DAC: Distributed Asynchronous Cooperation for Wireless Relay Networks

Xinyu Zhang and Kang G. Shin
Department of Electrical Engineering and Computer Science
The University of Michigan
Email: {xyzhang, kgshin}@eecs.umich.edu

Abstract*—Cooperative relay is a communication paradigm that aims to realize the capacity of multi-antenna arrays in a distributed manner. However, the symbol-level synchronization requirement among distributed relays limits its use in practice. We propose to circumvent this barrier with a cross-layer protocol called *Distributed Asynchronous Cooperation* (DAC). With DAC, multiple relays can schedule concurrent transmissions with packet-level (hence coarse) synchronization. The receiver then extracts multiple versions of each relayed packet via a collision-resolution algorithm, thus realizing the diversity gain of cooperative communication. We demonstrate the feasibility of DAC by prototyping and testing it on the GNURadio/USRP software radio platform. To explore its relevance at the network level, we introduce a DAC-based MAC, and a generic approach to integrate the DAC MAC/PHY layer into a typical routing algorithm. Considering the use of DAC for multiple network flows, we analyze the fundamental tradeoff between the improvement in diversity gain and the reduction in multiplexing opportunities. DAC is shown to improve the throughput and delay performance of lossy networks with medium-level link quality. Our analytical results are also confirmed by network-level simulation in ns-2.

I. Introduction

It has been well-understood in information theory that relays' cooperation can improve the rate and reliability of wireless links [1]. A typical cooperative communication protocol allows a relay to overhear the source's transmission, and then forward the data to the desired receiver in case the direct delivery attempt failed. Such a two-stage cooperative relay protocol essentially establishes a virtual antenna array among multiple distributed single-antenna transmitters, so that the link capacity from the source to the destination may be boosted.

Orthogonal space-time codes [2], originally designed for point-to-point MIMO links, have been proposed to exploit the additional degrees of freedom (referred to as the *diversity gain*) offered by relay nodes. Non-orthogonal schemes [3] that allow relays and the source to transmit concurrently in the forwarding stage can achieve the same level of diversity gain as MIMO. In these seminal information-theoretic approaches, perfect time synchronization among relays is assumed *a priori*. However, unlike point-to-point MIMO links, cooperative communication is asynchronous by its nature since there is no global clock shared by the relays. The randomness introduced by propagation delay and higher-layer operations typically generates a time offset/skew in the order of several microseconds or more [4]. In contrast, the typical symbol duration of wireless communication standards (*e.g.*, 802.11) is well below 1 μs, and even half a symbol shift in time will completely offset the advantage of synchronous cooperative communications [4].

The work described in this paper was supported in part by the NSF under Grant CNS-0721529.

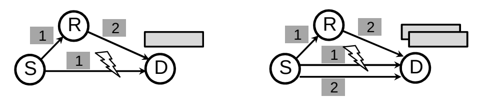

Fig. 1. A contrast between traditional relaying and DAC. The shaded tags denote the order of transmission.

In this paper, we circumvent the synchronization barrier with a cross-layer relay protocol called **D**istributed **A**synchronous **C**ooperation (DAC). In contrast with orthogonal relay protocols (Fig. 1(a)), DAC allows two relays (or the source and one relay) to concurrently forward the same packet to the destination (Fig. 1(b)). Even if one of them fails, the other can still be decoded without incurring additional channel access time. Hence, DAC improves the link reliability by exploiting additional spatial diversity from co-located relays.

Unlike the non-orthogonal relaying in information theory [3], DAC only needs to maintain coarse-grained packet-level synchronization among relays, which is achieved via MAC-layer sensing and scheduling. At the PHY layer, it extracts multiple versions of a packet from different relays using a *collision resolution* algorithm. Specifically, DAC takes advantage of the natural time offset among these packet copies to decode clean bits in the packet. It bootstraps an iterative cancellation procedure that recovers those symbol positions where different bits collide, by remodeling the known bits and cancelling them from collided symbols.

To make the above idea concrete, we design and implement the DAC PHY layer on the GNURadio/USRP software radio platform [5]. The core components in our design include packet-offset identification, channel parameter estimation, and sample-level signal modeling and cancellation, which are detailed in Sec. III. Our experimentation on a small relay network show that DAC can indeed make a diversity gain for typical SNR ranges.

To translate this advantage into network performance enhancement, we design a MAC protocol that extends the widely-used CSMA/CA and integrates the DAC PHY with it. A key idea in our design is to use *cut-through relaying* to maintain maximal compatibility with the 802.11-style mechanism. Specifically, the relays forward a packet immediately (without buffering it) upon overhearing or seeing a retransmission header from the original source node. Hence, the relays make transparent contributions without disrupting the retransmission, carrier sensing and exponential backoff decisions at the source.

We further introduce a generic approach that can use the DAC MAC/PHY to improve existing routing protocols. In applying this approach to multiple network flows, we identify an important tradeoff between the diversity gain provided by concurrent relays, and the multiplexing loss due to expanded interference region. Our analysis reveals that DAC improves network throughput when the link loss rate is below a certain threshold, which can be exactly profiled for simplified topologies. Therefore, DAC is best applicable to lossy wireless networks (such as unplanned mesh networks [6]), where it can enhance the network throughput by improving the reliability of bottleneck links with a low reception rate.

Due to the limitation of our software radio platform (*i.e.*, USRP), we cannot directly implement the DAC-based MAC and routing protocols. Therefore, we develop an analytical model with closed-form characterization of DAC's achievable bit error rate (BER) and packet error rate (PER). We modify the ns-2 PHY with this new packet reception model, and implement the DAC MAC and routing protocol based on it. Our simulation experimentation demonstrates that DAC can significantly improve the throughput and delay performance of existing loss-aware routing protocols. It thereby reveals the potential and practicality of non-orthogonal cooperation in wireless relay networks.

The main contributions of this work can be summarized as follows.

- We design and implement an asynchronous, non-orthogonal relaying scheme and test it on an actual ratio platform. The PHY layer BER and PER performances are characterized theoretically.
- We design a MAC protocol that exploits distributed asynchronous relays with minimal signalling overhead and maximal transparency to the 802.11 MAC.
- We propose a generic approach that incorporates the DAC-based relaying scheme into existing routing protocols. Based on an asymptotic analysis in tractable network models, we profile the sufficient condition when DAC improves the performance of existing routing protocols.

The rest of this paper is organized as follows. Sec. II discusses related efforts in wireless relay networks. Sec. III describes the design and implementation of the DAC PHY. The DAC-based MAC and routing protocols are presented in Sec. IV and V, respectively. Sec. VI analyzes the BER, PER, and network-level asymptotic performance for DAC. Further simulation experiments are presented in Sec. VII to validate DAC's performance. Finally, Sec. VIII concludes the paper.

