuum of values. In this section, we develop a binary model, NSI^B, as an approximation of the full NSI, for the five device, two-flow network in Figure 5.3. We show that the states in this model can be explicitly classified by the effect that

[1]We limit the discussion to devices that can only transmit and receive. Our approach can easily be employed to consider applications such as sensor networks where devices may have additional states such as being idle.

relay transmission *would* have on the network if the relay were to transmit in such states. We then define relaying policies that operate with incomplete $\widehat{\mathsf{NSI}}^B \subset \mathsf{NSI}^B$ and evaluate their relative performances using the binary approximation.

5.3.1 Network Model Approximation

We approximate the signal and detection models presented in Section 5.2 in two fundamental ways. First, we consider a binary approximation of instantaneous channel fades where $H_{ij}^B \in \{0,1\}$ is a Bernoulli random variable with parameter $p_{H_{ij}^B}$. In effect, $p_{H_{ij}^B}$ acts as a proxy for SNR_{ij} where high SNR_{ij} ($p_{H_{ij}^B} \to 1$) makes a high gain channel ($H_{ij}^B = 1$) very likely. Additionally, device state $X_i \in \{0,1\}$ is a Bernoulli random variable with parameter p_{X_i} where $X_i = 0$ represents reception and $X_i = 1$ represents transmission.

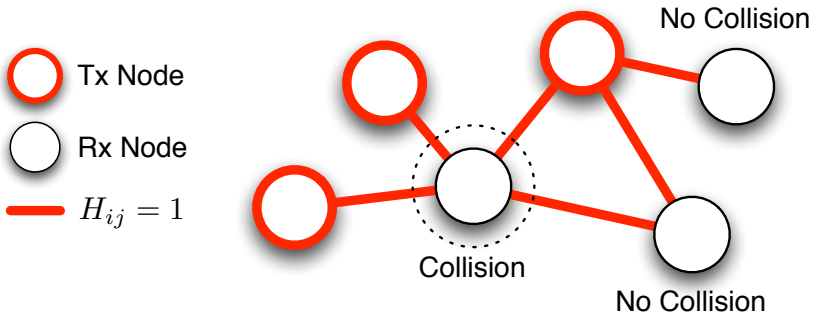
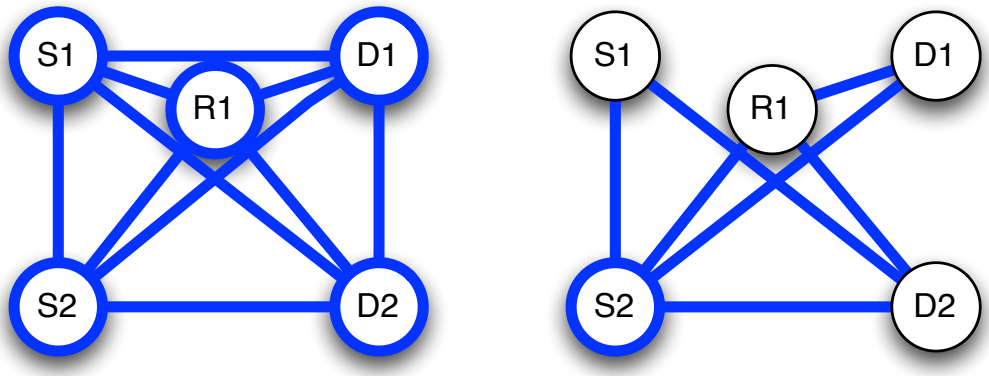


Figure 5.4: Devices form vertices and channel fades form edges in the network graph.

Second, we approximate the SINR-based detection model in Section 5.2 with a graph-based collision model illustrated in Figure 5.4. In this model, devices form vertices that are interconnected by the instantaneous edges formed by H_{ij}^B. If two devices m and n are both in transmit states $X_m = X_n = 1$, and are linked to a common receiver l with $H_{ml}^B = H_{nl}^B = 1$, then a collision occurs and neither transmission is decodable. We note that the binary collision model, without the probability law on



(a) 15 network state elements. (b) 8 network state elements.

Figure 5.5: The highlighted links and devices represent network elements that can take on active or inactive states.

the links, is commonly used in medium access layer protocol design [70].

In Section 5.4, we remove both of these assumptions and translate the policies generated using the binary approximation into SINR-based cooperative protocols.

The shift to binary-valued network states reduces the uncountably infinite number of states that make up NSI to a finite number. That said, the cardinality of NSI^B still grows with the cube of the number of devices in the network just like its continuous-valued counterpart in Equation (5.4). For tractability, we limit our study to the case of the two-flow, five device network graph shown in Figure 5.5. The device R1 represents a relay device that is *a priori* paired with source S1. Figure 5.5(a) shows that 15 possible random variables (10 channel states + 5 device states) make up any given snapshot of the network. Since each of these 15 bits can take on one of two values, there are a total of $2^{15} = 32768$ possible network states. To reduce this state space to a more manageable size, we limit the scope of the discussion to relay policies designed for the NACK-based cooperative protocols discussed in Section 5.2 and presented in Chapter 3. This reduction allows us to focus on a *relay-centric* network model that ignores all interactions that are unaffected by relay activity.

The goal of this study is to determine the conditions under which the relay should transmit (i.e. when $X_{\mathsf{R1}} = \mathsf{Tx}$). Note that this choice of whether to transmit can occur in concert with the decisions that fall out of DECC regarding whether to act as a relay to begin with. A relay that chooses not to transmit due to inter-device interference concerns will essentially save energy, and, if DECC is enabled, will redirect this energy towards other flows that it can help in a more socially responsible manner.

Let $\psi^B \in \mathsf{NSI}^B$ represent a single state of the network. This state is formed by the $H^B_{ij\in\mathcal{N}}$ and $X_{i\in\mathcal{N}}$ bits present in the two-flow network. We need not consider the value of X_{R1} in the construction of ψ^B because the goal is to determine X_{R1} as a function of the other elements. By assuming a NACK-based protocol where the relay is only ever requested to transmit under the condition that its source is unable to communicate to its destination, we can further reduce the following states as follows:

- $X_{\mathsf{S1}} \equiv 1$: A NACK from $\mathsf{D1}$ triggers *simultaneous* transmissions at $\mathsf{S1}$ and $\mathsf{R1}$. If it decides to transmit, $\mathsf{R1}$ will overlap transmission with $\mathsf{S1}$.

- $X_{\mathsf{D1}} \equiv 0$: If the cooperation request is signaled by $\mathsf{D1}$ via a NACK, then $\mathsf{D1}$ knows that a cooperative transmission is to follow and it will not initiate any transmissions.

- $X_{\mathsf{D2}} \equiv 0$: In general, $\mathsf{D2}$ can potentially generate transmissions in the form of ACK/NACK control packets meant for $\mathsf{S2}$. To reduce the number of states that must be considered, we assume that this cannot occur. In Section 5.4, we broaden the policies generated by this model to include an arbitrary number of flows among an arbitrary number of devices. Since flows can be bidirectional, this effectively also captures the case of interference caused by control packets and thus relaxes this assumption.

- $H^B_{\mathsf{S1D1}} \equiv 0$: In a reactive cooperative protocol, relay transmissions occur only

when the corresponding source transmission fails due to insufficient channel gain. Thus, we can assume that this channel is disconnected[1].

- $H^B_{\mathsf{S1R1}} \equiv 1$: Given the decode-and-forward physical layer operating at the relay, the link between S1 and R1 must be connected for the relay to transmit.

Figure 5.5(b) shows that this conditional model reduces the number of state elements in the network to only 8, leaving a far more manageable total of $2^8 = 256$ possible states.

5.3.2 State Classification

A relay transmission can have a number of effects on the network as a whole. We classify these effects into three sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. Set $\mathcal{A}$ contains all states where a relay transmission *assists* the S1-D1 flow in recovering a packet. Set $\mathcal{B}$ contains all states where S2 is forced to defer a *backoff* while receiving when it otherwise would not because of R1 triggering a carrier-sense. Finally, set $\mathcal{C}$ contains all states where D2 fails to decode a message from S2 because of a *collision* caused by R1. Formally,

$$\mathcal{A} \in \{\psi^B | H^B_{\mathsf{R1D1}} \overline{X_{\mathsf{S2}} H^B_{\mathsf{S2D1}}} = 1\} \tag{5.8}$$

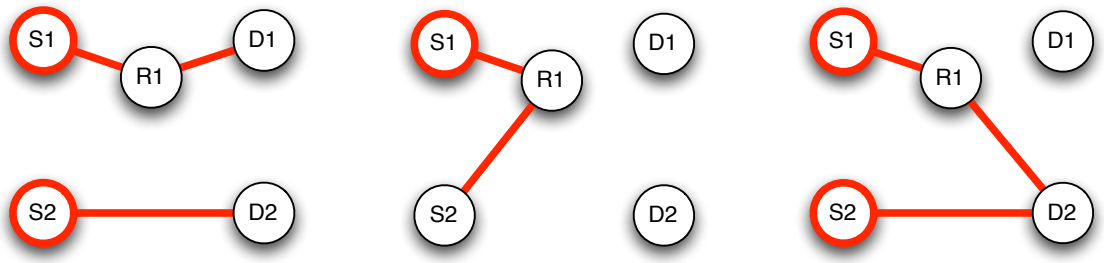
$$\mathcal{B} \in \{\psi^B | \overline{X_{\mathsf{S2}}} H^B_{\mathsf{R1S2}} \overline{H^B_{\mathsf{S1S2}}} = 1\} \tag{5.9}$$

$$\mathcal{C} \in \{\psi^B | X_{\mathsf{S2}} H^B_{\mathsf{S2D2}} H^B_{\mathsf{R1D2}} \overline{H^B_{\mathsf{S1D2}}} = 1\} \tag{5.10}$$

where the overline represents a logical complement.

Figure 5.6 highlights three example network states ψ^B that occupy the subsets $\mathcal{A}$,

[1] A relaying phase is triggered by an explicit NACK broadcast from destination to source and relay. Hence, an assumption that $H^B_{\mathsf{S1D1}} = 0$ appears dissatisfying since the NACK must be communicated over this channel back to the source. In practice, NACKs can be coded at far lower rates and thus be far more resilient to channel outages than data payloads. Thus, $H^B_{\mathsf{S1D1}} = 0$ represents the case where the channel gain is low enough to not support a full data payload yet high enough to support a NACK.



(a) Assist Example ($\mathcal{A}$). (b) Backoff Example ($\mathcal{B}$). (c) Collision Example ($\mathcal{C}$).

Figure 5.6: Every state can be labelled with membership in the $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ subsets.

$\mathcal{B}$, and $\mathcal{C}$. Since events $\mathcal{B}$ and $\mathcal{C}$ correspond to mutually exclusive events ($\mathsf{S2}$ reception and transmission respectively), these subsets are also mutually exclusive. Using Equations (5.8) through (5.10), we label each network state ψ^B with its membership in these subsets, or

$$\psi^B \in \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{A} \cap \mathcal{B}, \mathcal{A} \cap \mathcal{C}, \mathcal{D}\}, \tag{5.11}$$

where $\mathcal{D}$ represents a set of states where relay transmission has neither positive nor negative impact on the network. In Appendix D, we classify each possible network state.

5.3.3 Relaying Policies with Partial NSI

In the previous section, we showed that network states can be classified according to the relay's effects on the network. Given these labels, relaying policies can be derived that govern whether a relay should transmit as a function of the current state of the network ($X_{\mathsf{R1}} = f\left(\psi^B\right)$). In this section, we first define relaying policies assuming that the relay is fully aware of the current global network state ψ^B. We then consider relaying policies where the relay has incomplete network state information ($\widehat{\mathsf{NSI}}^B$).

