Real-Time Testbed Implementation of a Distributed Cooperative MAC and PHY

Christopher Hunter, Patrick Murphy, and Ashutosh Sabharwal
Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005
Email: {chunter, murphpo, ashu}@rice.edu

***Abstract*—We present the implementation and experimental evaluation of a new, fully distributed protocol for random access systems that exploits symbol-level physical layer cooperation. By allowing single-antenna nodes to cooperate with their neighbors, MIMO-like performance is achieved. Our Distributed On-demand Cooperation (DOC) protocol is unique in its ability to realize a cooperative mode only under circumstances where cooperation can assist. Thus, under high SNR scenarios where cooperation is rarely necessary, DOC gracefully reverts to a standard CSMA/CA protocol. Our implementation of the custom DOC MAC and PHY is built on the Rice University Wireless Open-Access Research Platform (WARP). It operates in real-time without any offline processing, allowing for standalone operation and packet exchanges at timescales comparable to commercial IEEE 802.11 devices. This implementation of the DOC MAC/PHY system addresses real-world degradations like imperfections in synchronization and link-level coordination. Extensive experimental results demonstrate that our implementation delivers substantial improvement in end-to-end throughput over that of a non-cooperative link.**

I. INTRODUCTION

Multiple antenna technology [1, 2], while commonplace in many wireless standards, cannot be effectively implemented on small form-factor handheld devices. The primary reason is that for maximal gains, multiple antennas should have separation of 10-30cm for center frequencies of 1.8-2.5GHz. To address the antenna limitation on handsets, an alternate approach is to pool the distributed resources of neighboring nodes. Labeled as *physical layer cooperative communications* [3, and references therein], the main idea is to harness the antennas of different nodes to achieve MIMO-like gains. However, many cooperative protocols assume tight symbol-level synchronization across nodes, which is often not implemented in random access networks. As a result, it is unclear how well cooperative gains predicted by theory can be realized in random access networks.

In this paper, we propose and complete a real-time testbed implementation of a new distributed cooperative protocol known as Distributed On-demand Cooperation (DOC). Our implementation shows that *at least* 20% throughput gains can be realized 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 and not in the event of packet errors due to collision. 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.

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 when relaying is available cooperative mode gains are automatically realized, and when relaying is not available the system automatically reduces to a traditional non-cooperative CSMA/CA system without requiring any adjustments at the physical or MAC layers. Figure 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.

As part of the overall Rice WARP research project, all implementation source files for the DOC MAC and PHY design will be made available in the WARP repository [4].

Related Work: The related work can be divided into three categories: information theoretic, protocol development, and testbed evaluation. There is a significant body of work on the information-theoretic aspects of cooperative communications [3, 5, and references therein]. These works are largely geared towards arriving at upper bounds on achievable performance for different network topologies, by working with idealized models for analytical tractability. The key assumptions include perfect time-synchronization, often combined with perfect channel estimation and no radio distortions or imperfections. Our work, in contrast, is largely focused on understanding the impact of implementation challenges on cooperative systems.

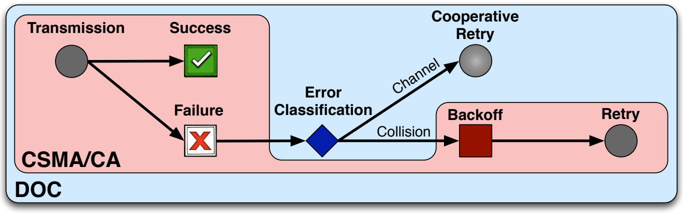


Fig. 1. DOC employs cooperation only under scenarios when it can help while gracefully reverting to a standard CSMA/CA access mechanism when it cannot.

