

Design of a Cooperative OFDM Transceiver

Patrick Murphy, Christopher Hunter, and Ashutosh Sabharwal

Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005

Email: {murphpo, chunter, ashu}@rice.edu

Abstract—We present the FPGA implementation of a MIMO and cooperative OFDM physical layer transceiver. Our transceiver is designed to support multiple antenna configurations, including SISO (single-antenna), receive switching diversity, transmit diversity using Alamouti’s space-time block code and 2x2 spatial multiplexing. It also supports a fully-distributed physical layer cooperation scheme which allows two nodes to simultaneously transmit a common payload, achieving spatial diversity with only single-antenna transmissions. Our transceiver supports the common cooperative scheme of amplify-and-forward and provides a flexible interface for higher layers to enable cooperation as needed. The transceiver is implemented as a single FPGA core with a common datapath for all modes. This allows both efficient resource reuse between modes and a per-packet selection of antenna mode. This flexibility enables a wide variety of MIMO and cooperative protocols. We present architectural details of the cooperative transceiver design and early performance results using Rice University’s Wireless Open-Access Research Platform (WARP).

I. INTRODUCTION

Cooperative communications [1, and references therein] is a promising technique for pooling the resources of multiple single-antenna wireless nodes to achieve MIMO-like performance gains. Cooperative schemes are the focus of significant research interest, especially from a theoretical perspective. Our focus is on the less well-studied issues of implementing cooperative communications and to characterize the gains cooperation can enable in real-world wireless systems.

This paper presents the FPGA implementation of a cooperative OFDM physical layer transceiver, a key component of our ongoing work to explore real cooperative communications systems. In our previous work, we constructed a prototype of a cooperative communications as a proof-of-concept [2]. This previous design required an external connection between nodes to establish synchronization for cooperative transmissions. Our ongoing research project focuses on exploring various ways physical layer cooperation can be utilized in random access networks, where such external synchronization is infeasible. This application imposes a number of new requirements on the transceiver, necessitating a complete overhaul of the previous design. First, in order to build networks of cooperative nodes, the transceiver must operate stand-alone and in real-time. Second, it must provide a flexible interface to higher layers in order to support a variety of protocols which trigger cooperative transmissions. Finally, to build networks without centralized controllers, the transceiver must achieve the coordination and synchronization necessary for cooperative transmissions using only packets received wirelessly from other nodes.

Our cooperative OFDM transceiver meets all of these requirements. We present some of its key architectural features in this paper, along with experimental results from a real 3-node cooperative wireless link built using the transceiver. These experiments demonstrate significant performance gains provided by physical layer cooperation, reaching 80% improvement in packet error rate in some topologies.

The rest of this paper is organized as follows. Section II describes the architecture of our cooperative transceiver, Section III presents some experimental results and Section IV offers concluding remarks.

II. TRANSCEIVER DESIGN

A. Overview

Our cooperative OFDM transceiver is part of the Rice University Wireless Open-Access Research Platform [3]. WARP provides an FPGA-based hardware platform and an open-source repository of wireless building blocks and reference designs. The OFDM transceiver is implemented entirely in the fabric of the FPGA at the heart of the WARP hardware platform, which enables the design to interface with real RF interfaces and process packets in real-time. The transceiver is implemented in Xilinx System Generator [4], an FPGA design tool which uses The Mathworks Simulink for design entry and simulation. The source model for the transceiver is available in the WARP repository [3].

B. Architecture

Our cooperative OFDM design implements complete transmit and receive processing chains, realizing a true bytes-to-waveforms transceiver. The basic architecture is similar to any OFDM transceiver. The transmit path consists of a modulator, FFT, cyclic prefix insertion and filtering. The receiver consists of a front end (for packet detection, AGC and symbol timing estimation), IFFT, channel estimator, equalizer and detector. The Tx and Rx paths share a bank of packet buffers, which are also accessible to user code running in the FPGA’s PowerPC core. The transceiver also provides a large number of software-accessible registers, used to configure the core’s many parameters at run-time.

Achieving the goal of a transceiver capable of real-time distributed cooperation did impact the architecture of the transceiver in a number of ways. A few of these are discussed in detail below.

1) *Automatic Response Processing*: One of the primary goals of our design was 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. Our cooperative OFDM implementation builds on the WARP OFDM Reference Design, one of the open-source reference projects we maintain as part of the WARP repository [3]. In previous 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 $24 \pm 0.8 \mu\text{s}$.