5.3.3.1 Full NSI

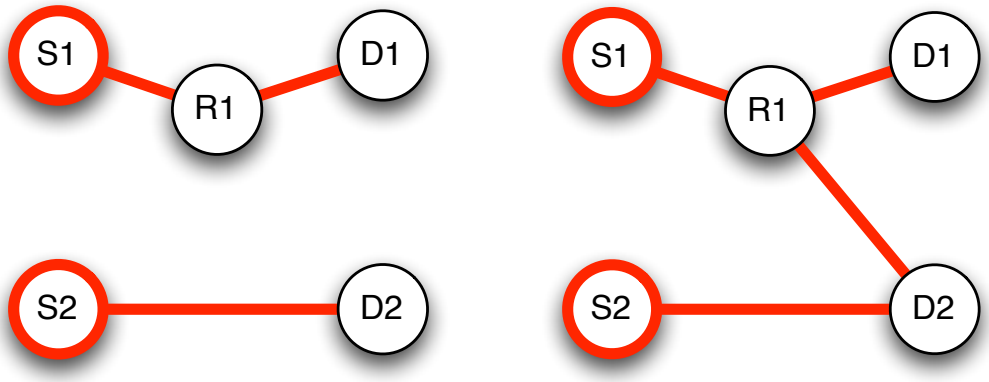
When a relay has access to full NSI^B, it can accurately determine the current state of the network ψ^B. As such, the relay knows perfectly what effect transmission during this state will have on the network as a whole. We can define a relay policy that minimizes negative impact on a surrounding flow by disallowing transmission when ψ^B is labelled with events $\mathcal{B}$ or $\mathcal{C}$ since these reduce spatial reuse by interfering with the operations of the other flow:

$$X_{\mathsf{R1}} = \begin{cases} 0 & \text{if } \psi^B \in \mathcal{B} \cup \mathcal{C} \\ 1 & \text{if } \psi^B \in \mathcal{A} \cap \left(\overline{\mathcal{B} \cup \mathcal{C}}\right) \\ Z & \text{otherwise,} \end{cases} \tag{5.12}$$

where Z represents a "don't care" where neither a relay transmission nor the lack thereof will impact the network in any way. Counting the number of states that are members of $\mathcal{B}$ or $\mathcal{C}$ in Appendix D, we see that 48 of the 256 total states represent conditions where the relay should avoid transmitting. One can write the Boolean expression that ties the values of the individual network state elements to the behavior of the relay $\overline{X_{\mathsf{R1}}}$ (the relay avoiding transmission). One can employ standard Boolean reduction techniques to convey this behavior more simply than the sum-of-products form of 48 cases, or

$$\overline{X_{\mathsf{R1}}^{\mathsf{FNSI}}} = \underbrace{X_{\mathsf{S2}} H_{\mathsf{S2D2}}^B H_{\mathsf{R1D2}}^B \overline{H_{\mathsf{S1D2}}^B}}_{①} + \underbrace{\overline{X_{\mathsf{S2}}} H_{\mathsf{R1S2}}^B \overline{H_{\mathsf{S1S2}}^B}}_{②}, \tag{5.13}$$

In this expression, we use the FNSI acronym as representation of the "Full NSI" policy. There are two critical components to this behavior. The ① term addresses the potential for the relay to cause a packet drop due to a collision with a transmission



(a) $\overline{X_{\mathsf{R1}}^{\mathsf{FNSI}}} = 0$ since $\psi^B \in \mathcal{A}$. (b) $\overline{X_{\mathsf{R1}}^{\mathsf{FNSI}}} = 1$ since $\psi^B \in \mathcal{C}$.

Figure 5.7: If H_{R1D2} is unknown, the relay cannot distinguish between these two states.

from S2. Specifically, the relay should avoid transmitting when S2 is transmitting and S2 would not collide with a transmission from S1 but would collide with a transmission from R1. The ② term addresses the potential for the relay to cause unnecessary backoff deferrals at S2. The relay should avoid transmitting when S2 is receiving and no link is present between S1 and S2 but a link is present between S1 and R1. This behavior establishes the baseline performance of a relaying policy that has access to all of the elements required to calculate Equation (5.13). The power of this methodology lies in the fact that we can also determine the relay behavior for any arbitrary subset of NSI^B.

5.3.3.2 Incomplete NSI

One can use exactly the same full NSI^B table in Appendix D to construct incomplete NSI^B policies by recognizing that eliminating knowledge is equivalent to binning network states into coarser delineation.

Figures 5.7(a) and 5.7(b) show two network states where the full NSI policy enables and disables relay transmission respectively. However, the only difference between

the network states is the H_{R1D2} state element. If this state element is unknown to the relay, the two states are binned together creating a *conflict set* where the lack of knowledge yields ambiguity in what the relay should do; the relay is unable to determine whether the current state of the network is in an assist classification $\mathcal{A}$ or also in a collision classification $\mathcal{C}$. In dealing with these conflict sets that arise with incomplete knowledge available to the relay, we consider two approaches to this problem:

Conservative View: When a relay is unable to distinguish between multiple states, it may assume the worst about the state elements it does not know. This assumption yields a disabled relay ($X_{\mathsf{R1}} = 0$) in the case that *any* state within the conflict set demands a disabled relay.

Aggressive View: Adopting the best-case viewpoint about unknown states, a relay can enable transmission ($X_{\mathsf{R1}} = 1$) when *any* state within the conflict set demands relay transmission.

These approaches apply to any arbitrary subset of the full NSI^B knowledge, so a remaining task is to determine what subsets of full NSI^B to consider. A useful way of sorting NSI^B is considering the hop-distance of the NSI^B elements from the relay. This approach allows a quantitative description of how "local" a device's view of the network is [71].

Let $\widehat{\mathsf{NSI}}^B(n)$ represent the set of NSI elements no further than n hops away from the relay. In our two-flow network, these sets are defined as

$$\widehat{\mathsf{NSI}}^B(2) = \{H^B_{\mathsf{S1S2}}, H^B_{\mathsf{R1S2}}, H^B_{\mathsf{S1D2}}, H^B_{\mathsf{S2D2}}, H^B_{\mathsf{R1D2}}, H^B_{\mathsf{R1D1}}, H^B_{\mathsf{S2D1}}\} \tag{5.14}$$

$$\widehat{\mathsf{NSI}}^B(1) = \{H^B_{\mathsf{R1S2}}, H^B_{\mathsf{R1D2}}, H^B_{\mathsf{R1D1}}\} \tag{5.15}$$

$$\widehat{\mathsf{NSI}}^B(0) = \{\emptyset\}. \tag{5.16}$$

In the case of $\widehat{\mathsf{NSI}}^B$ (2), all wireless channels are at least two hops away so the relay knows full NSI^B with the notable exception of the transmission state of $\mathsf{S2}$ (i.e. X_{S2}).

In the following sections we use the notation $\mathsf{Cons}(n)$ to identify policies that use the conservative mapping with n hops of knowledge. Similarly, the notation $\mathsf{Aggr}(n)$ is used to identify policies that use the aggressive mapping.

Conservative Policies:

Using the same Boolean reduction techniques as before, conservative relaying policies can be identified for different numbers of hops of information made available to the relay.

$$\overline{X_{\mathsf{R1}}^{\mathsf{Cons}(2)}} = H_{\mathsf{S2D2}}^B H_{\mathsf{R1D2}}^B \overline{H_{\mathsf{S1D2}}^B} + H_{\mathsf{R1S2}}^B \overline{H_{\mathsf{S1S2}}^B} \tag{5.17}$$

$$\overline{X_{\mathsf{R1}}^{\mathsf{Cons}(1)}} = H_{\mathsf{R1S2}}^B + H_{\mathsf{R1D2}}^B \tag{5.18}$$

$$\overline{X_{\mathsf{R1}}^{\mathsf{Cons}(0)}} = 1. \tag{5.19}$$

In the full NSI^B case in Equation (5.13), the X_{S2} acts as a kind of switch to determine whether the relay's behavior is dominated by collision avoidance or backoff deferral avoidance. In Equation (5.17), this switch is missing and both terms apply because the two-hop policy does not have access to the device state. In Equation (5.18), the relay is able to base its decision of whether to transmit only on the set $\widehat{\mathsf{NSI}}^B$ (1). Acting conservatively, the relay is able to transmit only when the links between $\mathsf{R1}$ and *both* $\mathsf{S2}$ and $\mathsf{D2}$ are disconnected. The relay guarantees that it cannot cause a backoff deferral at $\mathsf{S2}$ or a collision at $\mathsf{D2}$. Finally, the relay in Equation (5.19) is never able to transmit since it can never guarantee that it will not harm another flow.

Aggressive Policies:

Aggressive relaying policies can be identified for different numbers of hops of

information made available to the relay.

$$\overline{X_{\mathsf{R1}}^{\mathsf{Aggr}(2)}} = H_{\mathsf{S2D2}}^{B} H_{\mathsf{R1D2}}^{B} \overline{H_{\mathsf{S1D2}}^{B}} \cdot H_{\mathsf{R1S2}}^{B} \overline{H_{\mathsf{S1S2}}^{B}} + \overline{H_{\mathsf{R1D1}}^{B}} \tag{5.20}$$

$$\overline{X_{\mathsf{R1}}^{\mathsf{Aggr}(1)}} = \overline{H_{\mathsf{R1D1}}^{B}} \tag{5.21}$$

$$\overline{X_{\mathsf{R1}}^{\mathsf{Aggr}(0)}} = 0. \tag{5.22}$$

The key difference between Equations (5.13) and (5.20) is that if only one condition (collision or backoff deferral) instructs the relay to halt, it is assumed that the unknown X_{S2} state element would have disabled that term. In other words, the relay halts transmissions only when either a $\mathcal{B}$ or $\mathcal{C}$ event would occur regardless of the X_{S2} state. Additionally, another case for disabling the relay appears when R1 and D1 are disconnected since the relay cannot assist the cooperative flow in this case. In Equation (5.21), the relay disables transmission only when it knows that it will not be able to help. In these cases, there is *only* an opportunity to harm the network, so even the aggressive relay disables transmission. Finally, the relay in Equation (5.22) knows nothing about the network and aggressively transmits whenever it is requested.

5.3.4 Discussion of Protocol Overhead

The binary model-based relay policies dictate the behavior of the relay *given* elements of NSI. In this section, we discuss how such information might be collected in actual protocols. The amount of overhead for collecting this information is determined by two factors: (i) the rate of change of NSI and (ii) how much knowledge is desired.

First, the rate of change of NSI depends on the amount of mobility in the system. For low-mobility, slow-fading environments such as indoor Wi-Fi, channel coherence times can be many tens or hundreds of packet intervals. As such, NSI knowledge at the relay need only be updated on the timescales of these coherence times. Second, the

amount of overhead required differs from one NSI element to the next. For instance, the one-hop NSI states may be logged passively (with zero overhead) at the relay by simply overhearing surrounding transmissions. In fact, even some two-hop knowledge may be acquired without additional overhead. Assuming the non-cooperative flow employs the same NACK-based protocol as the cooperative flow, the relay can infer the link quality between the non-cooperative source and destination by overhearing ACKs and NACKs generated by the non-cooperative destination.

5.3.5 Performance Evaluation

As a mechanism to compare the performance of different policies, we evaluate the probability of the network entering a particular event subset while simultaneously considering whether a relaying policy transmits. In other words, a relaying policy can be penalized for transmitting within the $\mathcal{B}$ or $\mathcal{C}$ event subsets and rewarded for transmitting in $\mathcal{A}$. If a relay transmits in $\mathcal{A}$, the throughput and energy efficiency of the device it is helping will improve. If the relay transmits in $\mathcal{B}$ or $\mathcal{C}$, the throughput and energy efficiency of the other device in the network will degrade. The probability of a relay transmitting in event $\mathcal{A}$ is

$$Pr\{X_{\mathsf{R1}} \cap \psi^B \in \mathcal{A}\} = \sum_{\psi^B \in \mathcal{A}} X_{\mathsf{R1}}\left(\psi^B\right) \cdot Pr\{\psi^B\}, \tag{5.23}$$

where $Pr\{\psi^B\}$ can be calculated by the product of the Bernoulli parameters. Similar expressions for event spaces $\mathcal{B}$ and $\mathcal{C}$ can be derived.