II. Related Work

Cooperative diversity was originally proposed in information theory to realize the capacity of MIMO systems. The distributed space-time code [2] for two-stage cooperative communications has been widely explored to improve the performance of relay networks (see [7] for a survey). Azarian *et al.* [3] showed that non-orthogonal cooperation schemes can approximate the performance of centralized MIMO systems through multiple relays. However, these cooperative relay protocols assume perfect time synchronization among relay nodes. Recently, Wei [8] and Li *et al.* [9] reduced the synchronization constraint to sub-symbol level, but assumed known and controllable time offsets between relays. DAC's diversity gain is incomparable with such synchronized schemes, and it only allows for two concurrent relays. However, to our knowledge, it is the first non-orthogonal relaying protocol without any symbol-level timing constraint.

The implication of cooperative relaying for higher layers has been studied recently. Jakllari *et al.* [10] directly applied the synchronized space-time code to establish virtual MISO links for routing. Sundaresan *et al.* [11] showed that the more practical two-phase orthogonal relaying scheme (Fig. 1(a)), driven by the retransmission diversity from relays equipped with smart antennas, can make a remarkable throughput gain.

An alternative approach to exploiting diversity gain is the orthogonal *opportunistic relaying* [4], which selects the best among all relays that overheard the source's packet, based on *instantaneous* channel feedback. In Sec. IV, we show that DAC can serve as a complement to opportunistic relaying. By allowing two relays, it provides redundancy across independently faded packets, thus further improving the link reliability.

The feasibility of allowing concurrent transmissions has also been explored in distributed beamforming [12]. Beamforming protocols synchronize the relays, such that their signals can combine coherently at the receiver. However, they require strict frequency, phase, and time synchronization among distributed transmitters, which remains an open challenge [12], due to the limited time resolution at wireless nodes, and the variation of wireless channels. In [13], transmit beamforming is directly tested on 802.11 wireless cards. However, the overlapping signals are not guaranteed to combine coherently due to lack of explicit synchronization. Therefore, both gains and losses are observed when compared with single-packet transmission.

The advent of high-performance software radios has inspired signal-processing-based solutions to overcome the deficiency of the CSMA/CA based 802.11 MAC. For example, the ZigZag protocol [14] overcomes the hidden terminal problem in WLANs by identifying repeated collisions of two hidden transmitters. It then treats each collided packet as a sum over two packets. The two original packets are recovered from two known sums, similar to solving a linear system of equations. DAC' collision resolution PHY is similar to ZigZag, but aims to resolve packets from a single collision with *sample level estimation and cancellation*. DAC aims at improving the end-to-end throughput and delay performance of lossy wireless mesh networks, where it exploits cooperative diversity, based on cut-through relaying and optimal relay selection.

III. Collision Resolution PHY in DAC

The core component of DAC PHY lies in the signal processing module at the receiver, which can decode two overlapping packets carrying the same data. In this section, we focus on the design and implementation of this customized receiver module. Note that in a separate paper [15], we have adopted a similar PHY layer based on simulation, but its objective is to achieve the optimal broadcast delay in wireless mesh networks.

A. An Overview of Iterative Collision Resolution

Suppose in the second stage of the basic DAC relaying scheme (Fig. 1(b)), the relay and the source transmit the same packet towards the destination. Due to the randomness introduced by the transmitters' higher-layer operations, the two copies of the same packet are unlikely to be aligned perfectly. The receiver identifies the natural offset between these two

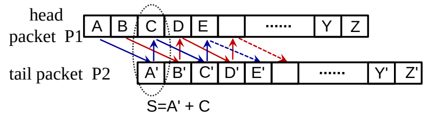

Fig. 2. Iteratively decoding two collided packets carrying the same information, coming from two relays (or one source and one relay), respectively.

copies by detecting a preamble attached in their headers. It first decodes the clean symbols in the offset region, and then iteratively subtracts decoded symbols from the collided ones, thereby obtaining the desired symbols.

For instance, in Fig. 2, two packets (named *head packet* P1 and *tail packet* P2, respectively, according to their arrival order) overlap at the receiver. We first decode the clean symbols A and B in P1. Symbol C is corrupted as it collides with A' in P2, resulting in a combined symbol S. To recover C, we note that A' and A carry the same bit, but the analog forms are different due to independent channel distortions. Therefore, we need to reconstruct an image of A' by emulating the channel distortion over the corresponding bit already known via A.

After reconstruction, we subtract the emulated A' from S, obtaining a decision symbol for C. Then, we normalize the decision symbol using the channel estimation for P1, and use a slicer to decide if the bit in C is 0 or 1. For BPSK, the slicer outputs 0 if the normalized decision symbol has a negative real part, and 1 otherwise. The decoded bit in C is then used to reconstruct C' and decode E. This process iterates until the end of the packet. The iteration for other collided symbols proceeds in a similar way.

B. DAC Transceiver Design

The transmitter module in DAC (Fig. 3) is similar to legacy 802.11b, except that it adds a *DAC preamble* that assists packet detection. The transmitter maps a digital bit to a symbol according to a complex constellation (“1” and “0” are mapped to 1 and -1, respectively). The symbol then passes through a root raised cosine (RRC) filter, which interpolates the symbol into I samples (we adopt a typical value $I = 8$) to alleviate inter-symbol interference. The RRC shaped symbol is the final output from the transmitter.

The receiver module in DAC is also illustrated in Fig. 3. In the normal case of decoding a single head packet, the receiver acts like a typical 802.11b receiver. Upon detecting a tail packet immersed in a head packet, the receiver identifies the exact start of the tail packet, rolls back to its first symbol, and starts the iterative cancellation algorithm. The receiver needs to replay the bit-to-samples transformation at the transmitter, as well as the channel distortion, when reconstructing a symbol in the tail packet. The channel distortion, including amplitude attenuation, phase shift, frequency offset, and timing offset, must be estimated and updated dynamically, since channel parameters vary during the decoding procedure, and the estimation error can accumulate, eventually corrupting the entire packet.

The main challenge in implementing DAC lies in identifying the exact offset between the two packets, and remodeling the symbols in the tail packet based on channel parameter estimation. Unlike interference cancellation [16], we must deal with the common case where collided packets have comparable RSS. Otherwise, the weak packet may be captured and offers no diversity gain. Unlike the symbol cancellation algorithm in

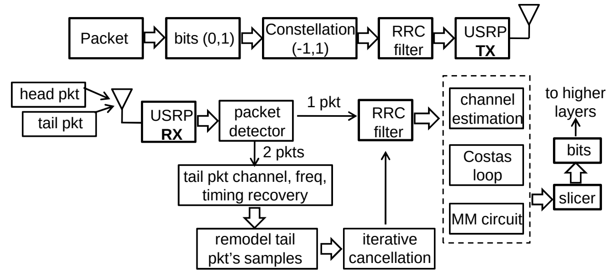

Fig. 3. Flow-chart for DAC transmitter (upper) and receiver (lower).

