Cooperation Without Synchronization: Practical Cooperative Relaying for Wireless Networks

Xinyu Zhang and Kang G. Shin

Abstract—Cooperative relay 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 medium access control (MAC) protocol, and a generic approach to integration of 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 intermediate link quality. Our analytical results have also been confirmed via network-level simulation with ns-2.

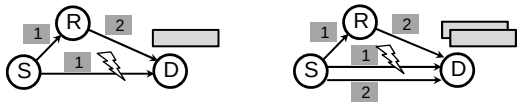
Index Terms—Relay network, cooperative communications, asynchronous cooperative relaying, cross-layer design, diversity-multiplexing tradeoff.

1 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 fails. Such a two-stage cooperative relay protocol establishes a virtual antenna array among multiple distributed transmitters, each with a single antenna, in order to boost the capacity of the link from the source to the destination.

Orthogonal space-time codes [2], originally designed for point-to-point MIMO (multiple-input-multiple-output) 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 information-theoretic approaches, perfect time synchronization among relays is assumed. However, unlike point-to-point MIMO links, cooperative communication, by its nature, occurs asynchronously as there is no global clock shared among the relays. The randomness resulting from 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 offset the advantage of synchronous cooperative communications [4].

• *The work reported in this paper was supported in part by the NSF under grants CNS-1160775, CNS-1317411 and CNS-1318292.* *Xinyu Zhang is with the Department of Electrical and Computer Engineering, University of Wisconsin-Madison. Kang G. Shin is with the Department of Electrical Engineering and Computer Science, The University of Michigan. The authors' email addresses are: xyzhang@ece.wisc.edu, kgshin@eecs.umich.edu*



(a) Orthogonal cooperative relaying (b) DAC, an asynchronous, non-orthogonal relay protocol

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

In this paper, we overcome 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 nodes (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 requiring additional channel access. Hence, DAC improves the link reliability by exploiting spatial diversity via 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 the symbol positions where different bits collide, by re-modeling the known bits and canceling them from collided symbols.

To realize the above idea, 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. 3. Our experimentation on a small relay network testbed shows 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 made by the source.

We also 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 the expanded interference region. Our analysis shows that DAC improves network throughput when the link loss rate is below a certain threshold, which can be accurately 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 inherent limitation of the 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 protocols based on it. Our simulation results show that DAC can significantly improve the throughput and delay performance of existing loss-aware routing protocols. Hence, DAC has potential for non-orthogonal cooperation in wireless relay networks.

Our main contributions can be summarized as follows.

- Design and implementation of an asynchronous, non-orthogonal relaying scheme and its evaluation on an actual software ratio platform. We have characterized the PHY-layer BER and PER analytically.
- Design of a MAC protocol that exploits distributed asynchronous relays with minimal signalling overhead and maximal transparency to the 802.11 MAC.
- Proposal of 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. 2 discusses the work related to wireless relay networks. Sec. 3 describes the design and implementation of the DAC PHY. The DAC-based MAC and routing protocols are presented in Sec. 4 and Sec. 5, respectively. Sec. 6 analyzes the BER, PER, and network-level asymptotic performance for DAC. Further simulation experiments are presented in Sec. 7 to validate DAC's performance. Finally, Sec. 8 concludes the paper.

2 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 explored widely to improve the performance of relay networks (see [7] for a survey). One important progress is attributed to Azarian *et al.* [3] who showed non-orthogonal cooperation schemes to 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 a sub-symbol level, but assumed known and controllable time offsets between relays. Wang *et al.* [10] surveyed recent research along this line of work. Bletsas *et al.* [11] also studied the PHY-layer realization of beamforming mechanisms that eliminate the need for perfect *carrier phase synchronization*. However, they still need *time synchronization*, with a precision of orders of magnitude smaller than symbol period. Such a timing requirement is still quite challenging to meet for distributed transmitters, but is not needed in DAC. Zhang *et al.* [12] demonstrated the feasibility of implementing a cooperative communication scheme on a USRP testbed, by synchronizing multiple nodes with GPS clocks. Space-time coded cooperation for OFDM communication systems was proposed in [13], which leverages the cyclic prefix to tolerate sub-symbol level timing offset between cooperative relays. Rahul *et al.* [14] further implemented the scheme in [13]. With a custom-built FPGA platform, they were able to achieve synchronization accuracy of 50ns and demonstrate a substantial diversity gain over non-cooperative communications. DAC's diversity gain is incomparable with these synchronized schemes, and it only allows for two concurrent relays. However, to the best of our knowledge, it is the first non-orthogonal relaying protocol without any symbol-level timing constraint.

Recently, the higher-layer implication of cooperative relaying has been studied. Jakllari *et al.* [15] directly applied the synchronized space-time code to establish virtual MISO links for routing. Sundaresan *et al.* [16] showed that a 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. 4, we show that DAC can complement opportunistic relaying. By allowing two relays, it provides redundancy across independently-faded packets, thus improving the link reliability.

The feasibility of allowing concurrent transmissions has also been explored in distributed beamforming [17]. Beamforming protocols synchronize the relays, such that their signals can combine coherently at the receiver. However, they require strict frequency, phase, and time synchronization at the symbol level, among distributed transmitters, which remains an open problem [17], due to the limited time resolution at wireless nodes, and the variation of wireless channels. In [18], transmit beamforming is directly tested on 802.11 wireless cards. However, the overlapping signals are not guaranteed to be combined coherently due to the lack of explicit synchronization. Therefore, both gains and losses are observed when compared with single-packet transmissions.

The advent of high-performance software radios has prompted signal-processing-based solutions to overcome the deficiency of the CSMA/CA-based 802.11 MAC. For example, the ZigZag protocol [19] 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, similarly to solving a linear system of equations. DAC's collision resolution PHY is similar to ZigZag, but aims to resolve packets from a single collision with *sample-level estimation and cancellation*. DAC aims to improve 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.