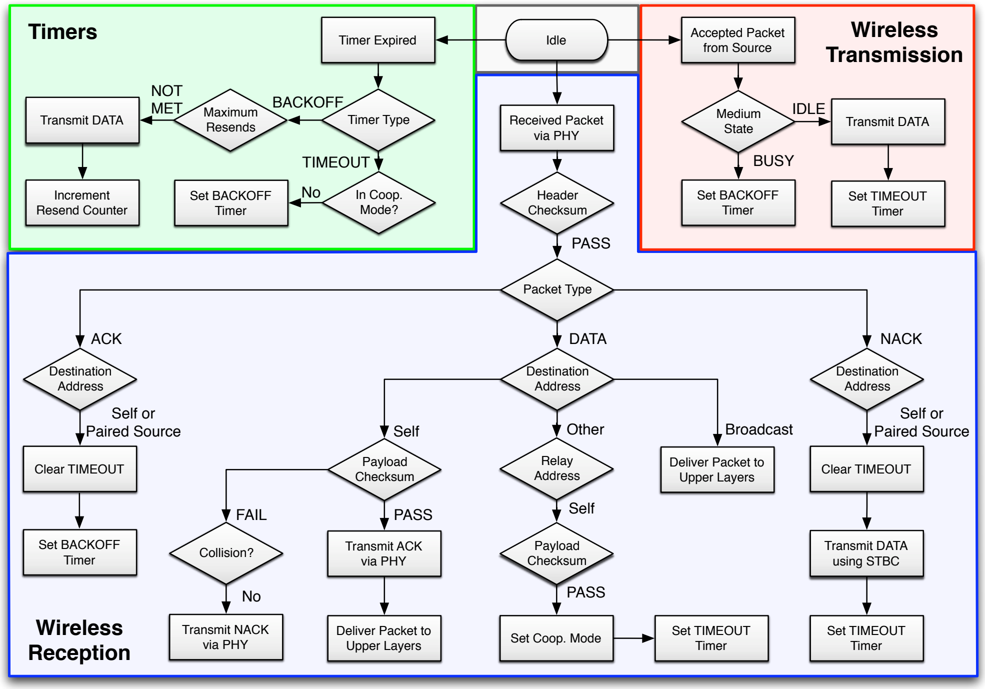


Fig. 2. The DOC MAC protocol can be visualized as a state machine. The protocol is completely distributed, allowing each node in a network to traverse this state diagram independently.

The work on cooperative MAC protocols is relatively sparse [6–9]. The work in [6] 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 [7–9] are designed for distributed systems but rely on orthogonal transmissions, and hence can be cast as opportunistic routing. In contrast, the gains from DOC are due to simultaneous transmissions in the same frequency band by the source and relay during the cooperative phase.

Finally, we note there are very few implementations of cooperative protocols [10–12]. 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. They also utilized BER as the performance metric. Further, in [11, 12] the implementation was geared towards narrowband systems, approximately 68kHz 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 planned extensions up to 20MHz, the operating regime of most wideband wireless networks. In [13], 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. Since our real-time implementation includes the MAC, our work enables the future construction of deployable large-scale networks.

Paper Outline: The rest of this paper is organized as follows. Section II presents a detailed description of the DOC protocol and the requirements it imposes on an implementation. Our real-time implementation of the DOC MAC and PHY is discussed in Section III. The design of our experiments and choice of experimental parameters is presented in Section IV. Our experimental results are discussed in Section V, and Section VI offers concluding remarks.

II. PROTOCOL DESCRIPTION

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

The key mechanism in DOC is the identification of when a cooperative mode is needed. We use the fact that packet headers are often much better protected than payloads. For example, in 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 utilizes a mid-packet collision detector, similar to that in [14], to avoid triggering a cooperative phase if a mid-packet collision was the cause of payload error. This ensures that the relay does not send cooperative packets while other nodes are attempting to resolve contention for medium access.

---

[1] In 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.

The MAC-level behavior of DOC can be broken up into three basic branches, as shown in Figure 2. One branch handles packets passed down from a higher 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.

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, it transmits the packet and starts a timeout timer to wait for an acknowledgment. If the medium is busy, it 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 2 are implicit in this way. Notice that this behavior is identical to traditional CSMA/CA basic access mechanisms like the 802.11 DCF.

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) 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 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.

III. IMPLEMENTATION

In order to realize a complete real-time wireless system, we use the Rice Wireless Open-Access Research Platform [4]. WARP combines an FPGA-based hardware platform and open-source repository of MAC/PHY building blocks in the form of WARPMAC [15] and WARP OFDM PHY frameworks [16]. Our implementation includes both a real-time cooperative OFDM physical layer, running in the fabric of the FPGA, and the MAC protocol, whose processing is divided between logic and C code running in the FPGA's PowerPC core. The real-time physical layer we employ is the custom cooperative physical layer described in our previous work [16]. In this work, 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 [16].