ZigZag [14], the channel parameters must be estimated in a single collision. To obtain accurate estimation and reconstruction of the symbols, we extensively use *sample-level* correlation, remodeling, and cancellation, as discussed below.

C. Packet Detection and Offset Estimation

The original 802.11b PHY detects the start of a packet by identifying a sequence of known bits from the slicer output. In DAC, we need to detect the presence of one or more packets before feeding the symbols into the slicer. This is achieved by using a combination of *energy* and *feature* detection.

Energy detection estimates a packet's arrival by locating a burst in the magnitude and phase of the received symbol. According to our experiment, a data symbol typically has at least 8dB SNR in order to be decoded error-free. Therefore, it is easy to identify the first symbol of the head packet. When the tail packet arrives and overlaps with the head packet, their corresponding complex samples add up. The magnitude and phase of the resulting symbol thus deviates from the previous symbols, which are relatively stable. DAC uses this deviation as a hint for packet collision.

Energy detection can provide a symbol-level offset estimation, whereas DAC necessitates sample-level estimation accuracy, since the overlapping symbols do not align perfectly. In addition, energy detection's false positive rate increases when ambient noise raises the RSS variation. Therefore, we combine it with feature detection to reduce false positives. Specifically, we correlate the raw decoded symbols with a 256-bit known preamble to confirm the packet arrival event. We use *differential correlation* (*i.e.*, correlating the phase difference of adjacent symbols with the known difference obtained from the preamble) in order to cancel out the transmitter/receiver frequency offset. The correlator outputs a peak whenever a packet arrives. The threshold configuration for peak detection is similar to [14]. Note that the correlation peak is 256 bits behind the first symbol, and therefore DAC maintains a circular buffer storing the latest 256 symbols and their samples, and rollbacks to the first symbol before cancellation.

The energy and feature detection confirms the packet arrival and indicates the symbol-level offset. The exact sample-level collision position is then identified by correlating the *samples* near the beginning of the tail packet with the known samples in the first 16 bits of the preamble (hence 128 known samples in total). The position where the maximum correlation magnitude occurs indicates the start of useful samples. To isolate channel distortion from transceiver distortion, the known samples are obtained offline from the output of a transmit filter.

D. Channel Estimation

We use the collision-free symbols in the beginning of the head packet to estimate its channel. The beginning of the tail packet is immersed in strong noise (*i.e.*, the signals in the head packet), and hence, a direct estimation is severely biased. Unlike prior signal cancellation algorithms [14], [16] that exploit signal capture or repeated collisions, we obtain coarse estimation of the tail packet by correlating and cancelling the known preamble, and then refine the estimation on-the-fly.

1) Amplitude and phase distortion: A coarse estimation of the channel can be obtained via sample level correlation. Suppose the known samples are $x(t), \forall t \in [1, K_s]$ ($K_s = 128$, as discussed above), then the received complex samples after channel distortion should be: $y(t) = Ax(t)e^{j\theta + j2\pi\Delta f t} + n(t)$, where $n(t)$ is the noise process; A and θ are the channel amplitude and phase distortion; Δf is the frequency offset between the transmitter and the receiver. After correlation, we get $Y = A\sum_{t=1}^{K_s}[x(t)e^{j\theta + j2\pi\Delta f t} + n(t)]x(t)$. The phase error due to frequency offset is typically on the order of 10^{-4} rad per sample, and thus, its accumulating effect over the K_s samples is negligible. Further, the ambient noise plus the random samples from the head packet can partly cancel out, resulting in $\sum_{t=1}^{K_s} x^2(t) \gg \sum_{t=1}^{K_s} x(t)n(t)$. Therefore, we approximate the complex channel distortion as $C_d = Y(\sum_{t=1}^{K_s} x^2(t))^{-1}$.

2) Frequency offset estimation: We use the Costas loop [17] to estimate the residual frequency error in the received baseband signals, which is also the frequency offset between the transmitter and the receiver. Costas loop calculates the phase change between two adjacent symbols, and then updates the frequency error via first-order differentiation: $\delta f = \delta f + \omega \cdot (p(t+1) - p(t))$, where $p(t)$ is the symbol phase at time t, and ω is an update parameter, typically set on the order of 10^{-5}.

3) Timing recovery: Ideally, a receiver should align its sampling time with the transmitter to achieve maximum SNR. In practice, the sampling time may deviate from the peak position of the RRC-shaped sample envelop, reducing the effective SNR. A widely-adopted method to correct for sampling offset is the MM circuit [18], which uses a nonlinear hill-climbing algorithm to tune the received signals, such that the sample point is asymptotically aligned with the optimal sampling time.

Remarkably, the MM circuit works only when adjacent symbols have a comparable magnitude, which holds for single-packet decoding. For DAC, the collided symbols have large variations since they consist of symbols from different channels. Hence, we enable the MM circuit timing update only after the symbol cancellation. Further, we need to freeze the MM circuit, *i.e.*, fix its sampling step, whenever an energy burst is detected, indicating a potential collision. We re-enable it for each symbol in the head packet after the corresponding symbol in the tail packet is subtracted out.

4) Transmitter distortion: Beside the channel distortion, the transmitter also pre-processes the signals using the RRC filter to combat multi-path fading. The RRC converts a symbol (1 or -1) into $I = 8$ samples as follows:

$$s_i(t) = x(t-1)F(\frac{3I}{2} + i) + x(t)F(\frac{I}{2} + i), i \in [0, \frac{I}{2})$$
$$s_i(t) = x(t)F(\frac{I}{2} + i) + x(t+1)F(i - \frac{I}{2}), i \in [\frac{I}{2}, I)$$

where $F(i)$ denotes the i-th filter coefficients. At the receiver side, this filtering process is replayed for the tail packet, observing that the digital bits $x(t)$ are already known from prior decoded bits in the head packet.

5) Correcting channel-estimation errors: Recall the initial correlation only provides coarse estimation of the channel gain in the tail packet. During the iterative cancellation procedure, we need to refine the estimation via a simple feedback algorithm. Specifically, we reconstruct an image of symbols in the head packet, and subtract these symbols, to get a refined estimation of symbols in the tail packet. We use the difference between this refined estimation and the original reconstructed image to calculate the channel estimation error, and then update the frequency and time offsets, in a similar manner to the above estimation for the head packet. Observing that the channel gain remains relatively stable for one packet, we use a moving average approach to update the channel amplitude and phase distortion for the tail packet.