3 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. Here we focus on the design and implementation of this customized receiver module.

3.1 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 probability that the two versions of the packet being time-aligned perfectly is very low. The receiver identifies the natural offset between these two packet 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 symbols A' and A carry the same bit, but their analog forms are different due to the independent channel distortion. Therefore, we need to reconstruct an image of A' by emulating the channel distortion over the corresponding bit that is already known from A.

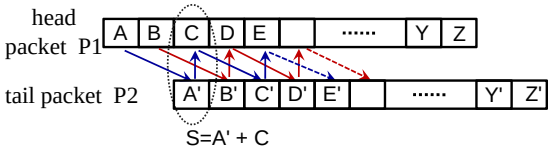


Fig. 2. Iteratively decoding two collided packets carrying the same information, coming from two relays (or one source and one relay), respectively.

After reconstructing the image of A', we subtract the emulated A' from S, thus 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. Likewise, the iteration for other collided symbols proceeds.

The above description assumes the head and tail packets are aligned at symbol level. In practice, each symbol contains multiple samples (*e.g.*, 11 in 802.11b). Since the corresponding transmitters are asynchronous, there is no guarantee that the head/tail symbols align perfectly with each other. However, DAC's iterative decoding algorithm still works under such misalignment. In effect, the actual decoding and channel estimation algorithm runs at sample-level (Sec. 3.2 and Sec. 3.4), thus a misaligned symbol can be reconstructd from samples of two adjacent symbols that were already decoded.

3.2 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 dynami-

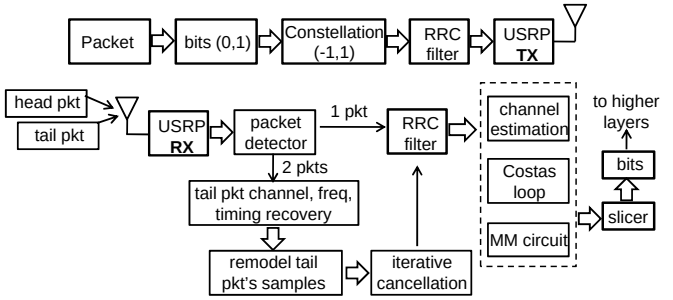


Fig. 3. Flow-chart for DAC transmitter (upper) and receiver (lower).

cally, 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 estimation. Unlike interference cancellation [20], 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 ZigZag [19], 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 next.

3.3 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, while 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 that in [19]. 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 rolls back to the first symbol before cancellation.

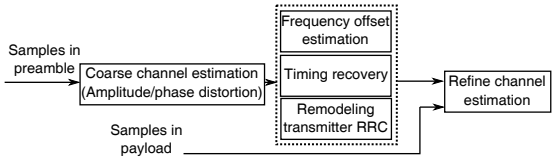


Fig. 4. Channel-estimation procedure for the tail packet that is immersed in the signals of the head packet.

The energy and feature detections confirm the packet arrival and indicate the symbol-level offset. The position of exact sample-level collision 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.

3.4 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 greatly biased. To overcome this problem, we obtain coarse estimation of the tail packet by correlating and canceling the known preamble, and then refining the estimation on-the-fly. Fig. 4 illustrates the flow of operations for channel estimation and below we describe its components in detail.

3.4.1 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[1], and thus, its accumulating effect over 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 $C_d = Y(\sum_{t=1}^{K_s} x^2(t))^{-1}$.

1. This number was obtained from our experiments with a USRP node, which has a frequency stability (around 25ppm) comparable to a typical WiFi chipset (*e.g.*, Atheros 802.11 chipset [21])

3.4.2 Frequency offset estimation

We use the Costas loop [22] to estimate the residual frequency error in the received baseband signals, which is also the frequency offset between the transmitter and the receiver. The 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}.

3.4.3 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 [23], 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.

Note that 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. We also 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.

3.4.4 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:

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

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.

3.4.5 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 using 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 similarly 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.

3.5 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 evaluated 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 [24]. 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 re-modeling 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 likewise, except that the signal constellation is mapped to different complex vectors [19].

3.6 Discussion

Alternative decoding mechanisms. DAC's PHY-layer mechanism can be considered as a zero-forcing (ZF) interference cancellation, which re-models the decoded symbols and eliminates their interference to the symbols to be decoded. DAC's performance can be enhanced by advanced signal processing mechanisms like minimum mean square error (MMSE) decoding and by integrating with error control codes. Such an extension, however, is beyond the scope of this paper.

4 CSMA/CR: DAC-BASED MAC

We now introduce a MAC protocol that 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 similarly to typical two-stage relay protocols (Fig. 1). It is, however, different from others in that at the second stage, the relay can transmit immediately after overhearing a retransmission indicator and packet header from the source. The packets from the source and the relay partially 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 would like to introduce the least overhead and modification to 802.11. We propose the following solutions to meet this goal.

4.1 Protocol Operations

The basic MAC-level operations of DAC is illustrated in Fig. 5. Suppose a direct source-destination link has already been established by a routing protocol. The source makes a first attempt to transmit the data packet, which can be heard by both the relay and the destination. If the packet reaches the destination, then CSMA/CR proceeds like normal CSMA/CA. If the source receives no ACK from the destination (*i.e.*, upon failure), it schedules a retransmission and sets an indicator bit in the header of the retransmitted packet. When the packet is retransmitted by the source, 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 an offset between the arrival times of the source's retransmission and the relay's forwarding, and provides the necessary condition for collision resolution at the receiver.

This asynchrony will make the source sense a busy channel immediately after the source's retransmission is completed. 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's retransmission and the relay's forwarding. 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, when it overhears the header of the ACK packet from the destination.