Table I lists the key parameters of our DOC implementation. Table II lists the FPGA resource usage for the full DOC MAC/PHY design as implemented in the WARP FPGA Board's Xilinx XC2VP70 FPGA.

TABLE I
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 II
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%)

IV. EXPERIMENTS

A. 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 DOC MAC/PHY design, implementing the full MAC protocol and OFDM transceiver. For each trial, the source node generates random packets

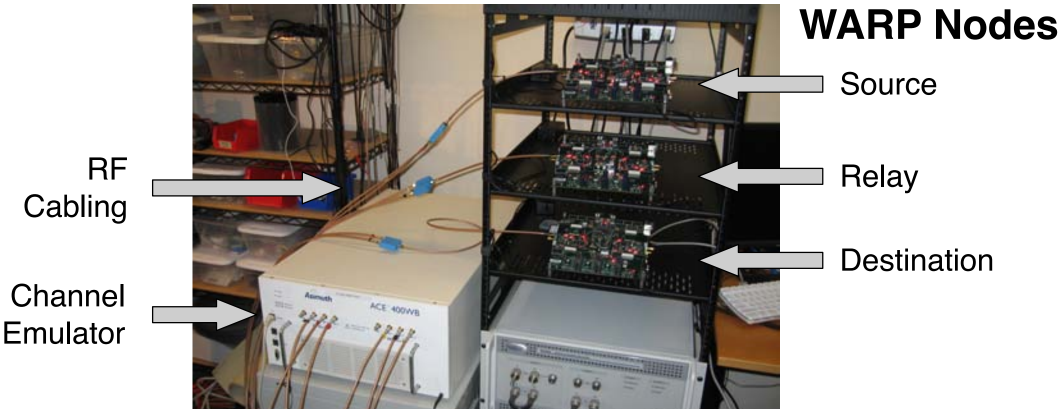


Fig. 3. A channel emulator is used to control average SNR between each node as well as apply random fading elements to each link.

with 24 byte headers and 1470 byte payloads. The source is fully backlogged, immediately initiating a new transmission whenever the previous one succeeds or times out. 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 an Azimuth ACE 400WB wireless channel emulator [17, 18]. 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. 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).

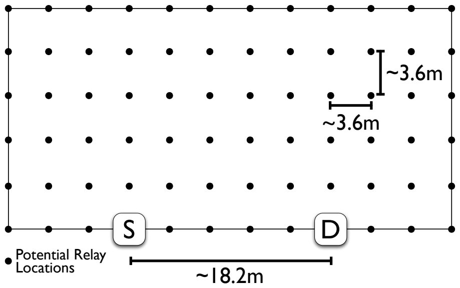
We choose the channel model for our tests from the TGn family of models proposed for the IEEE 802.11n standard [19]. Specifically, we use TGn model B, which models a channel with nine taps and delay spread of 80ns. This model realistically captures the scaling, phase and dispersion effects of scattering in an indoor environment. It specifies a Doppler spread of 2.6Hz, which we adopt in our experiments.

For every test, the three wireless links (S-D, S-R and R-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.

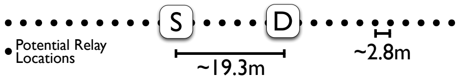
B. Topologies

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 20m, yielding a regime where many packets are being lost due to channel effects. With this distance (and thus SNR) fixed, we run experiments which sweep the position of the relay in a space around the source and destination nodes. As shown in Figures 4(a) and 4(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 only test 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.

In the following figures, the independent variable is presented as distance in meters. 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 [20]. 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.



(a) 2-D Topology



(b) 1-D Topology

Fig. 4. Node locations for each experiment, with the relay location as the independent variable.

C. 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.

D. 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 interfering with the protocol itself. Figure 5 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 5 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

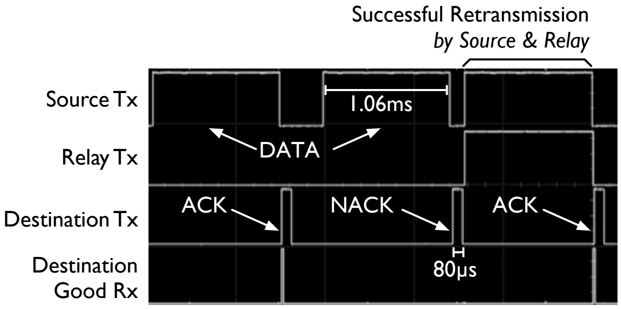


Fig. 5. Real-time Tx/Rx events showing a successful cooperative retransmission.