One observation from our implementation is that the collision offset identification may also deviate from the exact collision position by one or two samples, especially when SNR is low. We exploit the MM circuit output to compensate for this error. When the MM circuit outputs a sampling step larger than I, it indicates that the collision position is likely to be larger than initially estimated. Our algorithm then increases a credit value by Δt ($0 < \Delta t < 1$). When $\Delta t > 1$, we update the packet offset by 1. A symmetric update procedure is used when the sampling step is smaller than I.

E. Harvest Diversity with Packet Selection

Beside the iterative decoding in the forward direction, DAC can also work backward, starting from the clean symbols in the tail packet (symbol Y' and Z' in Fig. 2), until reaching its beginning, thus obtaining a different estimation of the packet. Since these two packets arrive at the receiver via two independent links, even if one fails in decoding, the other may still be correctly decoded. This is the basis of DAC's diversity gain, and will be rigorously justified in our analysis and experiments.

Note that the diversity gain comes at the expense of additional overhead, including the preamble and the extended reception time due to the packets' offset. However, the preamble length we use is only $K_b = 256$ bits, and the offset time can be easily confined within the duration of tens of bits, with state-of-the-art software radios [19]. In contrast, a typical data payload is around 1K Bytes. Therefore, the additional overhead of DAC is only on the order of 1%.

Also note that the channel estimation, sample remodeling and cancellation only involves linear-time operations. The correlation has $\Theta(n^2)$ complexity (n is the correlation length), but is only needed for around K_b symbols after the energy detection is triggered. In addition, the implementation of DAC is built on BPSK. However, the estimation, reconstruction and cancellation for higher-order modulation schemes, such as M-PSK (M=4, 8, 16, 64), can be realized in a similar way, except that the signal constellation is mapped to different complex vectors [14].

IV. CSMA/CR: DAC-Based MAC

We now introduce a MAC protocol which extends the 802.11-style CSMA, but adds optimal relay selection and *Cut-*

through Relaying to support DAC's *Collision Resolution* PHY, and hence, it is referred to as *CSMA/CR*. CSMA/CR exploits the retransmission diversity in a similar way to typical two-stage relay protocols (Fig. 1). The unique feature is that in the second stage, the relay can transmit immediately after overhearing a retransmission indicator and packet header from the source. The source and relay's packets partly overlap at the receiver, but the collision can be resolved and exploited by the DAC PHY.

One challenge in realizing CSMA/CR is how to select the best relay to maximize the diversity gain. Another problem is compatibility with 802.11 — we aim to add the least overhead and modification to 802.11. To this end, we propose the following solutions.

A. Protocol Operations

The basic MAC-level operations of DAC is illustrated in Fig. 4. Suppose a direct source-destination link is already established by a routing protocol. The source makes a first attempt to transmit the data packet, which can be overheard by both the relay and the destination. If the packet reaches the destination, then CSMA/CR proceeds like CSMA/CA. Upon a failure, *i.e.*, the source receives no ACK from the destination, then it schedules a retransmission and sets a indicator bit in the header of the retransmitted packet. When the packet is emitted, the relay will forward the same packet it overheard, immediately after decoding the retransmission bit and the packet's identity (flow id, sequence number, and transmitter id), which are included in its header. This *cut-through relaying* introduces offset between the arrival time of the source's and relay's retransmission packets, and provides the necessary condition for collision resolution at the receiver.

Due to this asynchrony, the source still senses a busy channel immediately after completing the retransmission. It thereby extends the ACK timeout by the duration between current time and the end of this busy period, which is also the offset between the source and relay's retransmissions. This procedure repeats until the source receives an ACK from the destination. To improve the reliability of ACK, the relay also schedules a cut-through relaying of the ACK packet, when it overhears the header of the ACK packet from the destination.

One remarkable point is that the relay facilitates the retransmission only when it asserts that the source be the only active transmitter within sensing range. This decision is made by looking into the NAV field in 802.11 MAC, which indicates activities in neighboring region, and by looking into the carrier sensing record right before the source's retransmission. If the relay senses a busy channel but cannot decode the identity of the transmitter, then it remains as a normal 802.11 transceiver.

The advantage of CSMA/CR is that the retransmission decision is made solely by the source node, and it needs not know whether the relay has overheard the first transmission. The relay's retransmission decision is also made locally, according to the header it overhears from the source. The idea of allowing relay operation in the middle of source transmission has long been adopted in wireline networks [20]. It has not been adopted in wireless networks, which typically operates on time-orthogonal mode, schedules transmission on a per-packet basis, and allows only one transmitter within the carrier sensing range. However, emerging high performance software radios makes it viable in wireless networks. For example, Sora [19] achieves a scheduling-granularity comparable with the high-rate wireless standards (such as 802.11a) via programmable software and reconfigurable hardware.

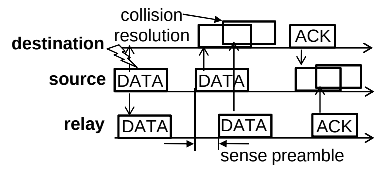

Fig. 4. DAC-based MAC operations.

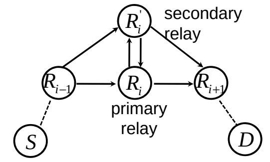

Fig. 5. Improving an existing routing protocol using DAC.

For radio devices incapable of cut-through relaying, we adopt the following scheme built atop the 802.11 RTS/CTS mechanism. Before retransmission, the source sends an RTS packet, piggy-backing the retransmission bit and the packet's identity information in it. Upon overhearing this RTS and the subsequent CTS, both the source and the relay transmit the data packet. In current wireless transceivers, the decision making time is typically on the order of several microseconds [4], this randomness is sufficient to offer several bits' offset between the two transmissions, thus allowing for collision resolution at the PHY. For transceivers with higher time resolution, randomness can be introduced by allowing the source and relay to randomly backoff before starting the retransmission.

B. Optimal Relay Selection

Intuitively, the best relay should have high-quality links to both the source and the destination. With simple analysis on the three-node relaying network (Fig. 1(b)), we can profile the packet delay as follows[1]. Denote p_{sd} as the reception probability of link $S \rightarrow D$, and $q_{sd} = 1 - p_{sd}$ (similar notation is used for link $S \rightarrow R$ and $R \rightarrow D$); and Z, D the packet size and data rate, respectively. Then we have:

Proposition 1 *When relay r is selected, the expected per-packet transmission delay is:*

$$T_r = \frac{Z}{D} \cdot \frac{1 + q_{sd}p_{sr}(1 - q_{sd}q_{rd})^{-1}}{1 - q_{sr}q_{sd}}$$

It follows immediately that the optimal relay that result in minimal delay should satisfy: $R^* = argmin_r \; T_r$.