One interesting point is that the relay facilitates the retransmission only when it asserts that the source be the only active transmitter within its sensing range. This decision is made by looking into the NAV field in 802.11 MAC, which indicates activities in a 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 need not know whether the relay has overheard the first transmission. The relay's forwarding decision is also made locally, according to the header it overhears from the source. The idea of allowing relay operation in the middle of the source's transmission has long been adopted in wireline networks [25]. It has not been adopted in wireless networks, which typically operates on time-orthogonal mode, schedules transmissions 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 [24] achieves a scheduling-granularity comparable to the high-rate wireless standards (such as 802.11a) via programmable software and reconfigurable hardware.

For radio devices incapable of cut-through relaying, we adopt the following scheme built atop the 802.11 RTS/CTS mechanism. Before a 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 an offset of several bits 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 the relay to randomly back off before starting a retransmission.

4.2 Optimal Relay Selection

The best relay should have high-quality links to both the source and the destination. Using a simple analysis on the three-node relaying network n Fig. 1(b), we can profile the packet delay as follows. Let p_{sd} be the reception probability of link $S \to D$, and $q_{sd} = 1 - p_{sd}$ (a similar notation is used for link $S \to R$ and $R \to D$); 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:*

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

Proof: The outcome of S's first transmission attempt can be one of three cases.

Case 1. The direct link $S \to D$ succeeds, which happens with probability p_{sd}, and takes $\frac{Z}{D}$ time units to finish this transmission.

Case 2. Only $S \to R$ succeeds, which happens with probability $(1 - p_{sd}) p_{sr}$, and takes $\frac{Z}{D}$. Both S and R will then transmit simultaneously to D using the cut-through

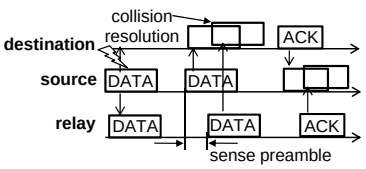


Fig. 5. DAC-based MAC operations.

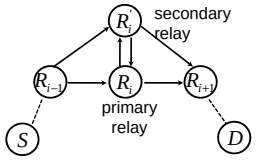


Fig. 6. Improving an existing routing protocol using DAC.

relaying. As will be proven in Theorem 2, the PER of the two combined links $S \to D$ and $R \to D$ equals $1-(1-p_{sd})(1-p_{rd})$. Therefore, the expected remaining time to reach D is $T_r' = \frac{1}{1-(1-p_{sd})(1-p_{rd})}$.

Case 3. Neither $S \to R$ nor $S \to D$ succeeds, which happens with probability $1-(1-p_{sr})(1-p_{sd})$, and wastes $\frac{Z}{D}$. The transmission then starts again from S, taking T_r time in expectation.

Overall, the expected time for a packet to reach D from S is:

$$T_r = p_{sd}\frac{Z}{D} + (1-p_{sd})p_{sr}(\frac{Z}{D} + T_r') + (1-p_{sr})(1-p_{sd})[T_r + \frac{Z}{D}]$$

from which we get:

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

thus completing the proof. □

It follows immediately that the optimal relay incurring minimal delay should satisfy: $R^* = \arg\min_r T_r$.

To simplify the DAC relaying protocol, we have adopted the average link reception rate for selecting a single fixed relay, 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 the quality of their link to the destination, in a way similar to [4]. Further investigation of this approach is a matter of our future work.

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

5 Integrating DAC with Routing Protocols

A joint design of DAC relay selection and routing can achieve optimal end-to-end delay performance. For simplicity and to stress the advantage of non-orthogonal cooperation, however, we restrict our attention to a generic relay-selection approach that integrates DAC with existing routing protocols, given that the routes had already been selected. Specifically, we use the ETX routing [26] as a basis, and show how to improve its reliability and throughput using the DAC mechanism.

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 each 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. 6, suppose a *primary path* ($S \cdots R_{i-1} \to R_i \to R_{i+1} \cdots D$) 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}. Note that if a node already serves as a secondary relay for R_i, then it cannot serve for R_{i+1}. This rule is enforced during the relay-selection procedure through signaling between relays.

Before analyzing the potential gain, we first introduce the cooperation between the primary and secondary relays. Take the scenario in Fig. 6 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. 4, we have the following proposition.

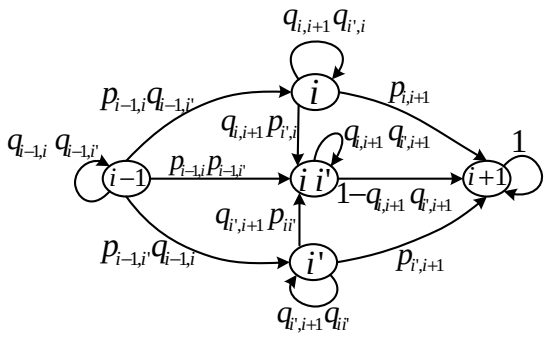


Fig. 7. Modeling the packet propagation in the DAC primary-secondary relay-selection algorithm as a Markov chain.

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.*

Proof: We model the propagation of a data packet as a Markov chain, as shown in Fig. 7. Each state denotes the current holder of the packet. State i represents the fact that R_i has received the packet but R'_i has not. State ii' denotes the fact that both R_i and R'_i have received the packet and the cut-through relaying starts. The expected transmission delay is essentially the first passage time from R_{i-1} to R_{i+1}, denoted as T_{i-1}. Similarly, the expected first passage time from state i, i', ii' to $i+1$ are denoted as $T_i, T_{i'}$ and $T_{ii'}$, respectively. Then, following similar reasoning as in the proof for Proposition 1, we have:

$$T_{i-1} = \frac{Z}{D} + p_{i-1,i}q_{i-1,i'}T_i + q_{i-1,i}p_{i-1,i'}T_{i'} + p_{i-1,i}p_{i-1,i'}T_{ii'} + q_{i-1,i}q_{i-1,i'}T_{i-1}.$$

When the packet is in state i, it may proceed with three possible outcomes. First, R_i succeeds in delivering it to R_{i+1} directly, which happens with probability $p_{i,i+1}$.