It is useful to consider a particular application scenario where the R1 device is geographically near the S1 device. This models a usage case where one user owns both the relay and source devices and both devices are located near the user. Such a scenario may be particularly suited for cooperation, as observed earlier in Sections 3.1

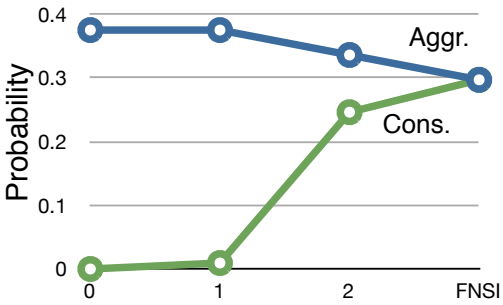
and 4.3. Furthermore, let us simplify the discussion of these systems by considering a single dominant parameter: flow separation. Specifically, let $p_{H^B_{\mathsf{S1S2}}} = p_{H^B_{\mathsf{S1D2}}} = p_{H^B_{\mathsf{R1S2}}} = p_{H^B_{\mathsf{R1D2}}} = p$ where p represents a single parameter that acts as a proxy for flow separation. As $p \to 1$, the flows are topologically connected with high probability, and as $p \to 0$, the flows are disconnected with high probability. For simplicity of discussion, assume every other state element probability is $\frac{1}{2}$. Using Equation (5.23), we compute expressions that determine the propensity of each policy to transmit in the $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ subsets as a function of the single independent parameter p.

Table 5.1: Performance Evaluation of Relaying Policies

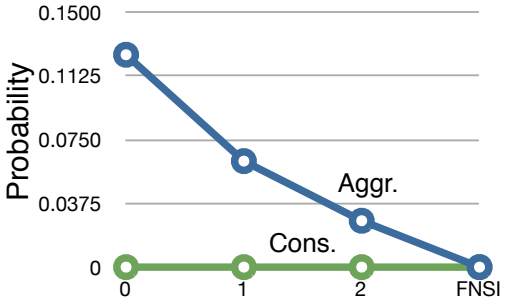
Policy	$Pr\{X_{\mathsf{R1}} \cap \psi^B \in \mathcal{A}\}$	$Pr\{X_{\mathsf{R1}} \cap \psi^B \in \mathcal{B}\}$	$Pr\{X_{\mathsf{R1}} \cap \psi^B \in \mathcal{C}\}$
FNSI	$\frac{5p^2-5p+6}{16}$	0	0
Cons(2)	$\frac{3p^4-6p^3+10p^2-7p+2}{16}$	0	0
Cons(1)	$\frac{3p^2-6p+3}{8}$	0	0
Cons(0)	0	0	0
Aggr(2)	$\frac{-p^4+2p^3-p+3}{8}$	$\frac{-p^4+2p^3-3p^2+2p}{16}$	$\frac{-p^4+2p^3-2p^2+p}{8}$
Aggr(1)	$\frac{3}{8}$	$\frac{-p^2+p}{4}$	$\frac{-p^2+p}{8}$
Aggr(0)	$\frac{3}{8}$	$\frac{-p^2+p}{2}$	$\frac{-p^2+p}{4}$

Table 5.1 summarizes the performance of each of the six previously described relaying policies. Of note, the full NSI^B policy and the conservative incomplete NSI^B policies never transmit in the subsets where doing so could potentially cause a collision or backoff deferral event. As such, the probability of harming the other flow by transmitting on these occasions is zero. The aggressive policies, however, allow some degradation in the other flow in order to improve the policies' abilities to assist their own flows.

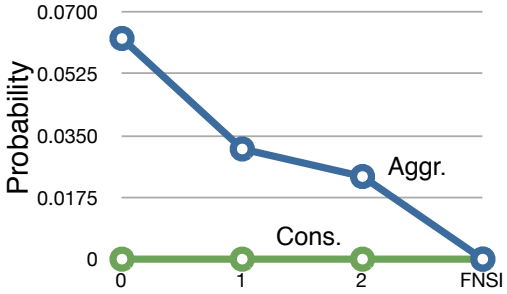
Consider the case that $p = \frac{1}{2}$. Figure 5.8 shows the performance of each policy as a function of NSI^B available to R1. In general, the trend is that more $\widehat{\mathsf{NSI}}^B$ knowledge results in less harm to another flow since the relay knows more about the network



(a) Probability of assisting.



(b) Probability of causing backoff deferral.

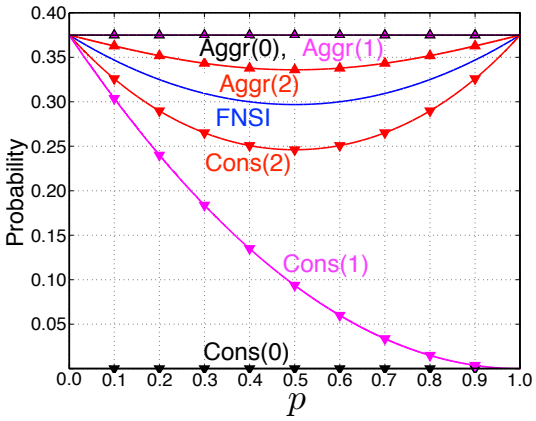


(c) Probability of causing collision.

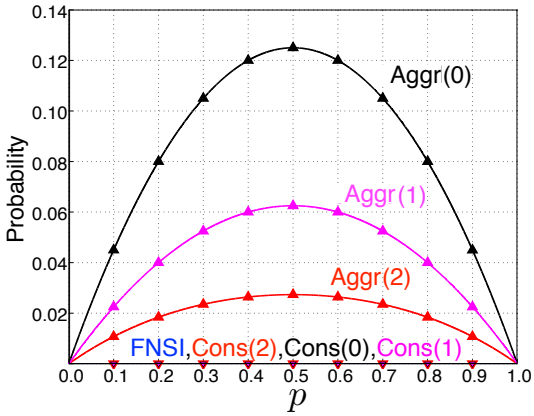
Figure 5.8: Conservative relaying behavior requires substantial NSI before cooperative gain is observed.

it needs to protect. Likewise, increasing $\widehat{\mathsf{NSI}}^{B}$ knowledge allows conservative relays to assist their flow more and eventually converge with their aggressive counterparts. Incrementally, the jump from zero hops of knowledge to one hop of knowledge has very little effect on the conservative policies – the improvement seen in performance of the cooperative flow is marginal. For the aggressive policies, however, having even a single hop of information provides a substantial drop in the amount of harm the relay will impart on the neighboring flow. Conservative policies require large amounts of NSI before cooperative gains can be seen.

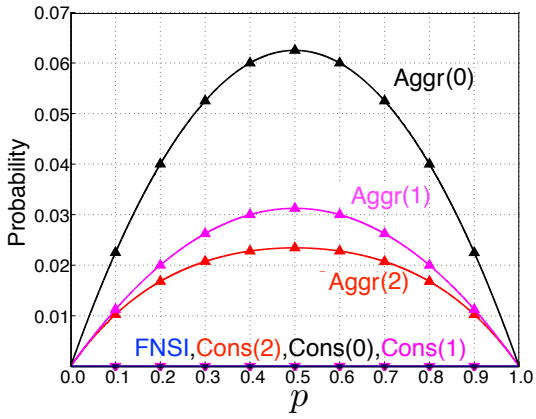
Figure 5.9 plots the expressions in Table 5.1 as functions of p. In Figure 5.9(a),



(a) Probability of assisting.



(b) Probability of causing backoff deferral.



(c) Probability of causing collision.

Figure 5.9: Each policy exhibits different behaviors in terms of the relay's propensity to transmit in the event subspaces.

we plot the probability of each scheme transmitting during the states where a relay is able to help its paired flow. The aggressive policies all improve performance over the full NSI^B policy since they transmit during cases where the full NSI^B policy halts relay transmission in accordance with minimizing negative impact on the neighboring flow. The conservative policies decrease performance over the full NSI^B policy since they avoid transmitting in states where the full NSI^B policy would – they are unable to distinguish these states from those where the relay should be halted. The $\mathsf{Cons}(1)$ policy in particular exhibits an unusual behavior in that it is able to help only as

$p \to 0$. This is due to the fact that, given only one hop of NSI^B knowledge, a relay is unable to align its transmissions to source interference that would be present anyway since it has no idea what the link qualities are between its source and other devices in the network.

In Figures 5.9(b) and 5.9(c), we see the impact of the relay policies on the probability of the neighboring flow deferring and colliding, respectively. As stated previously, the full NSI^B and conservative incomplete NSI^B policies cause no deferrals or collisions. The aggressive incomplete NSI policies, however, allow degradations in the interest of increasing the probability of assisting the cooperative flow.

5.4 Protocol Design and Simulation

The policies presented in Section 5.3 operate on binary network state information. Now, we translate the preceding two-flow policies into n-flow protocols. These protocols are then implemented in a custom network simulator and are evaluated in realistic fading environments.

5.4.1 Protocol Translation

The binary network model abstracts from reality in two key ways. First, only two unidirectional flows are considered, whereas arbitrary networks can potentially have many bidirectional flows. Second, channels take on only binary states whereas actual channels span a continuum of powers. We now translate the aforementioned policies into cooperative protocols that overcome these limitations of the model.

Specifically, we can directly translate the $\mathsf{Cons}(2)$, $\mathsf{Cons}(0)$, $\mathsf{Aggr}(2)$, and $\mathsf{Aggr}(0)$ policies. The one-hop policies, however, highlight a limitation in the binary network model when it applies to SINR-based protocol design. Consider the policy stated in

Equation (5.18). The relay transmits when its links to the other flow are disconnected. As defined by the binary collision model, the relay is able to guarantee that no collision or deferral event can take place in these states. This policy does not translate to an SINR-based scenario, where the measurement of a single link quality is insufficient to guarantee that a collision or deferral event will not take place. Even if the relay measured the instantaneous channel between itself and another destination as being very weak, it is still possible that a transmission will cause a collision if the channel supporting the other flow is also very weak. Despite this limitation, we are able to conclusively show that the remaining policies not only are capable of being translated into SINR-based protocols, but their *relative* performance in realistic fading environments is accurately predicted by our analysis of the binary model.

Conservative Protocols:

The Cons(2) and Cons(0) policies can be directly translated into protocols that operate on instantaneous SINR measurements. Consider a network $\mathcal{N}$ consisting of N devices.