This $1.6 \mu\text{s}$ 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 which enable the transmit and receive paths through the WARP hardware's radio transceiver. Second, it contains logic which 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 flexibility of this subsystem is motivated by our goal of building a transceiver which allows exploration of various protocols for triggering physical layer cooperation without having to design a custom PHY for each one.

The same subsystem can also be used to trigger non-cooperative transmissions. In this mode, 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 hardware.

Figure 1 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 our 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 ≈ 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.

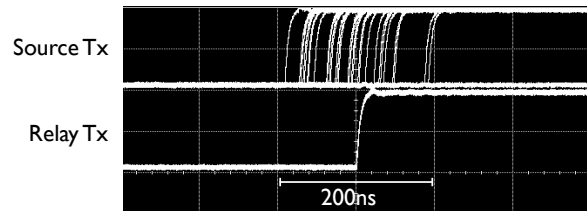


Fig. 1. Real-time capture of source transmission start relative to relay transmission start.

2) *PHY Frame Formats*: An important feature of the Alamouti space time block code is that the signal transmitted from each antenna (or each node, in a distributed version) contains the full packet payload. This provides the scheme's diversity, allowing the destination to receive the packet even if one stream is missing entirely. To preserve this feature in a real network, the Alamouti-encoded OFDM symbols must be wrapped in a frame format which also contains everything the receiver needs to process the packet should only a single node transmit. The frame format must include waveforms to enable packet detection, automatic gain control, carrier frequency offset correction, symbol timing, and channel estimation.

We use the frame formats illustrated in Figure 2. These formats were motivated by the 802.11a PHY specification. Our preamble has two parts: short training symbols, primarily for AGC convergence, and long training symbols, for carrier frequency offset and symbol timing estimation. Following the preamble are dedicated channel training symbols. The training symbols are transmitted for each stream, orthogonal in time. The Alamouti-encoded data follows the training, containing both the low-rate symbols (for the header) and full-rate symbols (for the user payload). The header and payloads are protected by separate checksums, allowing the MAC to take meaningful action upon receipt of a packet with a valid header but invalid payload.

3) *Carrier Frequency Offset*: CFO is a significant factor in the performance of OFDM systems [5]. Our OFDM receiver uses standard techniques for compensating for carrier offsets and phase noise [6, 7]. The effects are compounded when two nodes with different carrier frequency offsets are transmitting simultaneously to a common destination. Handling CFO for cooperative transmissions depends on the cooperative scheme. With amplify-and-forward, it is possible to ignore

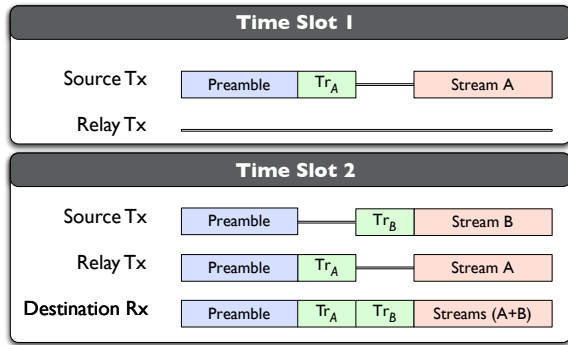


Fig. 2. Physical layer frame formats.

offsets relative to the relay entirely when oscillator drift is negligible on packet timescales [2]. An AF relay which uses the same carrier reference for downconversion and upconversion passively inverts the source-relay CFO when re-transmitting the analog waveform. From the destination's perspective, it receives a waveform subject to only the source-destination CFO. We exploit this property in our AF implementation and experimentally confirm it by observing identical CFO measurements at the destination during a source-only transmission and the subsequent AF transmission.

III. EXPERIMENTS

A. Integration

The cooperative OFDM transceiver is just one of many pieces required to build a fully operational wireless node. In order to evaluate our transceiver, we integrate it with a number of additional FPGA cores from the WARP repository. These include packet buffers, a timer and controllers for the various components on the WARP radio board. We also utilize cores from the Xilinx design tools, such as an Ethernet MAC and general purpose I/O controllers for real-time debug and monitoring of the design in hardware

We also use the FPGA's embedded PowerPC core extensively. Software running in the PPC handles PHY configuration and control, packet transfers between the wireless and Ethernet interfaces, and the recording of performance metrics. The code is built from both custom and Xilinx-provided peripheral drivers, the WARP MAC [8] framework for packet processing and PHY control and frameworks for running experiments and gathering statistics. The code also enables remote control of the node via Ethernet, which allows a central computer to configure multiple nodes, execute an experiment and extract results from each participating node. A block diagram of the full implementation is shown in Figure 3.