Second, the direct delivery fails, but $R_{i'}$ overhears the packet, and consequently the system evolves to state ii'. This happens with probability $q_{i,i+1}$.

If neither happens, then the system remains in state i and repeats the above trials.

Therefore, the expected transmission time from R_i to R_{i+1} is:

$$T_i = \frac{Z}{D}p_{i,i+1} + q_{i,i+1}q_{i'i}T_i + q_{i,i+1}p_{i'i}T_{ii'}. \quad (1)$$

Similarly, for state i', we have:

$$T_{i'} = \frac{Z}{D}p_{i,i+1} + q_{i',i+1}q_{ii'}T_i + q_{i',i+1}p_{ii'}T_{ii'}. \quad (2)$$

For state ii', the expected transmission time is the expectation of a geometric random variable with mean:

$$T_{ii'} = (1 - q_{i,i+1}q_{i',i+1})^{-1} \quad (3)$$

and the joint PER when both $R_i \to R_{i+1}$ and $R'_i \to R_{i+1}$ transmit concurrently is based on Theorem 2.

By solving the above equations, we can obtain a closed-form expression for T_{i-1}, thus completing the proof for Proposition 2. □

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], [26], 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 the traditional routing protocol based on orthogonal relaying [16] by adding a secondary relay for the existing primary relay. A similar idea can be applied to assist opportunistic routing [27], in which two forwarders who overheard the same packet can be scheduled concurrently, following a negotiation mechanism similar to the one in ExOR [27]. The pros and cons of using a DAC-based secondary relay will be discussed further in our analysis.

6 Asymptotic Performance Analysis

In this section, we analyze the performance of DAC, from both PHY and network layers' perspectives.

6.1 BER and PER in DAC's Collision Resolution

The iterative collision resolution in DAC 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 identical. In such a case, after subtracting the error image of A', we obtain a strengthened symbol that indicates the correct bit of C. Error propagation also stops when symbol C has a much greater strength than A'. Based on these two observations, we prove:[2]

2. The proofs for Lemma 1, Lemma 2 and Theorem 1 are similar to the analysis in our previous work [28], [29], and therefore, their details are omitted here.

Lemma 1 *The probability of error propagation 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 of error propagation over i bits decays exponentially as i increases, as reflected in the following result.

Lemma 2 *Let L and F denote the packet length and packet offset, respectively, then the probability of steady-state error length 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 $P_e = Q(\sqrt{2\gamma W D^{-1}})$ 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, respectively, denote the PER when the head and tail packets are decoded without collision, then the overall PER in bi-directional collision resolution is $P_v = P_h P_t$.*

Proof: We start with the forward-direction decoding, *i.e.*, decoding the head packet by subtracting the tail packet from it. We assume no error correction code is used, and therefore, the packet is corrupted once the first bit error occurs.

We represent symbols in the complex number form. Suppose at time t, symbol $\tilde{s}_1(t) = a_1 e^{j\theta_1} x_1(t)$ in P1 collides with $\tilde{s}_2(t) = a_2 e^{j\theta_2} x_2(t)$ in P2. Let v denote the receiver noise, then the received symbol $\tilde{s}(t) = \tilde{s}_1(t) + \tilde{s}_2(t) + v$. If we decode P1 first (forward-direction decoding), then $x_2(t) = x_1(t - F)$. In addition, the channel amplitude a_2 and phase θ_2 can be estimated via correlation, which can achieve high accuracy and introduces negligible noise [19]. Therefore, we can obtain a decision symbol for $x_1(t)$ as: $\tilde{s}(t) - \tilde{s}_2(t) = a_1 e^{j\theta_1} x_1(t) + v$. The resulting SNR level is: $\gamma_1 = \frac{|a_1 e^{j\theta_1}|^2}{2\delta^2} = \frac{P_1}{N}$, which equals the SNR when $s_1(t)$ is decoded independently. With γ_1, we can get the BER for typical fading pattern and noise profile [22]. Suppose the relation between BER and SNR is $P_e = f(\gamma_1)$, then P_h equals the probability that the first L trials succeed for a geometrical random variable with mean P_e, *i.e.*, $P_h = 1 - (1 - P_e)^L$, which is equivalent to the probability of a bit error event in the head packet when it is decoded alone.

Similarly, we can decode the tail packet with PER P_t. With a selective combination, the overall PER equals the probability that both forward and backward decoding fail, which is $P_h P_t$. □

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 yield better performance for DAC, and this is left as our future work.

6.2 Throughput Improvement for Multiple Flows

Although DAC improves link reliability via concurrent cooperative relays, it comes at the cost of reducing the multiple access opportunities of competing network flows. This essentially reflects the tradeoff between *diversity gain* and *multiplexing gain* at the network level, 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 only [30]. Here our analysis focuses 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. Let Φ_c and Φ_d denote the achievable network throughput of a CSMA-based routing protocol and the corresponding DAC-enhanced routing protocol, respectively, 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.*

Proof: Consider the grid network shown in Fig. 8. Let the edge length be e, and assume the transmission range equals e and interference range equals $2e$. Suppose $R_{i-1} \to R_i, R_i \to R_{i+1}$ are two consecutive links used along the path of a flow, which is selected by a CSMA-based routing protocol. Denote $I(R)$ as the interference region of node i. Following the secondary relay selection rules in Sec. 5, it is easy to see that the only secondary relay available is R_i'. Similar to the proof of Proposition 1, we can derive a balance equation for the expected packet

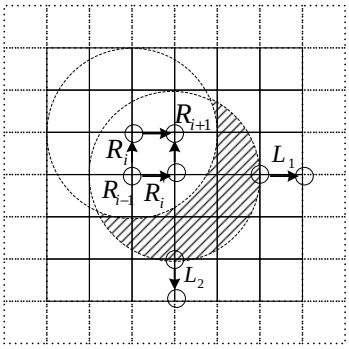


Fig. 8. Grid topology with homogeneous link-reception probability.

delay from R_{i-1} to R_{i+1}:

$$T_{i-1} = 2pq \cdot \frac{1}{p} + \frac{p^2}{1-q^2} + q^2 T_{i-1} + 1$$
$$T_{i-1} = \frac{1+2q}{1-q^2} + \frac{p^2}{(1-q^2)^2} \quad (4)$$

where $q = 1 - p$ and the packet transmission time is normalized to 1. In comparison, a CSMA-based routing entails $\frac{2}{p}$ average delay from R_{i-1} to R_{i+1}.