Algorithm 1: Cons(2)

$\mathcal{N} = \{0, 1, 2, \ldots, N-1\}$
$X_{\mathsf{R}}^{\mathsf{Cons(2)}} = \mathsf{Tx}$
for $i \in \mathcal{N} \setminus \{\mathsf{S}, \mathsf{D}, \mathsf{R}\}$ **do**
  for $j \in \mathcal{N} \setminus \{i, \mathsf{S}, \mathsf{D}, \mathsf{R}\}$ **do**
    if *(*$\overline{\mathsf{BO}_i^{\mathsf{S}}}$ *and* $\mathsf{BO}_i^{\mathsf{SR}}$*) or (*$\overline{\mathsf{COL}_{ij}^{\mathsf{S}}}$ *and* $\mathsf{COL}_{ij}^{\mathsf{SR}}$*)* **then**
      $X_{\mathsf{R}}^{\mathsf{Cons(2)}} = \mathsf{Rx}$

Protocol 1 formally specifies the Cons(2) behavior. The collision and backoff terms

are

$$\mathsf{COL}_{ij}^{\mathsf{S}} = \left[\frac{P_t L_{ij}|h_{ij}|^2}{P_t L_{\mathsf{S}j}|h_{\mathsf{S}j}|^2 + z} < \gamma_{\mathsf{DET}}\right]$$

$$\mathsf{COL}_{ij}^{\mathsf{SR}} = \left[\frac{P_t L_{ij}|h_{ij}|^2}{P_t L_{\mathsf{S}j}|h_{\mathsf{S}j}|^2 + P_t L_{\mathsf{R}j}|h_{\mathsf{R}j}|^2 + \sum_{k\in\mathcal{I}_1} P_t L_{kj}|h_{kj}|^2 + z} < \gamma_{\mathsf{DET}}\right]$$

$$\mathsf{BO}_{i}^{\mathsf{S}} = \left[P_t L_{\mathsf{S}i}|h_{\mathsf{S}i}|^2 + z \geq \gamma_{\mathsf{CS}}\right]$$

$$\mathsf{BO}_{i}^{\mathsf{SR}} = \left[P_t L_{\mathsf{S}i}|h_{\mathsf{S}i}|^2 + P_t L_{\mathsf{R}i}|h_{\mathsf{R}i}|^2 + \sum_{k\in\mathcal{I}_2} P_t L_{ki}|h_{ki}|^2 + z \geq \gamma_{\mathsf{CS}}\right]$$

where $[\cdot]$ represents the Iverson bracket. Additionally, P_t is a constant representing transmission power, L_{ij} is a multiplicative factor that reduces power according to path loss between devices i and j, z is a constant representing the thermal noise power in each radio, γ_{DET} represents a threshold for the minimum SINR required to decode a reception, and γ_{CS} represents a power threshold for carrier-sensing. Finally, the $\mathcal{I}$ subsets represent other potential transmitters in the network (including other relays) as defined by

$$\mathcal{I}_1 = \mathcal{N} \setminus \{\mathsf{S}, \mathsf{D}, \mathsf{R}, i, j\}$$

$$\mathcal{I}_2 = \mathcal{N} \setminus \{\mathsf{S}, \mathsf{D}, \mathsf{R}, i\}.$$

The $\mathsf{Cons}(0)$ protocol can simply be stated as $X_{\mathsf{R}}^{\mathsf{Cons}(0)} = \mathsf{Rx}$ since the relay never transmits.

These protocols ensure that the relay is disabled whenever it would cause a deferral or collision event in surrounding flows. As such, they guarantee zero reduction in spatial reuse.

Aggressive Protocols:

Similarly, the $\mathsf{Aggr}(2)$ and $\mathsf{Aggr}(0)$ policies can be directly translated into protocols that operate on instantaneous SINR measurements.

Algorithm 2: $\mathsf{Aggr}(2)$

$\mathcal{N} = \{0, 1, 2, \ldots, N-1\}$
$X_{\mathsf{R}}^{\mathsf{Aggr}(2)} = \mathsf{Tx}$
for $i \in \mathcal{N} \setminus \{\mathsf{S}, \mathsf{D}, \mathsf{R}\}$ **do**
  for $j \in \mathcal{N} \setminus \{i, \mathsf{S}, \mathsf{D}, \mathsf{R}\}$ **do**
    if $(\overline{\mathsf{BO}_i^{\mathsf{S}}}$ *and* $\mathsf{BO}_i^{\mathsf{SR}}$ *and* $\overline{\mathsf{COL}_{ij}^{\mathsf{S}}}$ *and* $\mathsf{COL}_{ij}^{\mathsf{SR}})$ *or* $\mathsf{COL}_{\mathsf{SRD}}$ **then**
      $X_{\mathsf{R}}^{\mathsf{Aggr}(2)} = \mathsf{Rx}$

Protocol 2 formally specifies the $\mathsf{Aggr}(2)$ behavior. The collision and backoff terms are

$$\mathsf{COL}_{\mathsf{SRD}} = \left[\frac{P_t L_{\mathsf{SD}} |h_{\mathsf{SD}}|^2 + P_t L_{\mathsf{RD}} |h_{\mathsf{RD}}|^2}{z} < \gamma_{\mathsf{DET}}\right]$$
$$\mathsf{COL}_{ij}^{\mathsf{S}} = \left[\frac{P_t L_{ij} |h_{ij}|^2}{P_t L_{\mathsf{S}j} |h_{\mathsf{S}j}|^2 + z} < \gamma_{\mathsf{DET}}\right]$$
$$\mathsf{COL}_{ij}^{\mathsf{SR}} = \left[\frac{P_t L_{ij} |h_{ij}|^2}{P_t L_{\mathsf{S}j} |h_{\mathsf{S}j}|^2 + P_t L_{\mathsf{R}j} |h_{\mathsf{R}j}|^2 + z} < \gamma_{\mathsf{DET}}\right]$$
$$\mathsf{BO}_i^{\mathsf{S}} = \left[P_t L_{\mathsf{S}i} |h_{\mathsf{S}i}|^2 + z \geq \gamma_{\mathsf{CS}}\right]$$
$$\mathsf{BO}_i^{\mathsf{S,R}} = \left[P_t L_{\mathsf{S}i} |h_{\mathsf{S}i}|^2 + P_t L_{\mathsf{R}i} |h_{\mathsf{R}i}|^2 + z \geq \gamma_{\mathsf{CS}}\right]$$

where all components share the same definitions. The $\mathsf{Aggr}(0)$ protocol can simply be stated as $X_{\mathsf{R}}^{\mathsf{Aggr}(0)} = \mathsf{Tx}$ since it makes no effort to defer any of its transmissions.

The aggressive protocols will increase the rate of the cooperative flow but will do so at some cost to surrounding flows.

5.4.2 Performance Evaluation

In Table 5.2, we specify the key simulation parameters. All other parameters in the experiment including SINR thresholds for packet decoding are identical to the default values specified in [52].

In Section 5.3, we evaluated the various policies with a single parameter p that

Table 5.2: Simulation Parameters

Header Rate	BPSK (1/2 rate code)
Payload Rate	16-QAM (3/4 rate code)
Path loss Exponent	3
Fading	Correlated Rayleigh
Doppler Freq. (f_d)	15Hz
RTS/CTS	Disabled
Traffic	CBR
Packet Size	1470 bytes

affects the likelihood of the cooperative flow being connected to the other flow in the network. In this section, the analogous parameter is the distance between the relayed and non-relayed flow as shown in the simulation topology in Figure 5.10.

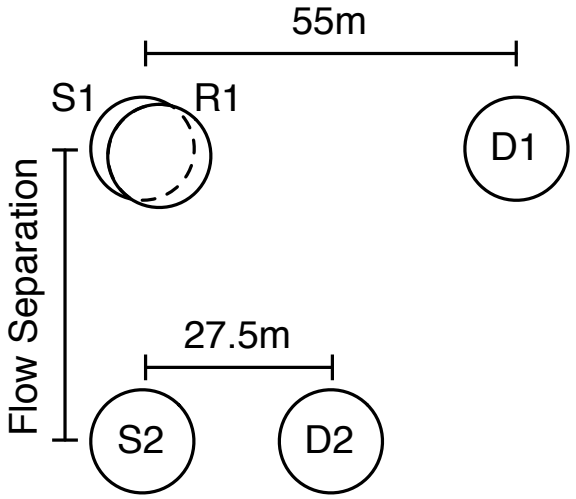


Figure 5.10: We vary the flow separation distance as the independent variable for the simulation.

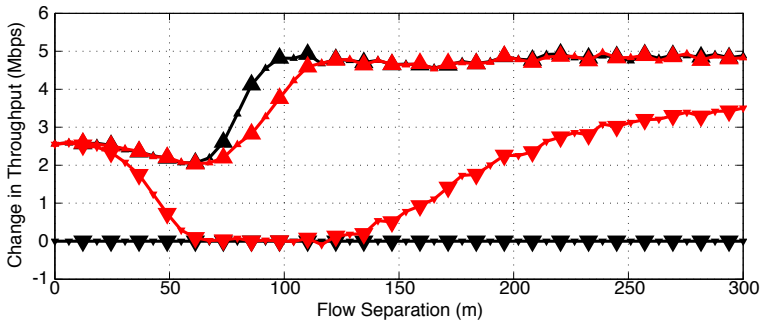
The cooperative flow has a large source-destination distance in order to place that flow in a fading-dominated regime (i.e. a significant number of the packet losses suffered by the destination are due to inadequate channel capacity between S1 and D1). The non-cooperative flow is in an interference-dominated regime where very few of its transmissions are lost due to fades. This topological selection emphasizes the

negative impact of the relay on the non-cooperative flow and allows clear differences between the distributed cooperative protocols to be seen.

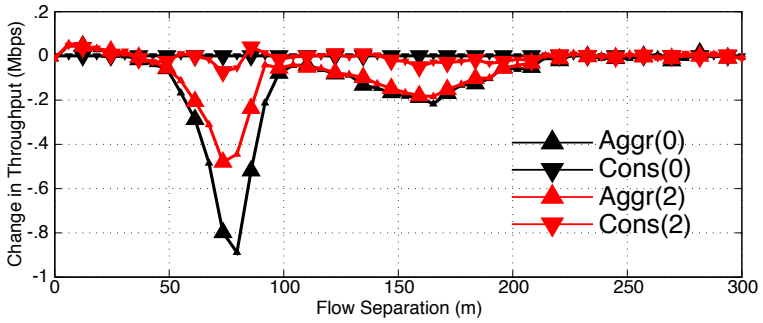
A useful metric for evaluating the performance of a protocol is to compare the throughput of each flow when a relay is present in the network with the throughput of each flow when there is no relay. We consider this change in throughput.

Figure 5.11 shows the measured throughput difference when a relay is present and when it is not for each previously described protocol. Figure 5.11(a) focuses on the impact of cooperation on the cooperative flow. Of note, the Cons(0) protocol provides no improvement over the case where the relay is absent from the network because the Cons(0) protocol never uses the relay. All other protocols, however, provide throughput improvement. In particular, the Aggr(0) protocol (that always uses the relay) provides the most improvement at all flow separation distances. This fact is predicted by the binary network model and was shown in Figure 5.9(a).

Figure 5.11(b) shows the impact of relaying on the non-cooperative flow. Since the relay can only increase the footprint of the cooperative flow, this means that spatial reuse can only degrade and not improve the performance of the non-cooperative flow. Again, the Cons(0) protocol never uses the relay so it never degrades the throughput of the non-cooperative flow. Aggr(0), however, has two distinct regions where the harm on the non-cooperative flow reaches local maxima. The reason there are two regions is that the locations where collisions and backoff deferrals each create the most impact are not necessarily the same; they depend on the many parameters specific to the simulation. Regardless, the Cons(2) protocol avoids *any* harm just as the corresponding policy predicts in Section 5.3. In Figure 5.9, each policy was analyzed as a function of a proxy for flow separation. Noting the similarities with the actual flow separation comparisons in Figure 5.11, this confirms the binary model as a robust mechanism for the procedural generation of cooperative protocols.



(a) Cooperative Flow Performance.



(b) Non-cooperative Flow Performance.

Figure 5.11: The relay can assist the flow with which it is paired and harm the flow with which it is not. These effects can be balanced with protocol selection.

5.5 Discussion

Signal-scale cooperation shows tremendous potential for performance improvement in wireless links that are able to use cooperative relays. However, for other links in the network, cooperation is a threat to their performance due to the loss of spatial reuse caused by additional transmitters in the shared wireless medium. In this chapter, we have presented a policy design methodology that allows the systematic study of relay behavior for arbitrary amounts of network knowledge at the relay. Through extensive network simulations, we demonstrate the successful application of this method to distributed protocols that operate in fading environments.

Chapter 6

Conclusion

Signal-scale cooperation provides great opportunity for overcoming the impairments of fading channels. At the same time, it presents a danger to devices in the form of degraded energy efficiency and inter-device interference. In this thesis, we have presented MAC-level solutions that allow devices to intelligently choose when and how to cooperate in order to overcome these losses.