To simplify the DAC relaying protocol, we have adopted the average link reception rate for selecting a single fixed relays, instead of per-packet SNR feedback [4]. DAC can be extended to complement opportunistic relaying [4], by dynamically selecting the best relay that overheard the source packet. This can be realized by allowing the relay candidates to set a backoff counter that is inversely proportional to their link quality with the destination, in a similar way to [4]. Further investigation of this approach is left as our future work.

DAC's MAC level relay selection algorithm can be further extended to multi-hop cases, to enhance existing routing protocols, as discussed below.

V. Integrating DAC with Routing Protocols

A joint design of DAC relay selection and routing can provide optimal end-to-end delay performance. For simplicity and to emphasize the advantage of non-orthogonal cooperation, however, we restrict our attention to a generic relay-selection

[1]Detailed proofs to the analytical results in this paper are available in [21].

approach that integrates DAC into existing routing protocols, given that the routes had already been selected. Specifically, we use the ETX routing [22] as a basis, and show how to improve its reliability and throughput using DAC relays.

Observing that real-world mesh networks tend to have a majority of links with intermediate quality [6], the ETX protocol adopts a loss-aware link metric, which is the expected number of transmissions needed for successfully delivering a packet on a link. This metric is used to find the shortest path for each data session (a source-destination pair).

Our basic idea is to optimize the ETX route on a per-hop basis. As shown in Fig. 5, suppose a *primary path* ($S \cdots R_{i-1} \rightarrow R_i \rightarrow R_{i+1} \cdots D$) consisting of *primary relays* has been established by ETX. For each primary relay R_i, we decide whether to add a *secondary relay* to it, and select the best secondary relay R_i', according to the potential performance gain in terms of reducing the delay from the previous hop R_{i-1} to the next hop R_{i+1}.

Before analyzing the potential gain, we first introduce the cooperation between the primary and secondary relays. Take the scenario in Fig. 5 as an example. In the normal mode, R_{i-1} makes a first attempt to forward a packet to R_i. Upon successful reception, either R_i or R_i' or both of them can return an ACK. The DAC collision-resolution PHY ensures no ACK collision happens. From the perspective of R_{i-1}, it proceeds to the next packet as long as it can decode an ACK.

If only R_i receives the packet, then it schedules the forwarding following a normal DAC MAC, regarding R_i' as the relay. If both of them receive the packet, then R_i' will perform the cut-through relaying immediately after it senses R_i transmitting the packet it overheard. A primary relay piggybacks the session ID (represented by the source-destination of the path), sequence, and sender ID in the forwarded packet's header, so that it can be recognized in time by the secondary relay. An exception happens when only the secondary relay R_i' receives the packet. R_i' estimates the occurrence of such an event via the absence of R_i's ACK header that is intended for R_{i-1}. In this case, R_i' sends the ACK immediately, and then temporarily takes the position of R_i, serving as the primary forwarder, forming a typical 3-node local relay network together with R_i, following the DAC MAC. The control goes back to the primary relay R_i in the next successful packet transmission from R_{i-1} to R_i.

The above protocol operations allow us to derive a model for analyzing the expected transmission delay, and selecting the optimal relay that incurs the minimum delay. Specifically, we model the progress of a packet as a Markov chain, driven by the transmission, cooperation and forwarding operations among primary and secondary relays. Following notations similar to those in Sec. IV, we have the following proposition.

Proposition 2 *The expected delay in delivering a packet from R_{i-1} to R_{i+1} is:*

$$T = (1 - q_{i-1,i}q_{i-1,i'})^{-1}[ZD^{-1} + p_{i-1,i}q_{i-1,i'}T_{i'} + p_{i-1,i}p_{i-1,i'}T_{i,i'} + q_{i-1,i}p_{i-1,i'}T_i]$$

where $T_{i'} = \frac{Z}{D} \cdot \frac{1+(q_{i',i+1}p_{i,i'})(1-q_{i,i+1}q_{i',i+1})^{-1}}{1-q_{i',i+1}q_{i,i'}}, T_{i,i'} = \frac{Z}{D(1-q_{i,i+1}q_{i',i+1})}, T_i = \frac{Z}{D} \cdot \frac{1+(q_{i,i+1}p_{i',i})(1-q_{i,i+1}q_{i',i+1})^{-1}}{1-q_{i,i+1}q_{i',i}}$. *The best relay should have minimal delay T^* among all secondary relay candidates.*

In the actual implementation of DAC, a relay R_i' is included in the candidate set of secondary relays only if it has a non-zero reception probability with R_{i-1}, R_i and R_{i+1}. Further, based on the above proposition, we can obtain a closed-form expression for the cooperation gain using DAC relaying in terms of throughput improvement: $g^* = \frac{D}{Z} \cdot (p_{i-1,i}^{-1} + p_{i,i+1}^{-1}) \cdot T^{*-1}$. We adopt a secondary relay only if the potential gain g^* is larger than a threshold T_D (set to 1.1 in our design).

To reduce the signaling overhead, we again used the mean link loss rate as a metric for selecting a fixed secondary relay, instead of adjusting the selection for each packet. As shown in existing measurement and routing design [6], [22], the mean link loss rate is relatively stable on an hourly basis, and it can be obtained from the delivery probability of data packets.

The above scheme based on secondary relay selection can be used to improve other routing protocols. For example, we can improve a traditional orthogonal relaying based routing protocol [11] by adding a secondary relay for the existing primary relay. Similar idea can be applied to assist opportunistic routing [23], in which two forwarders who overheard the same packet can be scheduled concurrently, following similar negotiation mechanism in ExOR [23]. The pros and cons of using a DAC based secondary relay will be further clarified in our analysis.

VI. Asymptotic Performance Analysis

In this section, we analyze the performance of DAC, from both the PHY layer and the network level.

A. BER and PER in DAC's Collision Resolution

The iterative collision resolution in Chorus can cause error propagation, due to the correlation between consecutively decoded symbols. For example, in Fig. 2, if symbol A produces an erroneous bit, then the error propagates to A', which affects subsequent symbols such as C. Fortunately, such error propagation stops if the actual bits of A' and C are the same. In this case, after subtracting the error image of A', we obtain a strengthened symbol indicating the correct bit of C. Error propagation also stops when symbol C has a much higher strength than A'. Based on these two intuitions, we prove:

Lemma 1 *The error propagation probability in forward-direction decoding can be characterized as:*

$$P_s \approx Q(\sqrt{2\gamma_1} - 2\sqrt{2\gamma_2})$$

where γ_i denotes the SNR of packet i. The Q-function *is defined as:*$Q(y) = (2\pi)^{-\frac{1}{2}} \int_y^\infty e^{-\frac{x^2}{2}} \mathrm{d}x$. *A symmetric equation holds for backward direction decoding.*

Based on Lemma 1, we further prove that the probability that an error propagates along i bits decays exponentially as i increases, as reflected in the following result.