Table I lists the key parameters of our cooperative OFDM PHY implementation. It also lists the rates and packet sizes used in the experiments discussed below.

Table II lists the FPGA resource usage for both the PHY and the overall FPGA design built to characterize the PHY in real-time. We used a WARP FPGA Board v1 with a Xilinx XC2VP70 FPGA for these experiments.

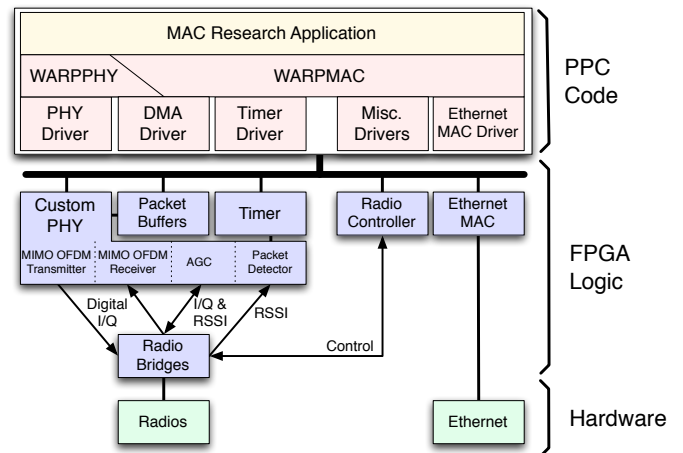


Fig. 3. Full wireless node block diagram, integrating cooperative transceiver and other PHY blocks.

TABLE I
PHY & EXPERIMENT PARAMETERS

Carrier Frequency	2427MHz
Transmit Power	10dBm
RF Bandwidth	10MHz
OFDM Symbol	64 subcarriers
OFDM Cyclic Prefix	1.6 μ s
Preamble Duration	48 μ s
Header Length	24 bytes
Header Rate	BPSK at 6Mbps
Payload Length	1016 bytes
Payload Rate	QPSK at 12Mbps
Packet Duration	757 μ s

TABLE II
FPGA RESOURCE USAGE

FPGA Resource	PHY Tx	PHY Rx	Full Design	Available (XC2VP70)
Slices	3771	9651	22430	33088
Multipliers	52	85	163	328
Block RAMs	9	138	304	328
I/O	-	-	548	964

B. Experimental Methodology

Our experimental setup consists of three WARP nodes, each consisting of a WARP FPGA and radio board. Every node is configured with the same design, implementing the full cooperative OFDM transceiver and the supporting FPGA cores and code to utilize the PHY in real-time. For each trial, the source node transmits random packets with 24 byte headers and 1016 byte payloads. The source transmits each data packet twice. In the first slot, the relay both attempts to decode the packet and records the raw I/Q samples it receives from the RF interface. If the relay is able to successfully decode the packet (i.e. both the header and payload checksums indicate no errors occurred), it enables its amplify and forward subsystem, preparing to re-transmit the received packet's raw waveform. The source also prepares to re-transmit the same packet. The two cooperating transmissions are triggered by a control packet transmitted from the destination node. These simultaneous transmissions constitute the second time slot, during which the destination attempts to decode the incoming packet.

This protocol is similar to the selection relaying scheme described by Laneman [9]. It is important to note that our transceiver does not impose this scheme; the conditions for triggering re-transmissions and cooperative transmissions are configured entirely by user code at run-time. Many protocols, including those in [9], can be explored using our design.

In order to realize a repeatable propagation environment, we use an Azimuth ACE 400WB wireless channel emulator [10, 11] to connect the RF interfaces of the WARP nodes. This emulator is designed 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 illustrated in Figure 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).

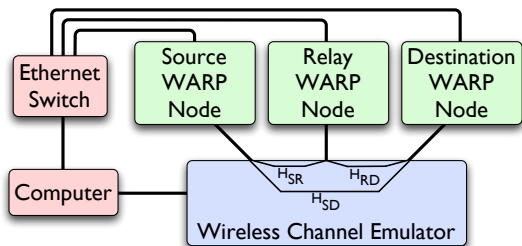


Fig. 4. Hardware connections and emulated channels for PHY experiments.