First, we have presented two novel, completely distributed MAC layers – DOC and PDOC. These MACs answer the challenge of distributed coordination through the use of NACKs to trigger cooperation. They also ensure that cooperation is only used when it can be helpful. Our next contribution, DECC, allows devices to become self-aware of their own cooperative performance – DECC scales the degree to which devices cooperate with one another such that the worst-case harm to energy efficiency is mitigated to a bounded amount. Finally, we have presented a technique for the generation of new MAC layer behavior to combat inter-device interference given different degrees of network awareness. In sum, these contributions provide mechanisms by which devices can tailor their cooperative efforts based on the effects of cooperation on *all* parties: the source/destination of traffic needing cooperation, the relay itself, and other unrelated devices in the network.

Future Work:

The protocols presented in this thesis and their implementations can be used in the study of new problems related to signal-scale cooperation. In this section, we highlight two example research problems, discuss their challenges, and offer insights into their possible solutions.

Automatic Rate Adaptation:

One area for extension of our work is the field of adaptive rate communication. In a wireless network that undergoes channel fluctuations, the MAC layer can adaptively change the modulation and coding rate of the PHY such that a reasonable packet error rate (PER) is maintained. Techniques to realize this behavior range from sender-based PER heuristics [72] to receiver-based per-packet SNR measurements [73]. In many ways, the goals of rate adaptation are aligned with those of signal-scale cooperation: improving link reliability through avoiding channel outages. Rate adaptation and signal-scale cooperation are not mutually exclusive, but their interactions are subtle and not well-understood.

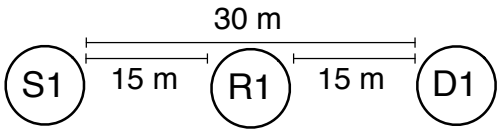


Figure 6.1: Single-flow linear topology with an altruistic relay placed equidistant from its source and its destination.

As an example, consider the topology shown in Figure 6.1. S1 is a fixed distance away from D1 that is large enough such that direct communication at the fastest coding rates is very unreliable. Through simulation, we consider the role of signal-scale cooperation with R1 as well as rate adaptation at S1. Specifically, we consider a very simple rate adaptation scheme known as Automatic Rate Fallback (ARF) [72].

Briefly summarizing, ARF simply increases the rate of transmitted packets after a certain number of successful (i.e. ACKed) transmissions. ARF decreases the rate of transmitted packets after a certain number of sequential failed transmissions. For this simulation, we have chosen both of these thresholds to be 10 packets, but this parameter can be changed to alter how aggressively ARF adjusts its rate.

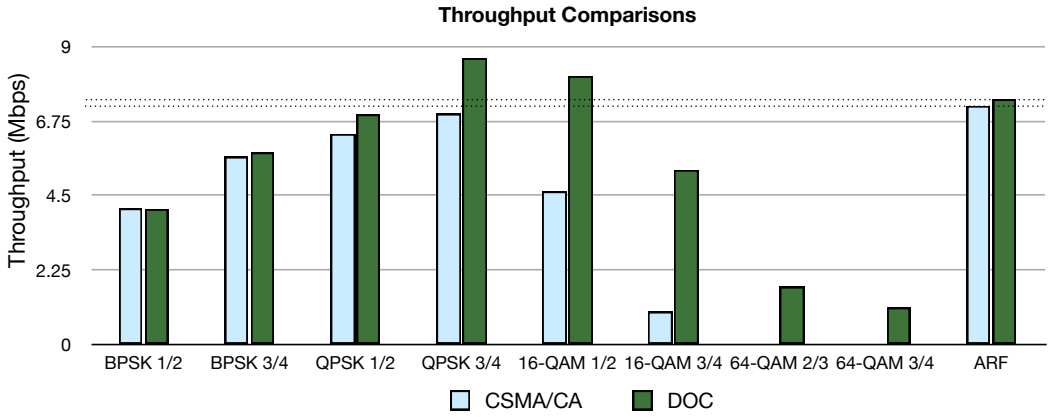


Figure 6.2: Cooperation out-performs direct communication at all fixed rates, but realizing cooperative gain with an adaptive rate protocol is an open question.

Figure 6.2 shows the achieved throughput of the S1 to D1 link for a variety of fixed rates as well as ARF. We can see that for every single fixed-rate scheme, DOC never performs worse than a non-cooperative CSMA/CA implementation and often does much better. With ARF, the non-cooperative CSMA/CA performs better than any non-cooperative fixed-rate scheme. Intuitively, fixed-rate schemes are prone to (a) overly cautious rates when instantaneous SNRs are high and (b) overly aggressive rates when instantaneous SNRs are low. Rate adaptation helps avoid these problems by allowing rate to track changes in channel qualities. With DOC, however, ARF performs *worse* than both fixed-rate QPSK with a 3/4 rate convolutional code and 16-QAM with a 1/2 rate convolutional code. ARF fails to harness the rate gains available with cooperation. ARF adjusts its rate such that direct S1-D1 communication is very reliable *even when* it may be faster overall to fail on the direct link and succeed with

cooperative diversity at a higher rate. Simply put, ARF was not designed to take advantage of cooperative resources.

The problem of adaptive rate signal-scale cooperation is very open. If a source knows that a relay is available, it may opt to send at a very aggressive coding rate. If, however, that relay is not available or otherwise refuses to assist, then the source may have been better off transmitting directly to the destination at a more conservative coding rate.

Furthermore, in scenarios where more than one relay may transmit at once, care must be taken such that they all have selected the same coding rate of their transmission. Otherwise, the signals will collide and nullify the gains cooperation can provide. One solution to this problem may have a similar structure to the solution to a relay selection problem presented in [74]. Relays that have selected the fastest possible PHY rate can transmit in concert with one another immediately after the reception of the NACK. Relays that have selected the next-fastest PHY rate carrier sense the medium for a short period and cancel their transmissions if they hear anything. If they do not, then this collection of relays can transmit in concert with one another at the slower PHY rate after the carrier sensing interval following reception of the NACK. This will help to ensure that simultaneous relay transmissions are always at the same selected coding rate.

Cooperative Ad Hoc Routing:

In ad hoc networks, a traffic source and its destination may be very geographically separate. As such, the flow of traffic may be routed through a series of direct links before it reaches its destination. There is a rich body of literature on routing techniques (e.g. [75, 76]). Signal-scale cooperation has the potential to improve the quality of various links along a multi-hop flow and, as such, should be considered in

the route selection process.

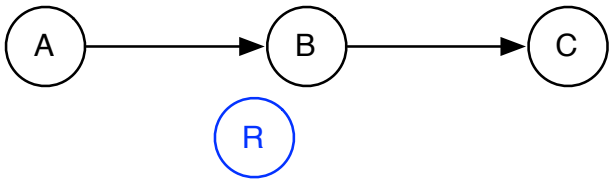


Figure 6.3: Multi-hop topology where A is routed to C through B.

Consider the topology shown in Figure 6.3. We assume that a routing protocol has established that A should transmit through B to reach C. Furthermore, we assume the presence of another device R outside of the route that is willing to aid the communication of its neighbors via cooperative transmissions.

In all of the protocols presented in this thesis, cooperation is a purely reactive measure to recover from losses due to channel fades. In Chapter 3, DOC was designed for explicitly this purpose. Multi-hop networks provide an excellent opportunity for cooperation to transition from reactively patching errors caused by fading events to also *proactively* making sure errors do not occur.

In Figure 6.4, we have illustrated how this transition to proactive cooperation can occur. In Hop 1 (A→B), suppose a channel fade makes the direct link between A to B undecodable. With a protocol such as DOC or PDOC, the relay R may react to this error and successfully communicate the packet to B via the diversity benefits seen at B through the two transmission channels. On the second hop (B→C), the relay R recognizes that it already knows the message being sent from the first hop. After updating header fields in the transmission to mimic the second hop, the relay R can immediately transmit in concert with B – the second hop can engage cooperative resource proactively without waiting for a failure to occur.

There is already a body of work surrounding this cooperative multi-hop network problem. One particularly promising example is the Barrage Relay Network (BRN) [77]. BRNs focus on multicast applications such as emergency responder and

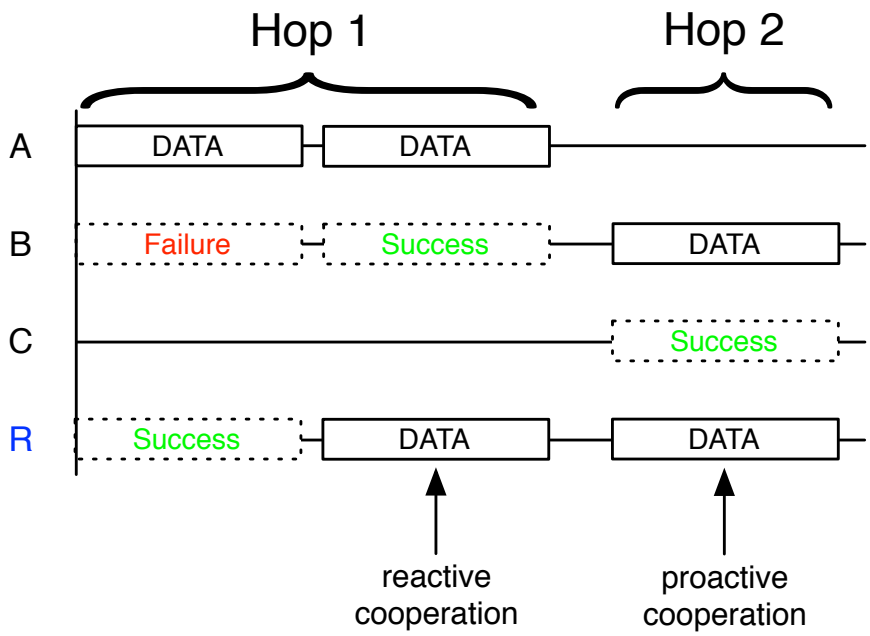


Figure 6.4: Cooperative relays can both assist reactively and proactively in a multi-hop network.

military communications in harsh fading environments. Communication occurs in stages. First, the device originating the message broadcasts it to all neighbors. Next, all neighbors that heard this message simultaneously forward the message to all of *their* neighbors through a combination of random phase dithering and forward error correction (FEC) codes to avoid collisions. Next, all of those devices forward to their neighbors and so on. Messages spread like wildfire through the network and can very reliably reach devices on the outskirts of networks in very harsh fading environments.

Fundamentally, the cooperative routing problem is related to the rate control problem described earlier. If cooperative resources exist, devices may want to choose fewer-hop routes where individual links are long rather than more conservative larger-hop routes where individual links are short.

Appendix A

Collision Detection

A received waveform failing to be decoded properly can be attributed to either channel fading or collision. In this appendix, we describe a simple, novel mechanism that allows receivers to make this determination. In [78], collision detection was solved using a correlation approach where receivers continue to search for the autocorrelation and cross-correlation of other packet preambles. The disadvantage of such an approach is that collisions can be detected only from other transmitters employing the known preamble structure. As an example, the 2.4 GHz ISM band is a piece of spectrum where many different wireless technologies transmit and receive. Collisions across these technologies are very difficult to detect using this approach.

Instead, we propose a collision detection system based on a very simple metric to calculate within the physical layer: error vector magnitude (EVM). EVM is a direct measurement of SINR and can be formally defined as

$$\mathsf{EVM} = |I_{\mathsf{Rx}} - I_{\mathsf{Tx}}|^2 + |Q_{\mathsf{Rx}} - Q_{\mathsf{Tx}}|^2, \tag{A.1}$$

where I and Q are the in-phase and quadrature portions of a symbol respectively.

Note that in order to calculate EVM, the transmitted symbols must be known. As such, EVM cannot be calculated on data payloads. However, EVM can be calculated on the known pilot symbols present in transmissions.