On the other hand, when R'_i is used by DAC, it transmits concurrently with R_i, expanding the interference region by $I(R_i) - I(R_i) \cap I(R'_i)$ (the shaded region in Fig. 8), compared to a CSMA-based orthogonal scheduling protocol. Within the expanded interference region, at most two transmitters can be scheduled (L_1 and L_2) at the same time without interfering with each other. However, on average, a perfect orthogonal scheduling protocol allocates one fifth of time to each node in this grid (because an optimal schedule achieves the minimal coloring of the nodes, and the minimal coloring of a grid has chromatic number 5 [31]). Further, note that the two nodes' transmissions succeed only with probability p, and the secondary relay of DAC is used with probability p^2 and over $\frac{1}{1-q^2}$ transmission attempts towards R_{i+1}. Therefore, on average, the loss of multiplexing time is $\frac{2}{5} \times p \times \frac{p^2}{1-q^2} = \frac{2p^3}{5(1-q^2)}$.

DAC is guaranteed to reduce the network delay if the diversity gain dominates the multiplexing loss, *i.e.*,

$$f(p) = \frac{2}{p} - \frac{1+2q}{1-q^2} - \frac{p^2}{(1-q^2)^2} - \frac{2p^3}{5(1-q^2)} > 0. \quad (5)$$

We can numerically solve the equation $f(p) = 0$ and get its solution within $(0, 1)$, which equals 0.86. By taking the first-order derivative of $f(p)$, it can be seen that $\frac{\mathrm{d}f(p)}{\mathrm{d}p} > 0, \forall p \in (0, 1)$. Therefore, $f(p)$ is monotonically decreasing within $(0, 1)$. Hence, the diversity gain of DAC always dominates its multiplexing loss when $p < 0.86$, thus completing the proof for Theorem 3. □

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 $p < 0.64$.*

Proof: In an arbitrary topology, DAC selects a secondary relay only if it is connected to the primary relay, the previous hop and the next hop. Therefore, the maximum interference expansion of DAC is $I(R_i) - I(R_i) \cap I(R'_i) < A(R)$, where $A(R)$ is the area of an equilateral triangle with edge length equal to the interference range R. Further, the maximum independent set that can be packed into $I(R'_i)$ is a regular hexagon with edge length R. Since $I(R_i) - A(R) < I(R_i) \cap I(R'_i)$, at least two vertices of this hexagon fall in $I(R_i) \cap I(R'_i)$. Therefore, the interference region expanded by the secondary relay affects at most 4 other vertices. Among the 4 vertices, at most two can transmit concurrently under a CSMA scheduler. Therefore, the average loss of multiplexing time is $2p \times \frac{p^2}{1-q^2} = \frac{2p^3}{(1-q^2)}$, and a sufficient condition for DAC to have performance gain is its diversity gain dominates multiplexing loss, *i.e.*,

$$f(p) = \frac{2}{p} - \frac{1+2q}{1-q^2} - \frac{p^2}{(1-q^2)^2} - \frac{2p^3}{(1-q^2)} > 0 \quad (6)$$

which yields $p < 0.64$ in $(0, 1)$. □

These analytical results imply that DAC is guaranteed to improve throughput only when the average link quality is sufficiently low. Note that real-world mesh networks tend to have a majority of links with intermediate quality [26] because of channel attenuation, and because optimal rate-adaptation schemes may prefer high data-rate links with low quality to 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 a diversity gain without reducing the channel access opportunity of any transmitter, and it has a higher throughput than CSMA, as long as the links have non-zero loss rates.

7 Experimental Evaluation

We now evaluate and demonstrate the feasibility and performance of DAC. We have built a small software radio network to validate the DAC collision-resolution PHY. Based on insights gained from our experimental and analytical results, we implemented the DAC-based MAC and routing protocols in the ns-2 simulator, and evaluated its effectiveness in a large network.

7.1 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.

Our ideal experiments would operate on a software-radio network running DAC-based MAC and PHY. However, the USRP does not yet support MAC operations requiring instantaneous response (*e.g.*, ACK, carrier sensing and cut-through relaying), because of the inefficient user-mode signal processing modules and its 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 digit 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 the 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 a 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 adjustments 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.

The received SNR is measured using the clean parts of the known preamble in the packet. Suppose the received symbol is $y(t)$, the data bit, noise magnitude, and channel gain are $x(t)$, $n(t)$ and A, *i.e.*, $y(t) = Ax(t)+n(t)$. Then, we calculate the noise variance for those symbols representing the bit 1: $\delta_1^2 = var(y^2(t))$. Similarly, we can calculate the noise variance for symbols representing bit -1, denoted as δ_{-1}, and the actual noise variance $\delta = \frac{\delta_1+\delta_{-1}}{2}$ is also the noise power, assuming the actual signals are mixed with AWGN after going through all the decoding modules [22]. A similar method is used to estimate the signal power, where the mean magnitude of bit 1 symbols is calculated by averaging over symbols containing bit 1.