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.

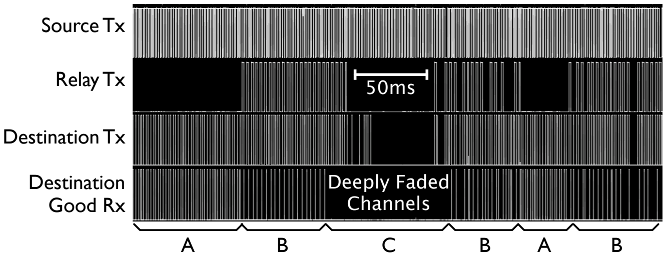
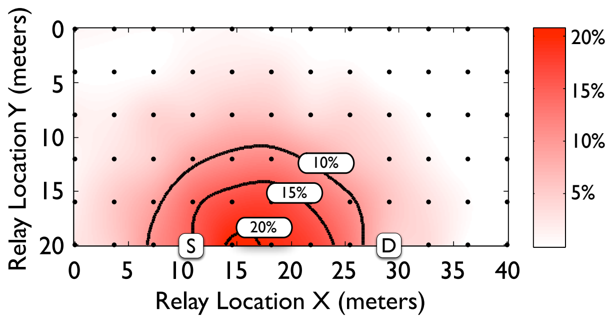


Fig. 6. Real-time Tx/Rx events showing on-demand cooperation initiated by packet losses due to channel fades. The labels highlight various channel conditions, where the source-destination link does not require help from the relay (A), where the S-D channel degrades sufficiently that the relay actively cooperates (B), and where the S-D link degrades to the point that it cannot sustain any communication (C).

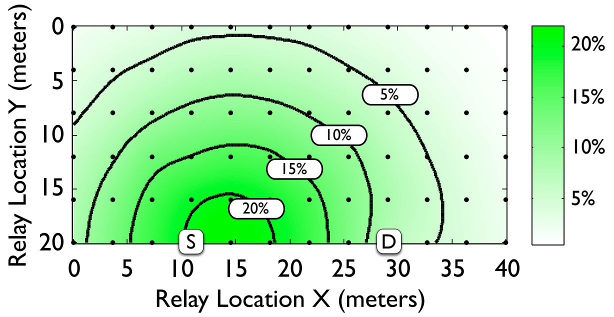
Figure 6 uses the same four signals as the previous figures, viewed over a much longer time scale (approximately 325ms). At this scale, it is possible to visualize channel variations and the resulting node transmissions. The node behaviors in Figure 6 are the real-time reactions of the cooperative MAC/PHY to the random channel coefficients imposed by the emulator.

V. EXPERIMENTAL RESULTS

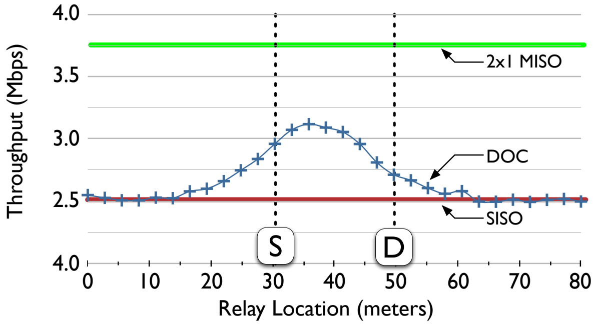
Figure 7 presents experimental results gathered at 72 locations for the relay with fixed source and destination locations. Each dot in the contour plots represent a relay location, with the contour lines tracing constant paths through values interpolated between relay locations. Figure 7(a) shows throughput improvement of a relay-aided cooperative link over a non-coopeartive SISO link, while 7(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. This is an intuitive result, as this proximity mimics the transmission of a two-antenna node. The peak performance improvement is significant, exceeding 20%, even though the relay was engaged in fewer than 25% of packet exchanges. This has two important implications. First, relay energy drain is mitigated simply by virtue of the fact that its operational duty cycle is small relative to the source. Second, in larger multi-flow networks, we can conjecture that the potential interference caused by the relay on other neighboring flows is also lessened. The full impact of DOC in larger network settings is an open question.