Lemma 2 *Denote the packet length as L and packet offset as F, then the steady state error length probability can be characterized as:*

$$\pi_i = \pi_0 P_e P_s^{i-1}, \forall i \in (1, G],\ \pi_0 = (1 + P_e \cdot \frac{1 - P_s^G}{1 - P_s})^{-1}$$

where $P_e = Q(\sqrt{2\gamma WD^{-1}})$ is the BER of a non-collided packet with SNR γ, data rate D and signal bandwidth W. $G = \lfloor \frac{L}{F} \rfloor$.

With the above lemmas, we can bound the BER in DAC's iterative collision-resolution algorithm.

Theorem 1 *Let P'_e be the BER in forward-direction decoding in DAC, and P_e be the BER of a single head packet without collision, then $P_e \leq P'_e < 2P_e$.*

A more relevant metric is the packet error rate (PER), which will be used to characterize the gain of DAC over CSMA/CA based non-cooperative schemes. With respect to PER, we have:

Theorem 2 *Let P_h and P_t denote the PER when the head and tail packets are decoded without collision, respectively, then the overall PER in bi-directional collision resolution is $P_v = P_h P_t$.*

Theorem 2 implies that *by allowing two relays to transmit concurrently, PER can be reduced to the PER product of the two independent packets*. It seems counter-intuitive that error propagation does not affect the PER. The reasons for this are twofold. First, since the channel estimation for the tail packet is based on preamble correlation, the estimation error is negligible compared to the bit errors in the head packet caused by channel distortion. Second, we do not use any error correction code, which is beneficial for single-packet decoding. A joint design of error correction and collision resolution may also guarantee better performance for DAC, and this is left as our future work.

B. Throughput Improvement for Multiple Flows

Although DAC improves link reliability via concurrent cooperative relays, it comes at the cost of reducing the multiple access opportunity of competing network flows. This essentially reflects the tradeoff between *diversity gain* and *multiplexing gain* at a network scale, and poses a question: does DAC increase or decrease the total network throughput when multiple flows co-exist? For multihop networks with cooperative relays, the general capacity-scaling law is still an open problem, and existing work has characterized it for special topologies with a single flow [24]. The focus of our analysis here is on characterizing the condition when DAC can outperform non-cooperative routing protocols without calculating the exact capacity bound. We start from a simplified grid topology. Denote Φ_c and Φ_d as the achievable network throughput of a CSMA-based routing protocol, and the corresponding DAC-enhanced routing protocol, then:

Theorem 3 *In a grid network with homogeneous link-reception probability p, $\Phi_d > \Phi_c$ when $p < 0.86$. The throughput gain $\frac{\Phi_d}{\Phi_c}$ decreases monotonically with p.*

Theorem 3 can be extended to a more general case as follows:

Corollary 1 *In an arbitrary network topology with homogeneous link-reception probability p, a sufficient condition for $\Phi_d > \Phi_c$ is $p < 0.64$.*

These analytical results imply that DAC is guaranteed to improve throughput only when the average link quality is sufficiently low. Remarkably, real-world mesh networks tend to have a majority of links with intermediate quality [22] because of channel attenuation, and because optimal rate adaptation schemes may prefer high data-rate links with low quality, than low data-rate links with full reception rate [6].

In a single 802.11 based wireless LAN, at any time, at most one transmitter can be active. Hence, the DAC relaying scheme achieves diversity gain without reducing the channel access opportunity of any transmitter, and it has higher throughput than CSMA, as long as the links have non-zero loss rates.

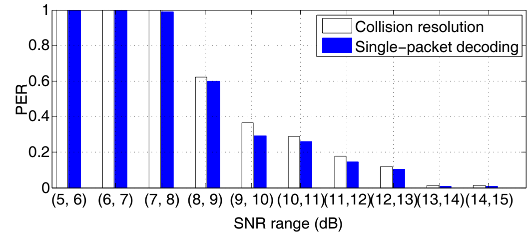

Fig. 6. Comparison between DAC collision resolution and single-packet decoding without collision.

VII. Experimental Evaluation

In this section, we present experimental justification for the feasibility and performance of DAC. We have built a small software radio network to validate the DAC collision-resolution PHY. Based on our experimental and analytical results, we implement the DAC-based MAC and routing protocols in the ns-2 simulator, and evaluate its effectiveness in a large network.

A. DAC Collision-Resolution PHY

We design and prototype the DAC PHY based on the GNURadio/USRP platform [5]. USRP is a software radio transceiver that converts digital symbols into analog waves centered around a carrier frequency within the ISM band. It can also receive analog signals via its RF front-end, and down convert them into the baseband. The baseband digitized raw signals are sent to a general-purpose computer running the Python/C++ based DAC PHY modules built atop the GNURadio library.

The USRP does not yet support MAC operations requiring instantaneous response (*e.g.*, ACK, carrier sensing and cut-through relaying), because of its inefficient user-mode signal processing modules and high communication latency with the computer. Therefore, we focus on the core components of the DAC PHY layer, *i.e.*, the collision-resolution modules. Our testbed environment consists of three USRP nodes, which is used to mimic the typical relay network in Fig. 1(b). The center frequency of USRPs is set to 2.4145GHz, located in between the 802.11 channel 1 and 2. The USRP transmitter's sampling rate is 128 MSamples/s and interpolation rate 32. With BPSK, each digital bit is mapped to one symbol, and each symbol consists of 8 samples after the RRC. Hence, the effective data rate is $\frac{128}{32\times 8} = 0.5$Mbps. Each packet has a 256B payload, which takes the same channel time as a 1KB packet in an actual 2Mbps-mode 802.11b network.

We use two USRPs as the source and relay and allow them to send packets with the same payload. In the common case of DAC relaying, the links of these two concurrent transmitters have comparable strength. Otherwise, the PER reduction is negligible according to Theorem 2 and we can just select a single best relay. In addition, the link with much higher SNR may capture the other, and collision resolution is no longer needed. Therefore, we make coarse adjustment on the SNR between each relay and the shared receiver by varying the transmit power and link distance, so that the difference in mean SNR falls below 1dB[2].