We chose the channel model for our tests from the TGn family of models proposed for the IEEE 802.11n standard [12]. Specifically, we used TGn model B, which models a channel with nine taps and delay spread of 80ns. This model realistically captures the scaling, phase and dispersion effects of scattering in an indoor environment. It specifies a Doppler spread of 2.6Hz, which we adopt in our experiments.

For every test, the three forward links ($S \rightarrow D$, $S \rightarrow R$, and $R \rightarrow D$) use the same model and have identical fading statistics but 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.

The reverse channels ($D \rightarrow S$ and $D \rightarrow R$) are configured as non-fading with very high SNR. This allows the source and relay to receive the destination's control packet with near certainty. In fact, we measured no losses of the control packet in any of our tests. This approach is designed to enable characterization our cooperative PHY directly without the complications of a higher-layer MAC protocol. Our transceiver does not require a perfect feedback path to operate. In fact, one of our primary goals in designing a distributed cooperative system is to explore protocols which preserve cooperative gains when feedback paths, frequently assumed to be ideal, are subject to the same wireless degradations as other transmissions.

C. Topologies

The channel emulator enables experiments with arbitrary topologies, achieved by choosing various average SNRs be-

tween each node. We chose to explore two topologies. In the first, the source-destination average SNR is fixed to emulate a physical separation of approximate 20m. We then sweep the SNR of the source-relay and relay-destination channels to emulate multiple relay positions along the source-destination line.

The second topology fixes the source-relay channel at a high SNR and sweeps the destination's position. In this setup, the source-destination and relay-destination channels have the same average SNR (i.e. same distance) but independent instantaneous channel coefficients. This configuration models the real-life scenario of co-located source and relay nodes, such as a single user's laptop and mobile phone cooperating to communicate with a distant base station.

D. Metrics

We use bit error rate (BER) and packet error rate (PER) as the metrics for our experiments. A packet is in error when at least one bit error occurs in either its header or payload, or if the receiver fails to detect the packet at all. Bit error rate is measured in real-time by the receiver only for packets whose headers are received without error. Packets with header errors and undetected packets are excluded from the BER tabulation. A final metric is the probability of cooperation. This is the probability of the relay participating in the second time slot of a packet exchange. The relay only participates if it is able to decode the packet it receives from the source.

E. Results

The results of our experiments are presented in the plots below. Each data point in these plots represent aggregate results from approximately 60,000 packet transmissions. The traces labeled non-cooperative reflect performance of the a source-destination link under identical channel conditions as the cooperative case, but with the relay disabled.

Figure 5(c) shows the bit error rate, packet error rate and probability of relay participation for the co-linear topology described above. The blocks labeled S and D indicate the position of the source and destination nodes. The peak performance improvement (80% reduction in both BER and PER) is significant and, as expected, occurs when the relay is positioned between the source and destination.

In this figure, the independent variable is presented as distance in meters. The actual experimental variable, as configured in the channel emulator, is the average path loss (in dB) along each emulated wireless channel. The mapping of path loss to distance requires selection of a path loss exponent representative of the propagation environment [13]. We chose 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.

Figure 6 presents results for the co-located source/relay topology described above. In this plot, the independent variable

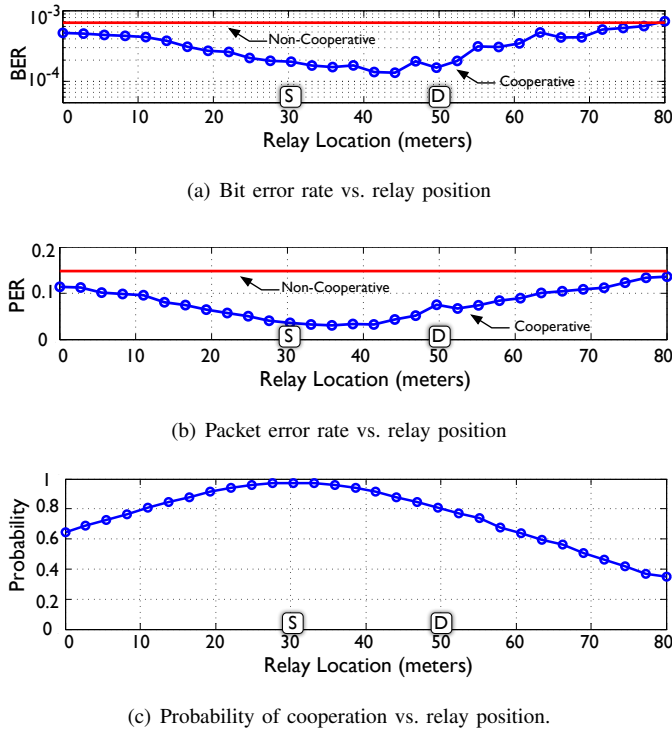


Fig. 5. Experimental results for linear topology, sweeping relay position.