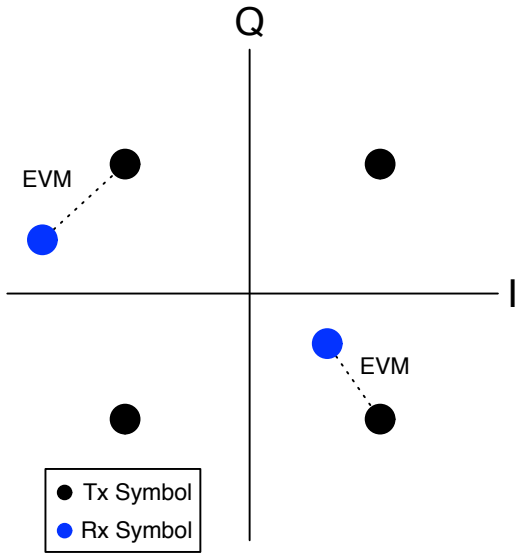
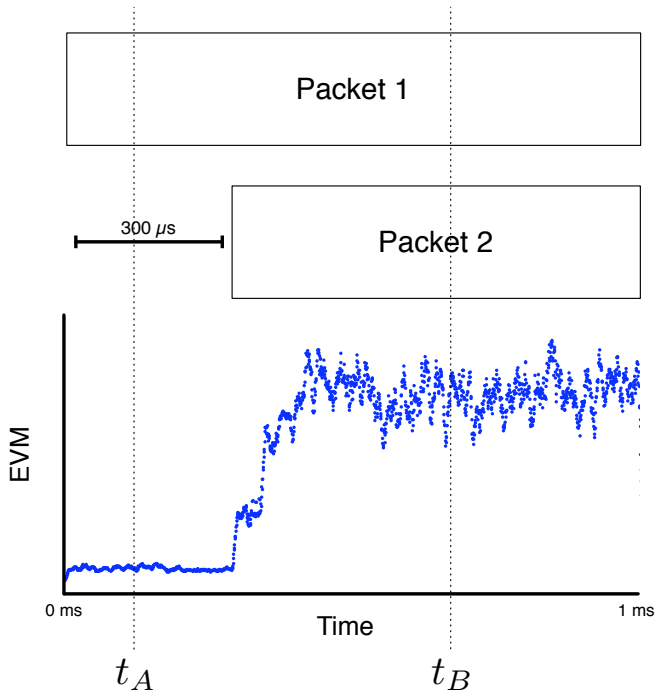


Figure A.1: EVM is a direct measurement of SNR. It captures the size of the noise cloud around valid symbols in a constellation.

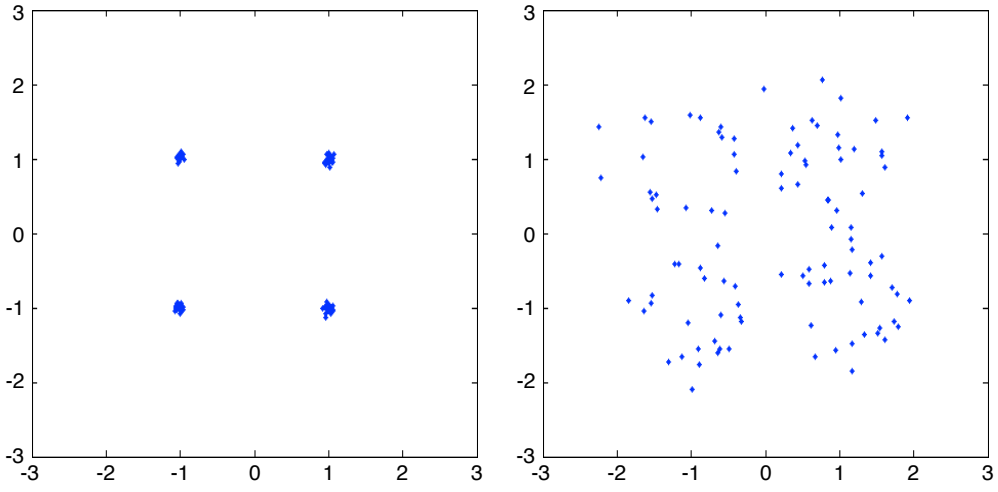
Both fading events and collisions decrease SINR and therefore increase EVM. However, collisions exhibit a property in the *change* in EVM over the course of the packet that fading events generally do not. As an example, we have constructed an experiment using the WARPLab physical layer prototyping framework [79]. In this experiment, we trigger a collision by beginning a transmission of one WARP node 300 μs after the start of another transmission from a different WARP node. On a third WARP node, we then measure the EVM of the received symbols over the course of the packet.

Figure A.2 shows the data from this experiment. When Packet 2 begins, the EVM of the receiver attempting to decode Packet 1 increases alongside the sudden addition of interference. The constellations at times t_A and t_B are plotted in Figure A.2.

Collisions can be detected by watching for sudden changes in EVM. Fading pro-



(a) Timeline



(b) t_A (c) t_B

Figure A.2: Sudden increases in EVM are indicative of collisions.

duces much slower changes in EVM. In fact, fading is assumed to be static over the course of a packet in order for preamble-based channel training to be effective. While fading may produce large or small baseline EVM values, the presence of a collision is recognizable by a sudden deviation away from that baseline.

Appendix B

MRC Emulation Calibration

The cooperative PHY from [10] employs a 2×1 Alamouti STBC to achieve spatial diversity. For architectural simplification, the PHY ignores the first "broadcast" time slot of cooperation and requires the source to retransmit its message alongside the relay during the second time slot. Equivalently, the PHY could combine the source's transmission from the first broadcast phase with a relay-only transmission during the relay phase via a technique known as maximal-ratio combining (MRC) and save the source from having to retransmit again. Adding MRC capability to the PHY would require fundamental re-architecting beyond the scope of this work. Instead, we use the 2×1 Alamouti mode of the PHY to characterize the performance of an equivalent MRC-capable physical layer. We then use these calibration values to emulate its performance on the current PHY with relay-only cooperative transmissions.

Figure B.1 shows the packet error rate (PER) of the destination node as a function of the powers of the source-destination and relay-destination links. The PER characterization was created using the same 16-QAM modulation used throughout the rest of this work. Note, the experiment to generate this data is precisely the same as the experiment that generated the PER data for the PDOC implementation in Figure 3.12. Figure B.1 has been included in this appendix only as reference; the

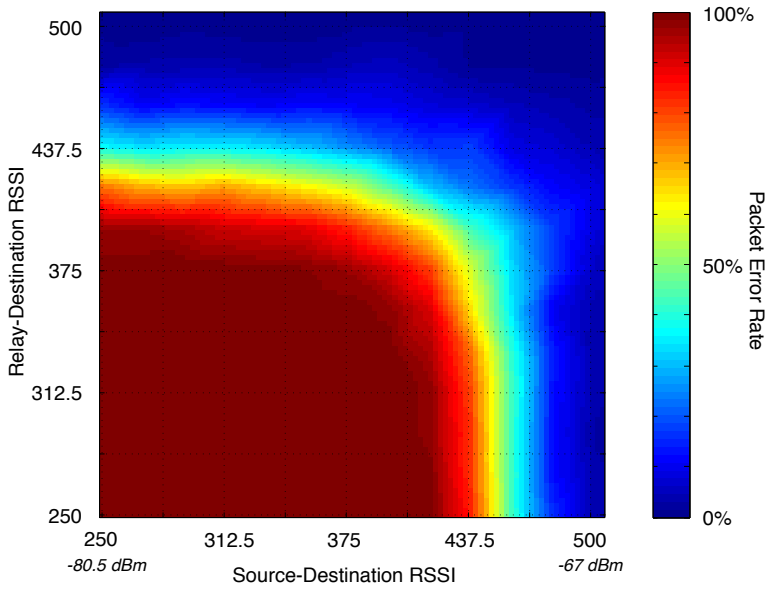


Figure B.1: PER of 2×1 Alamouti PHY.

underlying data differs slightly from the earlier experiment as different WARP nodes were tested.

In our emulated MRC, when a device sends a NACK packet it saves the received signal strength (RSSI) of the data packet it just received. Then, after receiving the RSSI of a relay-only transmission, the destination draws a random number according to a Bernoulli distribution whose parameter is derived from Figure B.1. If that random draw is 1, the packet is discarded. If it is 0, the packet is deemed successful. In this way, we encode the physical layer with the performance of an MRC mode without making the physical layer actually perform the task.

Appendix C

Radio Power Consumption

We used the WARP Radio Board v1.4 for all experiments in this thesis. There are four main components that are responsible for the bulk of the radio board's power consumption:

Digital-to-Analog Converter (DAC) and Analog-to-Digital (ADC): are responsible for converting baseband digital I and Q samples from the FPGA into analog waveforms and vice versa. Because of latencies involved with waking these devices from sleep, current WARP designs leave both the DAC and the ADC running at all times.

Transceiver (TRA): is responsible for upconverting to RF and downconverting to baseband during transmission and reception, respectively. Additionally, the transceiver contains a transmission variable gain amplifier (VGA) that covers a 32 dB range. The power draw from the transceiver depends on the output power of this VGA.

Power Amplifier (PA): The radio board has a power amplifier (PA) after the transceiver for boosting transmissions by another 32 dB. The power consumption of the part is a function of the power of the input signal from the transceiver. During reception,

the PA is deactivated and its current draw is negligible.

Table C.1: WARP Radio Power Consumption

	Tx	Rx
DAC [80]	.418W	.418W
ADC [81]	.713W	.713W
Transceiver [82]	Equation (C.1)	.354W
PA[1]	Equation (C.2)	-

Table C.1 summarizes the power consumption of the primary components on the WARP Radio Board. For the two parts that depend on transmission power (the transceiver and the PA), we have used their respective data sheets to perform a simple exponential curve fit. Their expressions are

$$P_D^{\mathsf{PA}} = .0034e^{.2072P_T} + .2162e^{.0275P_T} \tag{C.1}$$

$$P_D^{\mathsf{TRA}} = .2482e^{-.0084P_T} + .0922e^{.0264P_T}, \tag{C.2}$$

where P_T is the transmission output power (in dBm) and P_D^{PA}, P_D^{TRA} are the power dissipated by the PA and transceiver (in Watts), respectively.

Figure C.1 shows a comparison of reception and transmission power consumption.

In our analysis, we consider only the power consumption of the radio board since it would likely be the dominant energy sink in a final implementation that has been reduced to custom silicon. That said, our approach is general and can easily be extended to consider baseband processing costs as well as the energy costs of higher layer components such as the transport layer and applications. Updating DECC for additional power measurements would only require updating the offline power

[1]The WARP Radio Board uses a Sharp IRM046U7 power amplifier. Since this part does not provide supply current specifications as a function of transmission power, we have opted to substitute an equivalent power amplifier for this analysis [83].

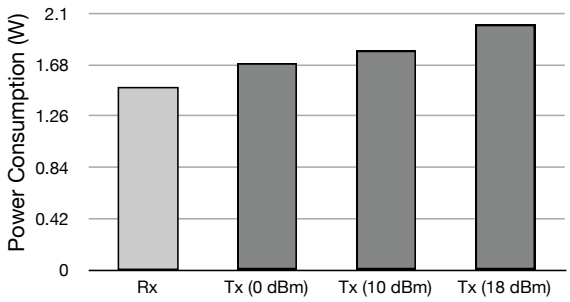


Figure C.1: Power consumption comparison of reception and transmission at a variety of output powers.

calculations. The architecture of the DECC implementation would remain unchanged.