We evaluate the PER when using DAC PHY to resolve two overlapping packets, and compare it with the decoding

[2]For SNR calculation, we note that the signal power is the square of the mean magnitude of non-collided known symbols. Noise power equals the statistical variance of these symbols [17].

probability of a single non-collided packet. Due to channel variations, the SNR value cannot be *precisely* controlled. We thus log the decoded packets, group them according to the received SNR, and calculate the mean packet error rate (PER) for packets falling in the same SNR range (in 1dB unit). The resulting SNR-PER relation is plotted in Fig. 6, where each vertical bar represents 10^4 packets collected over four different time periods. We observed a transition of PER from 1 to 0 when SNR becomes larger than 8dB. Overall, DAC PHY achieves a similar level of PER to the single-packet decoding, which verifies our claim that *single-direction collision resolution does not increase PER, compared to single-packet decoding, and thus bi-directional collision resolution achieves the PER product of the head and tail packet* (Sec. VI-A). Notably, our analysis is developed based on a Gaussian channel model, but the result is consistent with the testbed experiments which are carried out in an office environment with rich multipath fading. This is because the RRC filters partly cancel out the inter-symbol interference, rendering the noise approximately Gaussian. Unfortunately, the PER of DAC is still slightly higher than single-packet decoder, because of the imperfect channel estimation and symbol cancellation.

B. Performance of DAC-Enhanced Routing

1) Experimental setup: In the asymptotic analysis, for tractability, we make simplifications including fixed transmission range and homogeneous loss probability. To evaluate more realistic scenarios, we implement the DAC-enhanced routing protocol (Sec. V) in the ns-2 simulator. The primary path discovery is the same as the ETX routing (which is built atop existing ad-hoc routing protocols) [22]. The secondary relay-selection algorithm runs on each primary relay, which measures the quality of adjacent links, and exchanges link-quality information for those links connecting secondary relay candidates and their previous and next hops. The underlying CSMA/CR protocol is implemented based on the 802.11b MAC in ns-2. We add the DAC header and preamble to each packet, modify the carrier sensing and ACK timeout, so as to support the cut-through relaying, as discussed in Sec. IV and Sec. V.

The simulation runs in a mesh topology with 50 randomly-deployed nodes in a 1km$\times$1km region. We use the log-normal shadowing model with pass-loss exponent 4.0 and shadowing deviation 5.0dB. We replace the ns-2 PHY packet reception model with the analytical model for DAC PHY in Theorem 2, which has been verified in our experiments. The transmit power and reception threshold is configured such that the reception probability is 0.1 at 250m. Overall, this topology has an average link-quality 0.51 and median 0.47, consistent with the measurement from Roofnet [6] which indicates that most links have an intermediate quality.

2) Single-unicast scenario: We evaluate the performance of DAC in comparison with the original ETX routing for two scenarios: single-unicast and multiple-unicast. In the first case, a pair of source-destination nodes are randomly selected to start an end-to-end data session. Since no other competing flows co-exist, we are interested in the average end-to-end packet delay and reliability (indicated by packet-delivery ratio, PDR) for each session. This set of experiments essentially reveal the performance gain of DAC in an unsaturated network. We

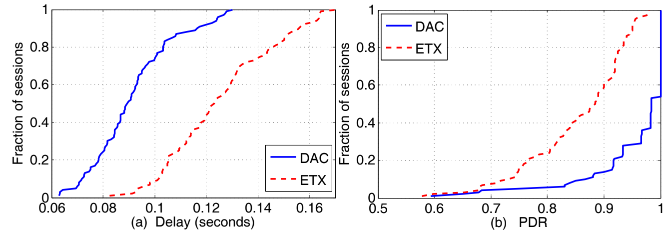

Fig. 7. Distribution of delay and packet-delivery ratio (PDR) for single-unicast sessions.

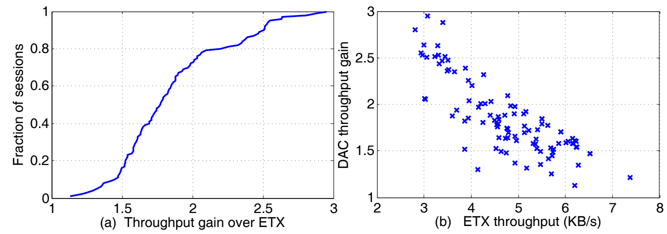

Fig. 8. Throughput gain of DAC over ETX. (a) the CDF plot; (b) the scatter plot, each point corresponding to one session.

evaluate these two metrics over 100 sessions, with packet size 1KB and source rate 0.2Mbps.

The CDF plot in Fig. 7(a) reveals that DAC reduces end-to-end delay for most sessions. The average delay reduction is 27.3%. This improvement comes with much higher PDR, as shown in Fig. 7(b). Since DAC boosts the reception rate of low-quality links with concurrent transmissions from secondary relays, the PDR for a majority of sessions is increased to more than 90%.

We further evaluate the saturated throughput of DAC. We increase the source rate such that the source node's transmit queue remains backlogged. We use *throughput gain* as the metric, defined as the end-to-end throughput of DAC divided by that of ETX. The throughput gain distribution for 100 random sessions is shown in Fig. 8(a). It can be seen that DAC can achieve a 3x throughput improvement over ETX, with an average throughput gain 1.73. In a saturated network, throughput depends on the bottleneck link, *i.e.*, the link with the lowest quality along the selected path. Hence, DAC is most effective for paths with low-quality links. This can be seen from the scatter plot in Fig. 8(b). Obviously, DAC achieves higher throughput gain for those sessions where ETX has below-average throughput. These sessions tend to have links with high loss rate along their paths.

3) Multiple unicast sessions: We proceed to examine DAC's performance when multiple competing flows co-exist, where the fundamental tradeoff between diversity and multiplexing gain becomes an important factor in determining the total network throughput. We evaluate the network throughput as a function of the traffic load. Specifically, we fix the source rate at 10Kbps and increase the total number of sessions. As illustrated in Fig. 9(a), the total network throughput increases with the number of sessions when traffic load is low. In such cases, DAC can have 2x improvement over ETX routing. As the network becomes congested, the non-orthogonal cooperation may sacrifice the channel access time of other concurrent sessions, and therefore, the advantage is less obvious.

While DAC's higher throughput comes from the diversity gain, we need to ensure this advantage does not reduce the

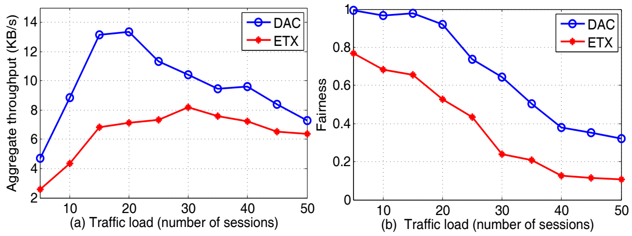

Fig. 9. Total network throughput and fairness vs. traffic load.