We evaluate PER when DAC PHY is used to resolve two overlapping packets, and compare it with the decoding 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. 9, where each vertical bar represents

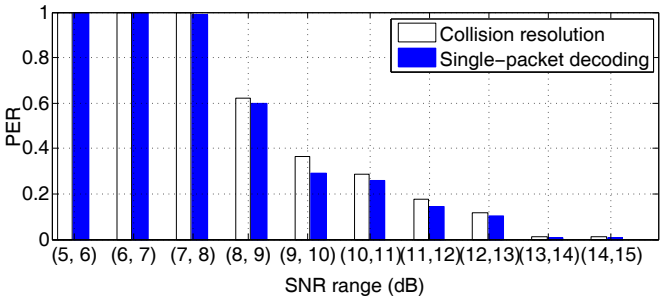


Fig. 9. Comparison between DAC collision resolution and single-packet decoding without collision.

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. DAC PHY achieves similar PER to the single-packet decoding, verifying our claim "*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 packets*" (Sec. 6.1). Our analysis is based on a Gaussian channel model, but the result is consistent with the testbed experiments 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.

7.2 Performance of DAC-Enhanced Routing

7.2.1 Experimental setup

In our asymptotic analysis, we made simplifications, such as fixed transmission range and homogeneous loss probability, for tractability. To evaluate more realistic scenarios, we have implemented the DAC-enhanced routing protocol (Sec. 5) in ns-2. The primary path discovery is the same as the ETX routing (which is built atop existing ad-hoc routing protocols) [26]. 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 by modifying the 802.11b MAC in ns-2. We add the DAC header and preamble to each packet, modify the carrier sensing and transmission timeout, so as to support the direct cut-through relaying, as discussed in Sec. 4 and Sec. 5. As for the PHY-layer simulation, we replace the ns-2 PHY packet-reception model with the analytical model for DAC PHY in Theorem 2, which has been verified with our experiments.

As another benchmark, we have also implemented an *ideal* version of opportunistic routing protocol [27], which will be referred to as *OR-Oracle*. In OR-Oracle, similar to DAC, a primary path is chosen, and two or more secondary relays are then chosen as potential forwarders. Unlike DAC, when multiple potential forwarders receive a packet, only the one with shortest distance (in terms of ETX metric) to destination will be chosen. Otherwise, if a single relay receives the packet, the packet will be forwarded directly. OR-

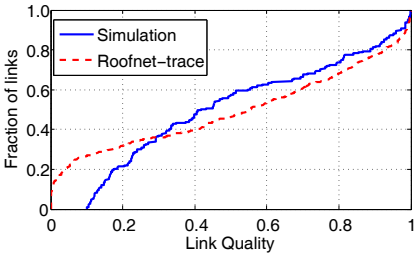


Fig. 10. Distribution of link quality in randomly-generated (simulation) and Roofnet (real) topologies.

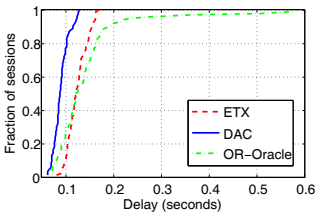


Fig. 11. Distribution of delay for single-unicast sessions.

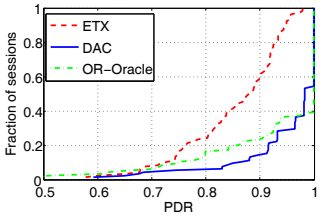


Fig. 12. Distribution of packet-delivery ratio (PDR) for single-unicast sessions.

Oracle assumes the existence of an oracle that can coordinate both relays without incurring any overhead. The coordination in practical opportunistic routing [27] needs to be realized by exchanging messages, which tends to require a significant channel time, especially in lossy networks. The major difference between OR-Oracle and DAC is that the former only allows two relays to forward packets concurrently, thus improving transmit diversity. However, this also implies OR-Oracle results in less interference (higher spatial reuse) between different flows.

Ideally, we would like to use real mesh link-quality traces, such as those from Roofnet [6], as a baseline simulation scenario. However, such traces do not specify the interference relation among nodes. So, we randomly generate a mesh topology with 50 nodes uniformly deployed in a 1km×1km area. We use the log-normal shadowing model with pass-loss exponent 4.0 and shadowing deviation 5.0dB (in consistence with the measurement in [32]). The transmit power and reception thresholds are configured such that the reception probability is 0.1 at 250m. Our topology generator repeatedly produces random topologies until we obtain one in which all nodes are connected, and the node degrees range between 2 and 8. Fig. 10 compares a randomly-generated topology with the traces (the trace ID is 0606062400) from Roofnet [33]. The simulation topology has average link-quality 0.51 and median 0.47, which is roughly consistent with the measurements from Roofnet. Note that we have pruned the links with the reception probability below 0.1. This simplification accounts for the deviation of the link-quality distribution from a real topology. However, since the majority of links have intermediate quality, the relaying protocols would favor routing along long but high-quality paths over short but ultra-low quality links. For instance, in ETX, even a four-hop route results in higher throughput than a single link with PDR 0.1.

7.2.2 Single-unicast scenario

We evaluate the performance of DAC in comparison with the original ETX routing for two scenarios: single and multiple unicasts. In the first case, a pair of source-destination nodes are randomly selected to start an end-to-end data session. Since there are no other competing flows, 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 reveals the performance gain of DAC in an unsaturated network. We evaluate these two metrics over 100 sessions, with packet size 1KB and source rate 0.2Mbps.

The CDF plotted in Fig. 11 shows that DAC reduces end-to-end delay for all the sessions. The average delay reduction is 27.3%. The delay performance of OR-Oracle exhibits a large variance, mainly because the relays are chosen online and may deviate a lot from the shortest path.

Fig. 12 shows the distribution of PDR. With DAC, the PDR for a majority of sessions is increased to more than 90%. It outperforms OR-Oracle for most sessions, because it boosts the reception rate of low-quality links with concurrent transmissions from secondary relays.