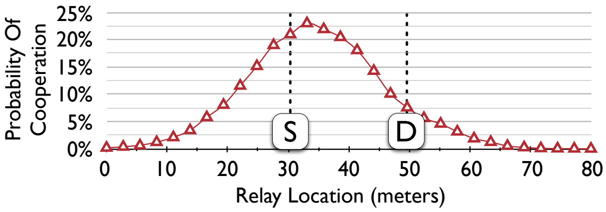
(a) Throughput improvement relative to a non-cooperative SISO link as a function of relay location.



(b) Probability of the relay cooperating vs. position.



(c) Throughput comparision of a SISO link without aid from a relay, a MISO 2×1 Alamouti link and DOC cooperative link.



(d) Probability of cooperation.

Fig. 7. Experimental results for cooperative implementation.

Figure 7 also presents results of throughput (7(c)) and probability of cooperation (7(d)) 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-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 same as a SISO system, which implies that the asymptotic growth of capacity for MISO and SISO have the same slope [21]. However, at the finite SNRs of interest, 2×1 MISO reduces packet losses due to an added diversity branch [22], 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 any cooperative scheme cannot. Most importantly, our cooperative implementation strictly outperforms the SISO link and achieves a significant fraction of the performance gain possible with true MISO.

VI. CONCLUSION

Our work was driven by the questions of feasibility and utility of cooperative communications in random access networks. Towards that end, our real-time implementation provides a promising start by settling the driving questions. While the experimental results presented here offer an encouraging look at the role of cooperation in networks, broader multi-flow network considerations are still unknown. The implementation of DOC on WARP enables the future consideration of these very issues.

Many significant improvements to our implementation are possible and underway, e.g, the use of maximal-ratio combining across two time-slots, incremental coding in the form of HARQ, and impact of physical layer cooperation on spatial reuse.

ACKNOWLEDGMENTS

The authors would like to thank Azimuth Systems for the use of the ACE 400WB emulator. We would also like to thank Xilinx for their continuing support of the WARP project. This work was partially funded by NSF grants CNS-0551692 and CNS-0619767.

REFERENCES

[1] E. Biglieri, *MIMO wireless communications*. Cambridge University Press, 2007.

[2] H. Bölcskei, *Space-time wireless systems: from array processing to MIMO communications*. Cambridge University Press, 2006.

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

[4] "Rice University WARP Project." [Online]. Available: http://warp.rice.edu/

[5] 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.

[6] 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.

[7] H. Zhu and G. Cao, "rDCF: a relay-enabled medium access control protocol for wireless ad hoc networks," in *Proceedings of IEEE INFOCOM*, vol. 1, 2005.

[8] P. Liu, Z. Tao, S. Narayanan, T. Korakis, and S. Panwar, "CoopMAC: A cooperative MAC for wireless LANs," *IEEE Journal on Selected Areas in Comm.*, vol. 25, no. 2, 2007.

[9] G. Böcherer, de Baynast, and R. Mathar, "A distributed MAC protocol for cooperation in random access networks," *arXiv:0809.5204*, 2008.

[10] 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.

[11] G. Bradford, "A framework for implementation and evaluation of cooperative diversity in software-defined radio," Master's thesis, University of Notre Dame, 2008.

[12] G. Bradford and J. N. Laneman, "An experimental framework for the evaluation of cooperative diversity," in *Proceedigs of CISS*, 2009.

[13] X. Zhang and K. Shin, "Dac: Distributed asynchronous cooperation for wireless relay networks," in *Proceedings of IEEE INFOCOM*, 2010.

[14] S. Gollakota and D. Katabi, "Zigzag decoding: combating hidden terminals in wireless networks," *SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 159–170, 2008.

[15] 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.

[16] P. Murphy, C. Hunter, and A. Sabharwal, "Design of a cooperative OFDM transceiver," in *Asilomar Conference on Signals, Systems and Computers*, 2009.

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

[18] G. Celine, "Effectively testing MIMO-enabled wireless devices," *RF DESIGN*, vol. 30, no. 8, p. 40, 2007.

[19] V. Erceg, et al., "TGn channel models," IEEE 802.11 document 03/940r4.

[20] T. Rappaport, *Wireless communications*. Prentice Hall PTR New Jersey, 2002.

[21] I. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. on Telecommunications*, 1999.

[22] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge University Press, 2005.