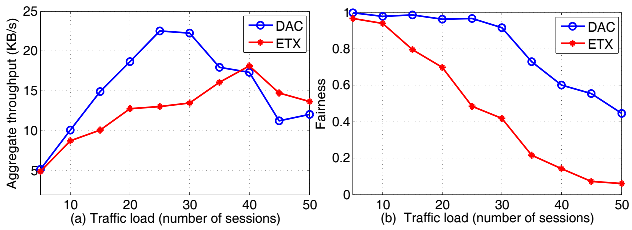

Fig. 10. Total network throughput vs. traffic load in a network with a high reception rate.

fairness among sessions. To evaluate fairness, we use the Jain's fairness index [25] as a metric. A fairness level of 1 indicates all sessions have the same throughput, whereas a close-to-zero fairness indicates some sessions achieve higher throughput by starving others. It can be seen from Fig. 9(b) that DAC always maintains a higher level of fairness. This is because it only rescues the bottleneck links on low-throughput paths (which is reflected in the threshold T_D in designing DAC routing). Overall, both the throughput and fairness are improved by exchanging the multiplexing opportunity of high-throughput sessions for the diversity gain in low-throughput sessions[3]

To make this intuition more concrete, we generate a mesh topology with a majority of high-quality links (the average link quality is 0.826). Fig. 10 shows the resulting network throughput and fairness. Although DAC still maintains a higher level of fairness, much less throughput gain is achieved. This is because ETX tends to select high-quality links whenever available, which are abundant in such a topology. For DAC, the opportunity of exploiting the diversity gain is scarce. Combined with the previous experimental results, this signifies the generality of the analysis in Theorem 3 , *i.e.*, as a non-orthogonal relaying scheme, DAC guarantees throughput gain for networks with intermediate link quality, such as the unplanned mesh network Roofnet [6].

VIII. Conclusion

In this paper, we have introduced DAC, a practical non-orthogonal approach to cooperative relaying without the symbol-level synchronization constraint. The key idea behind DAC is that two partially-overlapping packets carrying the same information from different relays can be decoded independently by using an iterative collision-resolution algorithm at the PHY layer. We provide theoretical and experimental results that demonstrate the decoding probability of DAC PHY and confirm the potential gain of using non-orthogonal relays. We further design a simple MAC protocol to exploit the benefit of DAC decoding, and a generic approach that incorporates DAC relaying into existing routing protocols. Using network-level simulation in ns-2, we show that DAC can improve the network performance in terms of throughput, delay and fairness, especially for lossy wireless mesh networks. As non-orthogonal relaying has fundamental advantage over traditional orthogonal relays [3], DAC marks an effective step towards exploiting the potential of non-orthogonal cooperative communications.

[3]Note that the traffic load higher than 40 sessions is less relevant since the fairness level is low, and most sessions are starved.

References

[1] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
[2] J. N. Laneman and G. W. Wornell, "Distributed Space-Time Coded Protocols for Exploiting Cooperative Diversity in Wireless Networks," *IEEE Trans. on Information Theory*, vol. 49, no. 10, 2003.
[3] K. Azarian, H. E. Gamal, and P. Schniter, "On the Achievable Diversity-Multiplexing Tradeoff in Half-duplex Cooperative Channels," *IEEE Trans. on Information Theory*, vol. 51, no. 12, 2005.
[4] A. Bletsas, A. Khisti, S. Member, and D. P. Reed, "A Simple Cooperative Diversity Method Based on Network Path Selection," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 3, 2006.
[5] The GNU Software Radio, http://gnuradio.org/trac/wiki.
[6] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and Evaluation of an Unplanned 802.11b Mesh Network," in *Proc. of ACM MobiCom*, 2005.
[7] G. Kramer, I. Maric, and R. D. Yates, "Cooperative Communications," *Foundations and Trends in Networking*, vol. 1, no. 3, 2006.
[8] S. Wei, "Diversity Multiplexing Tradeoff of Asynchronous Cooperative Diversity in Wireless Networks," *IEEE Trans. on Information Theory*, vol. 53, no. 11, 2007.
[9] Y. Li and X. Xia, "A Family of Distributed Space-Time Trellis Codes with Asynchronous Cooperative Diversity," in *Proc. of IEEE IPSN*, 2005.
[10] G. Jakllari, S. Krishnamurthy, M. Faloutsos, P. Krishnamurthy, and O. Ercetin, "A Framework for Distributed Spatio-Temporal Communications in Mobile Ad Hoc Networks," in *Proc. of IEEE INFOCOM*, 2006.
[11] S. K and R. Sivakumar, "Cooperating with Smartness: Using Heterogeneous Smart Antennas in Ad-Hoc Networks," in *Proc. of IEEE INFOCOM*, 2007.
[12] R. Mudumbai, D. Brown, U. Madhow, and H. Poor, "Distributed Transmit Beamforming: Challenges and Recent Progress," *IEEE Communications Magazine*, vol. 47, no. 2, 2009.
[13] M. Kurth, A. Zubow, and J. P. Redlich, "Cooperative Opportunistic Routing Using Transmit Diversity in Wireless Mesh Networks," in *Proc. of IEEE INFOCOM*, 2008.
[14] S. Gollakota and D. Katabi, "ZigZag Decoding: Combating Hidden Terminals in Wireless Networks," in *Proc. of ACM SIGCOMM*, 2008.
[15] X. Zhang and K. G. Shin, "Chorus: Collision Resolution for Efficient Wireless Broadcast," in *Proc. of IEEE INFOCOM*, 2010.
[16] D. Halperin, T. Anderson, and D. Wetherall, "Taking the Sting Out of Carrier Sense: Interference Cancellation for Wireless LANs," in *Proc. of ACM MobiCom*, 2008.
[17] B. Sklar, *Digital Communications: Fundamentals and Applications*. Prentice Hall, 2001.
[18] G. R. Danesfahani and T. G. Jeans, "Optimisation of Modified Mueller and Muller Algorithm," *IEEE Electronics Letters*, vol. 31, no. 13, 1995.
[19] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker, "Sora: High Performance Software Radio Using General Purpose Multi-core Processors," in *Proc. of NSDI*, 2009.
[20] M. Gerla, P. Palnati, and S. Walton, "Multicasting Protocols for High-Speed, Wormhole-Routing Local Area Networks," in *Proc. of ACM SIGCOMM*, 1996.
[21] X. Zhang and K. G. Shin, "DAC: Distributed Asynchronous Cooperation for Wireless Relay Networks," preprint, available at: http://www.eecs.umich.edu/ xyzhang/DAC-preprint.pdf.
[22] D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," in *Proc. of ACM MobiCom*, 2003.
[23] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-hop Routing for Wireless Networks," in *Proc. of ACM SIGCOMM*, 2005.
[24] K. Sreeram, S. Birenjith, and P. V. Kumar, "DMT of multi-hop cooperative networks-Part II: Layered and multi-antenna networks," in *Proc. of IEEE ISIT*, 2008.
[25] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination For Resource Allocation in Shared Computer Systems," DEC Research, Tech. Rep. TR-301, September 1984.