We also evaluate the saturated throughput of DAC by increasing the source rate such that the source node's transmit queue remains backlogged. We use *throughput gain*, defined as the end-to-end throughput of DAC (OR-Oracle) divided by that of ETX. The throughput gain distribution for 100 random sessions is plotted in Fig. 13. DAC is shown to be able to achieve a 3x throughput improvement over ETX, with an average throughput gain 1.73. Although OR-Oracle has a comparable delay as ETX (shown in Fig. 11), it still achieves a higher throughput (average gain is 1.42), because of the dominantly high PDR. Note that the absolute throughput gain of OR-Oracle is less than ExOR [27], mainly because ExOR targets batch transmissions, whereas OR-Oracle transmits each packet individually.

In a saturated network, throughput is dictated by the bottleneck link, *i.e.*, the link with the lowest quality along the selected path. Hence, DAC is most effective for paths containing low-quality links. This can be seen from the scatter plot in Fig. 14. Obviously, DAC achieves a higher throughput gain for those sessions where ETX has a below-average throughput. These sessions tend to have links with high loss rate along their paths.

7.2.3 Multiple unicast sessions

We now consider DAC's performance in the presence of multiple competing flows, where the fundamental trade-off between diversity and multiplexing gain becomes an important factor in determining the total network throughput, as discussed in Sec. 6. 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. 15, the total network throughput increases with the number

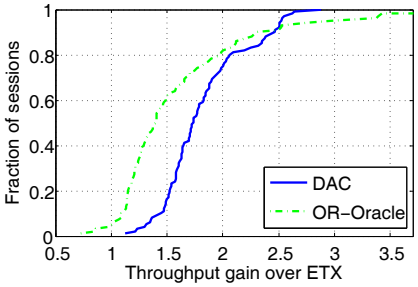


Fig. 13. Throughput gain of DAC and OR-Oracle over ETX.

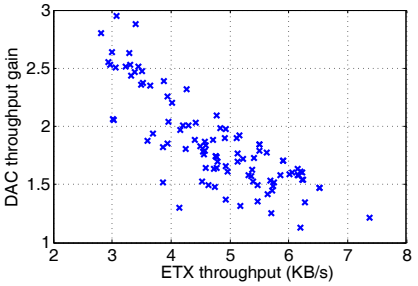


Fig. 14. Throughput gain of DAC over ETX: each point corresponding to one session.

of sessions when traffic load is low. In such cases, DAC can make a 2x improvement over ETX routing. As the network becomes congested, the non-orthogonal cooperation may degrade the channel access time of other concurrent sessions, thus making its advantage less obvious. Since OR-Oracle only employs one relay at each hop, it achieves better spatial reuse and may outperform DAC when the network is congested.

While DAC's higher throughput comes from the diversity gain, we need to ensure this advantage does not reduce the fairness among sessions. To evaluate fairness, we use the Jain's fairness index [34] as a metric. A fairness level of 1 indicates all sessions have the same throughput, whereas a close-to-0 fairness indicates that some sessions achieve a higher throughput by starving others. One can see from Fig. 16 that DAC always maintains a much higher level of fairness than ETX and OR-Oracle. This is because it only relieves the bottleneck links on low-throughput paths (which is reflected in the threshold T_D in designing DAC routing). Overall, compared to ETX, 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] Note that the fairness of OR-Oracle is even lower than ETX because it tends to overload the high-quality links, resulting in local congestion and poor throughput performance, especially for those sessions whose critical link is congested.

In Sec. 6.2 we used a simple model to prove that DAC improves the throughput of multiple flows only when the network is lossy. To make this analysis more concrete, we generate a mesh topology with a majority of high-quality links (the average link quality is 0.826). Figs. 17 and 18 show the resulting network throughput

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

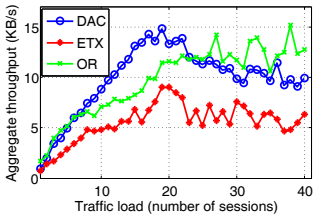


Fig. 15. Total network throughput vs. traffic load.

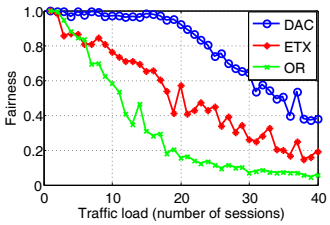


Fig. 16. Fairness vs. traffic load.

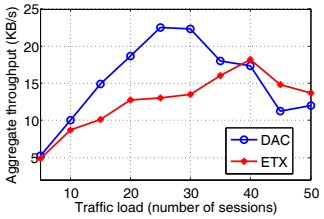


Fig. 17. Throughput vs. traffic load in a network with a high reception rate.

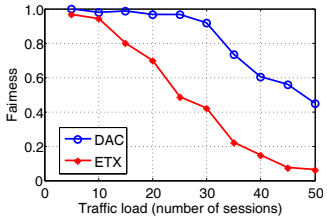


Fig. 18. Fairness vs. traffic load in a network with a high reception rate.

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 if 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 indicates the generality of the analysis in Theorem 3, *i.e.*, as a non-orthogonal relaying scheme, DAC guarantees a throughput gain for networks with intermediate link quality, such as the unplanned mesh network Roofnet [6].

8 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 verify the potential gain of using non-orthogonal relays. We also 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 simulations with 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 is shown to be an effective step towards exploiting the potential of non-orthogonal cooperative communications.

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] H.-M. Wang and X.-G. Xia, "Asynchronous Cooperative Communication Systems: A Survey on Signal Designs," *Science China Information Sciences*, vol. 54, no. 8, 2011.

[11] A. Bletsas, A. Lippman, and J. N. Sahalos, "Simple, Zero-Feedback, Distributed Beamforming with Unsynchronized Carriers," *IEEE J.Sel. A. Commun.*, vol. 28, no. 7, 2010.