Appendix D

Network State Classification

ψ^B	Label	ψ^B	Label	ψ^B	Label	ψ^B	Label	ψ^B	Label	ψ^B	Label	ψ^B	Label	ψ^B	Label
0	$\mathcal{D}$	32	$\mathcal{D}$	64	$\mathcal{D}$	96	$\mathcal{D}$	128	$\mathcal{D}$	160	$\mathcal{D}$	192	$\mathcal{D}$	224	$\mathcal{D}$
1	$\mathcal{D}$	33	$\mathcal{D}$	65	$\mathcal{D}$	97	$\mathcal{D}$	129	$\mathcal{D}$	161	$\mathcal{D}$	193	$\mathcal{D}$	225	$\mathcal{D}$
2	$\mathcal{A}$	34	$\mathcal{A}$	66	$\mathcal{A}$	98	$\mathcal{A}$	130	$\mathcal{A}$	162	$\mathcal{A}$	194	$\mathcal{A}$	226	$\mathcal{A}$
3	$\mathcal{A}$	35	$\mathcal{A}$	67	$\mathcal{A}$	99	$\mathcal{A}$	131	$\mathcal{A}$	163	$\mathcal{A}$	195	$\mathcal{A}$	227	$\mathcal{A}$
4	$\mathcal{B}$	36	$\mathcal{D}$	68	$\mathcal{B}$	100	$\mathcal{D}$	132	$\mathcal{D}$	164	$\mathcal{D}$	196	$\mathcal{D}$	228	$\mathcal{D}$
5	$\mathcal{B}$	37	$\mathcal{D}$	69	$\mathcal{B}$	101	$\mathcal{D}$	133	$\mathcal{D}$	165	$\mathcal{D}$	197	$\mathcal{D}$	229	$\mathcal{D}$
6	$\mathcal{A} \cap \mathcal{B}$	38	$\mathcal{A}$	70	$\mathcal{A} \cap \mathcal{B}$	102	$\mathcal{A}$	134	$\mathcal{A}$	166	$\mathcal{A}$	198	$\mathcal{A}$	230	$\mathcal{A}$
7	$\mathcal{A} \cap \mathcal{B}$	39	$\mathcal{A}$	71	$\mathcal{A} \cap \mathcal{B}$	103	$\mathcal{A}$	135	$\mathcal{A}$	167	$\mathcal{A}$	199	$\mathcal{A}$	231	$\mathcal{A}$
8	$\mathcal{D}$	40	$\mathcal{D}$	72	$\mathcal{D}$	104	$\mathcal{D}$	136	$\mathcal{D}$	168	$\mathcal{D}$	200	$\mathcal{D}$	232	$\mathcal{D}$
9	$\mathcal{D}$	41	$\mathcal{D}$	73	$\mathcal{D}$	105	$\mathcal{D}$	137	$\mathcal{D}$	169	$\mathcal{D}$	201	$\mathcal{D}$	233	$\mathcal{D}$
10	$\mathcal{A}$	42	$\mathcal{A}$	74	$\mathcal{A}$	106	$\mathcal{A}$	138	$\mathcal{D}$	170	$\mathcal{D}$	202	$\mathcal{D}$	234	$\mathcal{D}$
11	$\mathcal{A}$	43	$\mathcal{A}$	75	$\mathcal{A}$	107	$\mathcal{A}$	139	$\mathcal{D}$	171	$\mathcal{D}$	203	$\mathcal{D}$	235	$\mathcal{D}$
12	$\mathcal{B}$	44	$\mathcal{D}$	76	$\mathcal{B}$	108	$\mathcal{D}$	140	$\mathcal{D}$	172	$\mathcal{D}$	204	$\mathcal{D}$	236	$\mathcal{D}$
13	$\mathcal{B}$	45	$\mathcal{D}$	77	$\mathcal{B}$	109	$\mathcal{D}$	141	$\mathcal{D}$	173	$\mathcal{D}$	205	$\mathcal{D}$	237	$\mathcal{D}$
14	$\mathcal{A} \cap \mathcal{B}$	46	$\mathcal{A}$	78	$\mathcal{A} \cap \mathcal{B}$	110	$\mathcal{A}$	142	$\mathcal{D}$	174	$\mathcal{D}$	206	$\mathcal{D}$	238	$\mathcal{D}$
15	$\mathcal{A} \cap \mathcal{B}$	47	$\mathcal{A}$	79	$\mathcal{A} \cap \mathcal{B}$	111	$\mathcal{A}$	143	$\mathcal{D}$	175	$\mathcal{D}$	207	$\mathcal{D}$	239	$\mathcal{D}$
16	$\mathcal{D}$	48	$\mathcal{D}$	80	$\mathcal{D}$	112	$\mathcal{D}$	144	$\mathcal{D}$	176	$\mathcal{D}$	208	$\mathcal{D}$	240	$\mathcal{D}$
17	$\mathcal{D}$	49	$\mathcal{D}$	81	$\mathcal{D}$	113	$\mathcal{D}$	145	$\mathcal{C}$	177	$\mathcal{C}$	209	$\mathcal{D}$	241	$\mathcal{D}$
18	$\mathcal{A}$	50	$\mathcal{A}$	82	$\mathcal{A}$	114	$\mathcal{A}$	146	$\mathcal{A}$	178	$\mathcal{A}$	210	$\mathcal{A}$	242	$\mathcal{A}$
19	$\mathcal{A}$	51	$\mathcal{A}$	83	$\mathcal{A}$	115	$\mathcal{A}$	147	$\mathcal{A} \cap \mathcal{C}$	179	$\mathcal{A} \cap \mathcal{C}$	211	$\mathcal{A}$	243	$\mathcal{A}$
20	$\mathcal{B}$	52	$\mathcal{D}$	84	$\mathcal{B}$	116	$\mathcal{D}$	148	$\mathcal{D}$	180	$\mathcal{D}$	212	$\mathcal{D}$	244	$\mathcal{D}$
21	$\mathcal{B}$	53	$\mathcal{D}$	85	$\mathcal{B}$	117	$\mathcal{D}$	149	$\mathcal{C}$	181	$\mathcal{C}$	213	$\mathcal{D}$	245	$\mathcal{D}$
22	$\mathcal{A} \cap \mathcal{B}$	54	$\mathcal{A}$	86	$\mathcal{A} \cap \mathcal{B}$	118	$\mathcal{A}$	150	$\mathcal{A}$	182	$\mathcal{A}$	214	$\mathcal{A}$	246	$\mathcal{A}$
23	$\mathcal{A} \cap \mathcal{B}$	55	$\mathcal{A}$	87	$\mathcal{A} \cap \mathcal{B}$	119	$\mathcal{A}$	151	$\mathcal{A} \cap \mathcal{C}$	183	$\mathcal{A} \cap \mathcal{C}$	215	$\mathcal{A}$	247	$\mathcal{A}$
24	$\mathcal{D}$	56	$\mathcal{D}$	88	$\mathcal{D}$	120	$\mathcal{D}$	152	$\mathcal{D}$	184	$\mathcal{D}$	216	$\mathcal{D}$	248	$\mathcal{D}$
25	$\mathcal{D}$	57	$\mathcal{D}$	89	$\mathcal{D}$	121	$\mathcal{D}$	153	$\mathcal{C}$	185	$\mathcal{C}$	217	$\mathcal{D}$	249	$\mathcal{D}$
26	$\mathcal{A}$	58	$\mathcal{A}$	90	$\mathcal{A}$	122	$\mathcal{A}$	154	$\mathcal{D}$	186	$\mathcal{D}$	218	$\mathcal{D}$	250	$\mathcal{D}$
27	$\mathcal{A}$	59	$\mathcal{A}$	91	$\mathcal{A}$	123	$\mathcal{A}$	155	$\mathcal{C}$	187	$\mathcal{C}$	219	$\mathcal{D}$	251	$\mathcal{D}$
28	$\mathcal{B}$	60	$\mathcal{D}$	92	$\mathcal{B}$	124	$\mathcal{D}$	156	$\mathcal{D}$	188	$\mathcal{D}$	220	$\mathcal{D}$	252	$\mathcal{D}$
29	$\mathcal{B}$	61	$\mathcal{D}$	93	$\mathcal{B}$	125	$\mathcal{D}$	157	$\mathcal{C}$	189	$\mathcal{C}$	221	$\mathcal{D}$	253	$\mathcal{D}$
30	$\mathcal{A} \cap \mathcal{B}$	62	$\mathcal{A}$	94	$\mathcal{A} \cap \mathcal{B}$	126	$\mathcal{A}$	158	$\mathcal{D}$	190	$\mathcal{D}$	222	$\mathcal{D}$	254	$\mathcal{D}$
31	$\mathcal{A} \cap \mathcal{B}$	63	$\mathcal{A}$	95	$\mathcal{A} \cap \mathcal{B}$	127	$\mathcal{A}$	159	$\mathcal{C}$	191	$\mathcal{C}$	223	$\mathcal{D}$	255	$\mathcal{D}$

Recalling that H_{ij}^B and X_i are binary valued, let

$$\psi^B = H^B_{\mathsf{R1D2}} \cdot 2^0 + H^B_{\mathsf{R1D1}} \cdot 2^1 + H^B_{\mathsf{R1S2}} \cdot 2^2 + H^B_{\mathsf{S2D1}} \cdot 2^3 +$$
$$H^B_{\mathsf{S2D2}} \cdot 2^4 + H^B_{\mathsf{S1S2}} \cdot 2^5 + H^B_{\mathsf{S1D2}} \cdot 2^6 + X_{\mathsf{S2}} \cdot 2^7.$$

Each network state is labelled in the preceding table.

References

[1] D. Goldman, "Sorry, America: Your wireless airwaves are full." [Online]. Available: http://money.cnn.com/2012/02/21/technology/spectrum_crunch/?npt=NP1&hpt=hp_c2 1

[2] J. Laneman, D. Tse, and G. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, 2004. 1, 2.1, 3.2.2, 5.1, 5.2.4

[3] G. Kramer, I. Marić, and R. Yates, "Cooperative Communications," *Foundations and Trends in Networking*, vol. 1, no. 3, 2006. 1, 1, 2.1, 5.1

[4] P. Murphy, A. Sabharwal, and B. Aazhang, "On Building a Cooperative Communication System: Testbed Implementation and First Results," *EURASIP Journal on Wireless Communications and Networking*, 2009. 1, 2.2.1, 3.1.1

[5] G. Bradford, "A Framework For Implementation and Evaluation of Cooperative Diversity in Software-Defined Radio," Master's thesis, University of Notre Dame, 2008. 1, 3.1.1

[6] G. Bradford and J. Laneman, "An experimental framework for the evaluation of cooperative diversity," in *Proceedings of CISS*, 2009. 1, 3.1.1

[7] A. Bletsas and A. Lippman, "Implementing Cooperative Diversity Antenna Arrays with Commodity Hardware," *IEEE Communications Magazine*, vol. 44, no. 12, 2006. 1, 3.1.1

[8] T. Korakis, M. Knox, E. Erkip, and S. Panwar, "Cooperative Network Implementation Using Open-Source Platforms," *IEEE Communications Magazine*, vol. 47, no. 2, 2009. 1, 3.1.1

[9] P. Zetterberg, C. Mavrokefalidis, A. Lalos, and E. Matigakis, "Experimental Investigation of Cooperative Schemes on a Real-Time DSP-based Testbed," *EURASIP Journal on Wireless Communications and Networking*, 2009. 1

[10] P. Murphy and A. Sabharwal, "Design, Implementation, and Characterization of a Cooperative Communications System," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 6, 2011. 1, 2.1, 2.2.1, 3.1.1, B

[11] C. Hunter, P. Murphy, and A. Sabharwal, "Real-time testbed implementation of a distributed cooperative MAC and PHY," in *Proceedings of CISS*, 2010. 1.1, 3.1, 3.2.2, 5.2.4

[12] C. Hunter, M. Kanga, L. Zhong, and A. Sabharwal, "On-demand Cooperation with Power Control: Protocol and Experimental Results," in *Proceedings of IEEE Asilomar Conference on Signals, Systems, and Computers*, 2011. 1.1, 3.2, 5.2.4

[13] C. Hunter and A. Sabharwal, "Distributed Protocols for Interference Management in Cooperative Networks," *IEEE Journal on Selected Areas in Communications, to appear*, 2012. 1.1, 5

[14] E. Van der Meulen, "Transmission of Information in a T-terminal Discrete Memoryless Channel," Ph.D. dissertation, University of California at Berkeley, 1969. 2.1