is the average path loss of the source-destination and relay-destination channels. The packet error rates are presented on a logarithmic scale to illustrate both the significant performance gain enabled by cooperation as well as the steeper slope of the cooperative curve, indicating a realized diversity gain. The peak PER gain provided by cooperation in this topology is 6.5dB, 3dB of which represents the extra transmission power supplied by the relay, with the rest attributable to diversity gain.

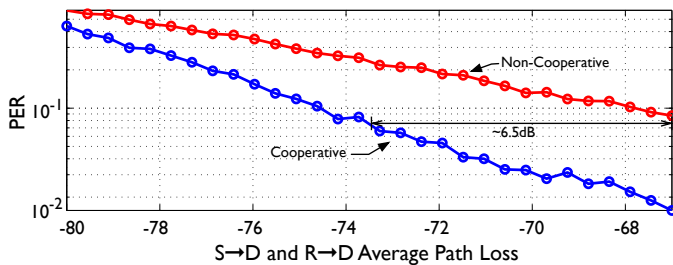


Fig. 6. Packet error rate vs. destination position for co-located source and relay.

IV. CONCLUSION

We have presented the design and evaluation of a real-time cooperative OFDM transceiver. This design is one component of our ongoing effort to explore the utility of physical layer cooperation in real-world wireless systems. Multiple extensions to our design are possible, including support for MRC across two time slots and support for cooperation-aware error correcting codes. Our future work will also include utilization of

this transceiver by designs seeking to exploit physical layer cooperation to improve performance at higher network layers. We also intend to explore the impact of cooperation on network-level performance, especially with regards to spatial reuse and interference management.

ACKNOWLEDGMENTS

The authors would like to thank Rice University undergrad Richard Latimer for his significant contributions to our test automation framework and to Azimuth Systems for the use of the ACE 400WB emulator. We would also like to thank Xilinx, Inc. and the Xilinx University Program for their continuing support of the WARP project. This work was partially funded by NSF grants CNS-0325971, CNS-0551692 and CNS-0619767.

REFERENCES

- [1] G. Kramer, I. Marić, and R. Yates, "Cooperative communications," *Foundations and Trends in Networking*, vol. 1, no. 3, 2006.
- [2] 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 (in press).
- [3] "Rice University WARP Project." [Online]. Available: <http://warp.rice.edu/>
- [4] "Xilinx System Generator for DSP." [Online]. Available: <http://xilinx.com/sysgen>
- [5] T. Pollet, M. Van Bladel, and M. Moeneclaey, "BER sensitivity of OFDM systems to carrier frequency offset and Wiener phase noise," *IEEE Trans. on Communications*, vol. 43, no. 234, 1995.
- [6] P. Moose, "A technique for orthogonal frequency division multiplexing frequency offset correction," *IEEE Trans. on Communications*, vol. 42, no. 10, 1994.
- [7] R. Sakata, K. Akita, and K. Sato, "Real-time phase tracking method for IEEE 802.11 a/g/n receiver under phase noise condition," in *IEEE Vehicular Technology Conference*, vol. 4, 2006.
- [8] C. Hunter, J. Camp, P. Murphy, A. Sabharwal, and C. Dick, "A flexible framework for wireless medium access protocols," in *Asilomar Conference on Signals, Systems and Computers*, 2006.
- [9] J. Laneman, D. Tse, and G. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *IEEE Trans. Information Theory*, vol. 50, no. 12, 2004.
- [10] "Azimuth ACE 400WB." [Online]. Available: <http://www.azimuthsystems.com/platforms-channel-400wb.htm>
- [11] G. Celine, "Effectively testing MIMO-enabled wireless devices," *RF DESIGN*, vol. 30, no. 8, p. 40, 2007.
- [12] V. Erceg, et al., "TGN channel models," IEEE 802.11 document 03/940r4.
- [13] T. Rappaport, *Wireless communications*. Prentice Hall PTR New Jersey, 2002.