[12] J. Zhang, J. Jia, Q. Zhang, and E. M. K. Lo, "Implementation and Evaluation of Cooperative Communication Schemes in Software-Defined Radio Testbed," in *Proc. of IEEE INFOCOM*, 2010.

[13] Z. Li and X.-G. Xia, "A Simple Alamouti Space-Time Transmission Scheme for Asynchronous Cooperative Systems," *IEEE Signal Processing Letters*, vol. 14, no. 11, 2007.

[14] H. Rahul, H. Hassanieh, and D. Katabi, "SourceSync: A Distributed Wireless Architecture for Exploiting Sender Diversity," in *Proc. of ACM SIGCOMM*, 2010.

[15] 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.

[16] K. Sundaresan and R. Sivakumar, "Cooperating with Smartness: Using Heterogeneous Smart Antennas in Ad-Hoc Networks," in *Proc. of IEEE INFOCOM*, 2007.

[17] R. Mudumbai, D. Brown, U. Madhow, and H. Poor, "Distributed Transmit Beamforming: Challenges and Recent Progress," *IEEE Communications Magazine*, vol. 47, no. 2, 2009.

[18] M. Kurth, A. Zubow, and J. P. Redlich, "Cooperative Opportunistic Routing Using Transmit Diversity in Wireless Mesh Networks," in *Proc. of IEEE INFOCOM*, 2008.

[19] S. Gollakota and D. Katabi, "ZigZag Decoding: Combating Hidden Terminals in Wireless Networks," in *Proc. of ACM SIGCOMM*, 2008.

[20] 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.

[21] Atheros Communications., "AR5213 Preliminary Datasheet," 2004.

[22] B. Sklar, *Digital Communications: Fundamentals and Applications*. Prentice Hall, 2001.

[23] G. R. Danesfahani and T. G. Jeans, "Optimisation of Modified Mueller and Muller Algorithm," *IEEE Electronics Letters*, vol. 31, no. 13, 1995.

[24] 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.

[25] M. Gerla, P. Palnati, and S. Walton, "Multicasting Protocols for High-Speed, Wormhole-Routing Local Area Networks," in *Proc. of ACM SIGCOMM*, 1996.

[26] 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.

[27] S. Biswas and R. Morris, "ExOR: Opportunistic Multi-hop Routing for Wireless Networks," in *Proc. of ACM SIGCOMM*, 2005.

[28] X. Zhang and K. G. Shin, "Chorus: Collision Resolution for Efficient Wireless Broadcast," in *Proc. of IEEE INFOCOM*, 2010.

[29] ——, "Delay-Optimal Broadcast for Multihop Wireless Networks Using Self-Interference Cancellation." *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, 2013.

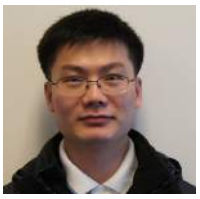
[30] K. Sreeram, S. Birenjith, and P. V. Kumar, "DMT of multi-hop cooperative networks-Part II: Layered and multi-antenna networks," in *Proc. of IEEE ISIT*, 2008.

[31] J. Bondy and U. Murthy, *Graph Theory With Applications*. Elsevier, 1976.

[32] J. Camp, J. Robinson, C. Steger, and E. Knightly, "Measurement Driven Deployment of a Two-tier Urban Mesh Access Network," in *Proc. of ACM MobiSys*, 2006.

[33] "Roofnet traces for SIGCOMM'04," available at: http://pdos.csail.mit.edu/roofnet.

[34] 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.



Xinyu Zhang is an Assistant Professor in the Department of Electrical and Computer Engineering, University of Wisconsin-Madison. He received the B.E. degree in 2005 from Harbin Institute of Technology, China, the M.S. degree in 2007 from the University of Toronto, Canada, and the Ph.D. degree in 2012 from the University of Michigan-Ann Arbor. His research interest lies in designing and implementing protocols that improve the capacity, interoperability and energy efficiency of wireless networks. He is a recipient of ACM MobiCom Best Paper Award in 2011 and NSF CAREER Award in 2014.

Kang G. Shin is the Kevin & Nancy O'Connor Professor of Computer Science in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor. His current research focuses on computing systems and networks as well as on embedded real-time and cyber-physical systems, all with emphasis on timeliness, security, and dependability. He has supervised the completion of 70 PhDs, and authored/coauthored more than 770 technical articles (more than 270 of these are in archival journals), one a textbook and more than 20 patents or invention disclosures, and received numerous best paper awards, including the Best Paper Awards from the 2011 ACM International Conference on Mobile Computing and Networking (MobiCom2011), the 2011 IEEE International Conference on Autonomic Computing, the 2010 and 2000 USENIX Annual Technical Conferences, as well as the 2003 IEEE Communications Society William R. Bennett Prize Paper Award and the 1987 Outstanding IEEE Transactions of Automatic Control Paper Award. He has also received several institutional awards, including the Research Excellence Award in 1989, Outstanding Achievement Award in 1999, Distinguished Faculty Achievement Award in 2001, and Stephen Attwood Award in 2004 from The University of Michigan (the highest honor bestowed to Michigan Engineering faculty); a Distinguished Alumni Award of the College of Engineering, Seoul National University in 2002; 2003 IEEE RTC Technical Achievement Award; and 2006 Ho-Am Prize in Engineering (the highest honor bestowed to Korean-origin engineers). He has chaired several major conferences, including 2009 ACM MobiCom, 2008 IEEE SECON, 2005 ACM/USENIX MobiSys, 2000 IEEE RTAS, and 1987 IEEE RTSS. He is the fellow of both IEEE and ACM, and served on editorial boards, including IEEE TPDS and ACM Transactions on Embedded Systems. He has also served or is serving on numerous government committees, such as the US NSF Cyber-Physical Systems Executive Committee and the Korean Government R&D Strategy Advisory Committee. He has also co-founded a couple of startups.