[15] ——, "Three-terminal communication channels," *Advanced Applied Probability*, vol. 3, no. 1, 1971. 2.1

[16] T. Cover and A. Gamal, "Capacity Theorems for the Relay Channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, 1979. 2.1

[17] I. Telatar, "Capacity of Multi-Antenna Gaussian channels," *European Transactions on Telecommunications*, 1999. 2.1, 3.1.5

[18] L. Zheng and D. Tse, "Diversity and Multiplexing: A Fundamental Tradeoff in Multiple-Antenna Channels," *IEEE Transactions of Information Theory*, vol. 49, no. 5, 2003. 2.1

[19] A. Sendonaris, E. Erkip, and B. Aazhang, "User Cooperation Diversity. Part I: System Description," *IEEE Transactions on Communications*, 2003. 2.1

[20] ——, "User Cooperation Diversity. Part II: Implementation Aspects and Performance Analysis," *IEEE Transactions on Communications*, 2003. 2.1

[21] S. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE Journal on Selected Areas in Communications*, 1998. 2.1

[22] "Rice University WARP Project." [Online]. Available: http://warp.rice.edu/ 2.2, 3.1

[23] C. Hunter, J. Camp, P. Murphy, A. Sabharwal, and C. Dick, "A Flexible Framework for Wireless Medium Access Protocols," in *Proceedings of IEEE Asilomar Conference on Signals, Systems, and Computers*, 2006. 2.2, 2.2.2

[24] K. Amiri, Y. Sun, P. Murphy, C. Hunter, J. R. Cavallaro, and A. Sabharwal, "WARP, a Unified Wireless Network Testbed for Education and Research," in *Proceedings of IEEE MSE*, 2007. 2.2

[25] ——, "WARP, a Modular Testbed for Configurable Wireless Network Research at Rice," in *Proceedings of IEEE SWRIF*, 2007. 2.2

[26] "WARPMAC Framework." [Online]. Available: http://warp.rice.edu/trac/wiki/WARPMAC 2.2.2

[27] "Rice University WARP OFDM Reference Design." [Online]. Available: http://warp.rice.edu/trac/wiki/OFDMReferenceDesign 2.2.2

[28] B. Raman and K. Chebrolu, "Revisiting MAC Design for an 802.11-based Mesh Network," in *Proceedings of Workshop on Hot Topics in Networks (HotNets)*, 2004. 2.2.2

[29] "Madwifi." [Online]. Available: http://madwifi.org/ 2.2.2

[30] MIT Roofnet. [Online]. Available: http://pdos.csail.mit.edu/roofnet/doku.php 2.2.2

[31] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "SoftMAC - Flexible Wireless Research Platform," in *Proceedings of Workshop on Hot Topics in Networks (HotNets)*, 2005. 2.2.2

[32] N. Abramson, "The ALOHA System - Another Alternative for Computer Communications," in *Proceedings of the Fall Joint Computer Conference*, 1970. 2.2.2

[33] "WARPMAC API." [Online]. Available: http://warp.rice.edu/WARP_API/warpmac_8c.html 2.2.2

[34] P. Murphy, C. Hunter, and A. Sabharwal, "Design of a Cooperative OFDM Transceiver," in *Proceedings of IEEE Asilomar Conference on Signals, Systems, and Computers*, 2009. 2.2.2, 3.1.3

[35] G. Jakllari, S. V. Krishnamurthy, M. Faloutsos, P. V. Krishnamurthy, and O. Ercetin, "A Framework for Distributed Spatio-Temporal Communications in Mobile Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, 2006. 3.1.1

[36] H. Zhu and G. Cao, "rDCF: a Relay-Enabled Medium Access Control Protocol for Wireless Ad Hoc Networks," in *Proceedings of IEEE INFOCOM*, 2005. 3.1.1

[37] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. Panwar, "CoopMAC: A Cooperative MAC for Wireless LANs," *IEEE Journal on Selected Areas in Communications*, 2007. 3.1.1

[38] G. Böcherer, de Baynast, and R. Mathar, "A Distributed MAC Protocol for Cooperation in Random Access Networks," *arXiv:0809.5204*, 2008. 3.1.1

[39] P. Liu, C. Nie, T. Korakis, E. Erkip, S. Panwar, F. Verde, and A. Scaglione, "STiCMAC: A MAC Protocol for Robust Space-Time Coding in Cooperative Wireless LANs," *IEEE Transactions on Wireless Communications*, no. 99, 2011. 3.1.1

[40] X. Zhang and K. Shin, "DAC: Distributed Asynchronous Cooperation for Wireless Relay Networks," in *Proceedings of IEEE INFOCOM*, 2010. 3.1.1

[41] "Azimuth ACE 400WB." [Online]. Available: http://www.azimuthsystems.com/platforms-channel-400wb.htm 3.1.4, 3.2.5

[42] G. Celine, "Effectively Testing MIMO-Enabled Wireless Devices," *RF DESIGN*, vol. 30, no. 8, 2007. 3.1.4

[43] V. Erceg, "TGn Channel Models," IEEE 802.11 document 03/940r4. 3.1.4

[44] T. Rappaport, *Wireless Communications*. Prentice Hall, 2002. 3.1.4.1

[45] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005. 3.1.5

[46] N. Ahmed, M. Khojastepour, and B. Aazhang, "Outage minimization and optimal power control for the fading relay channel," in *IEEE Information Theory Workshop*, 2004. 3.2.1

[47] W. Su, A. Sadek, and K. Liu, "SER Performance Analysis and Optimum Power Allocation for Decode-and-Forward Cooperation Protocol in Wireless Networks," in *Proceedings of IEEE Wireless Communications and Networking Conference*, 2005. 3.2.1

[48] N. Bambos, "Toward Power-Sensitive Network Architectures in Wireless Communications: Concepts, Issues, and Design Aspects," *IEEE Personal Communications*, 1998. 3.2.1

[49] J. Monks, V. Bharghavan, and W. Hwu, "A Power Controlled Multiple Access Protocol for Wireless Packet Networks," in *Proceedings of IEEE INFOCOM*, 2001. 3.2.1, 5.1

[50] E. Jung and N. Vaidya, "A Power Control MAC Protocol for Ad Hoc Networks," *Wireless Networks*, vol. 11, no. 1, 2005. 3.2.1

[51] Z. Zhou, S. Zhou, J. Cui, and S. Cui, "Energy-Efficient Cooperative Communication Based on Power Control and Selective Single-Relay in Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, 2008. 3.2.1

[52] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 Modeling and Simulation in ns-2," in *Proceedings of ACM MSWiM*, 2007. 3.2.4, 5.4.2

[53] B. Wang, Z. Han, and K. Liu, "Distributed Relay Selection and Power Control for Multiuser Cooperative Communication Networks Using Buyer/Seller Game," in *Proceedings of IEEE INFOCOM*, 2007. 4.1

[54] V. Srinivasan, P. Nuggehalli, C. Chiasserini, and R. Rao, "Cooperation in Wireless Ad hoc Networks," in *Proceedings of IEEE INFOCOM*, vol. 2, 2003. 4.1

[55] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*. Cambridge University Press, 2011. 4.1

[56] S. Bennett, *A History of Control Engineering, 1930-1955*. Peter Peregrinus Ltd, 1993, vol. 47. 4.4.2

[57] S. Gupta, C. Hunter, P. Murphy, and A. Sabharwal, "WARPnet: Clean Slate Research on Deployed Wireless Networks," in *Proceedings of ACM MobiHoc*, 2009. 4.5.1

[58] S. Gupta, P. Murphy, C. Hunter, and A. Sabharwal, *Cognitive Radio: System Design Perspective*. Springer, 2010, ch. WARPnet: A Platform for Deployed Cognitive Radio Experiments. 4.5.1

[59] "Rice University WARPnet Framework." [Online]. Available: http://warp.rice.edu/trac/wiki/WARPnet 4.5.1

[60] F. Kaltenberger, H. Jiang, M. Guillaud, and R. Knopp, "Relative Channel Reciprocity Calibration in MIMO/TDD systems," in *Proceedings of IEEE Future Network and Mobile Summit*, 2010. 1

[61] "The Network Simulator - ns-2." [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/User_Information 4.5.3

[62] M. Chiang, S. Low, A. Calderbank, and J. Doyle, "Layering as Optimization Decomposition: A Mathematical Theory of Network Architectures," *Proceedings of the IEEE*, 2007. 5

[63] V. Rodoplu and A. A. Gohari, "Challenges: Automated Design of Networking Protocols," in *Proceedings of ACM MobiCom*, 2008. 5

[64] A. Avestimehr, S. Diggavi, and D. Tse, "Wireless Network Information Flow: A Deterministic Approach," *IEEE Transactions on Information Theory*, 2011. 5.1

[65] K. Liu, H. Shin, and H. Chen, "Interference-Resistant Cooperative Wireless Networks Based on Complementary Codes," *Wireless Communications and Mobile Computing*, 2009. 5.1

[66] H. Lichte, S. Valentin, H. Karl, I. Aad, and J. Widmer, "Analyzing Space/Capacity Tradeoffs of Cooperative Wireless Networks Using a Probabilistic Model of Interference," in *Proceedings of ACM MSWiM*, 2009. 5.1

[67] A. Vasan, R. Ramjee, and T. Woo, "ECHOS-Enhanced Capacity 802.11 Hotspots," in *Proceedings of IEEE INFOCOM*, 2005. 5.1

[68] M. Cesana, D. Maniezzo, P. Bergamo, and M. Gerla, "Interference Aware (IA) MAC: an Enhancement to IEEE802. 11b DCF," in *Proceedings of IEEE VTC*, 2003. 5.1

[69] Z. Haas and J. Deng, "Dual Busy Tone Multiple Access (DBTMA)-a Multiple Access Control Scheme For Ad Hoc Networks," *IEEE Transactions on Communications*, vol. 50, no. 6, 2002. 5.1

[70] F. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part II – the Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution," *IEEE Transactions on Communications*, vol. 23, no. 12, 1975. 5.3.1

[71] V. Aggarwal, Y. Liu, and A. Sabharwal, "Sum-capacity of interference channels with a local view: Impact of distributed decisions," *IEEE Transactions on Information Theory*, 2012. 5.3.3.2

[72] A. Kamerman and L. Monteban, "WaveLAN-II: a High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical Journal*, vol. 2, no. 3, 1997. 6, 6

[73] G. Holland, N. Vaidya, and P. Bahl, "A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks," in *Proceedings of ACM MobiCom*, 2001. 6

[74] A. Bletsas, A. Khisti, D. Reed, and A. Lippman, "A Simple Cooperative Diversity Method Based on Network Path Selection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, 2006. 6

[75] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, 1996. 6

[76] C. Perkins and E. Royer, "Ad-hoc On-Demand Distance Vector Routing," in *Proceedings of IEEE WMCSA*, 1999. 6

[77] T. Halford and K. Chugg, "Barrage Relay Networks," in *Proceedings of Information IEEE Theory and Applications Workshop*, 2010. 6

[78] S. Gollakota and D. Katabi, "Zigzag Decoding: Combating Hidden Terminals in Wireless Networks," *Proceedings of ACM SIGCOMM*, vol. 38, no. 4, 2008. A

[79] "WARPLab Framework." [Online]. Available: http://warp.rice.edu/trac/wiki/WARPLab A

[80] "Analog Devices AD9777 Datasheet." [Online]. Available: http://www.analog.com/en/digital-to-analog-converters/da-converters/ad9777/products/product.html C.1

[81] "Analog Devices AD9248/BCP-65 Datasheet." [Online]. Available: http://www.analog.com/en/analog-to-digital-converters/ad-converters/ad9248/products/product.html C.1

[82] "Maxim MAX2829 Datasheet." [Online]. Available: http://www.maxim-ic.com/datasheet/index.mvp/id/4532 C.1

[83] "Anadigics AWL6950 Datasheet." [Online]. Available: http://www.anadigics.com/products/view/awl6950 1