# Taking the Sting out of Carrier Sense: Interference Cancellation for Wireless LANs

Daniel Halperin
University of Washington

Thomas Anderson
University of Washington

David Wetherall
University of Washington
Intel Research Seattle

## ABSTRACT

A fundamental problem with unmanaged wireless networks is high packet loss rates and poor spatial reuse, especially with bursty traffic typical of normal use. To address these limitations, we explore the notion of interference cancellation for unmanaged networks — the ability for a single receiver to disambiguate and successfully receive simultaneous overlapping transmissions from multiple unsynchronized sources. We describe a practical algorithm for interference cancellation, and implement it for ZigBee using software radios. In this setting, we find that our techniques can reduce packet loss rate and substantially increase spatial reuse. With carrier sense set to prevent concurrent sends, our approach reduces the packet loss rate during collisions from 14% to 8% due to improved handling of hidden terminals. Conversely, disabling carrier sense reduces performance for only 7% of all pairs of links and increases the delivery rate for the median pair of links in our testbed by a factor of 1.8 due to improved spatial reuse.

## Categories and Subject Descriptors

C.2.1 [**Computer Systems Organization**]: Computer-Communication Networks—*Network Architecture and Design*

## General Terms

Algorithms, Design, Experimentation, Performance, Reliability

## Keywords

Interference cancellation, spatial reuse, carrier sense, wireless networks, software radio

## 1. INTRODUCTION

Wireless LANs such as WiFi and ZigBee pose a challenge to system designers aiming to achieve reliable and high performance communication. Partly, this is due to the inherent characteristics of wireless communications, in which performance is heavily dependent on the details of the environment. But wireless LANs also add chaotic traffic patterns, irregular and sometimes dynamic topologies, heterogeneous hardware, devices in separate administrative domains and decentralized control. The result is that traditional techniques for allocating resources, such as TDMA, FDMA, CDMA, cannot readily be used. Instead, wireless LANs rely on carrier sense, in which a device will defer to a transmission already in progress. Nodes send with a minimum of delay when they have data, while avoiding collisions that would potentially cause loss.

Unfortunately, it is well-known that carrier sense does not always work as intended. *Hidden terminals* are nodes that do not sense each other's transmissions, even though they interfere at an intended receiver to prevent successful decoding. *Exposed terminals* are nodes that do sense and defer to each other, even though the intended receivers are located such that both would receive if data was sent simultaneously. Efforts to improve one of these effects often cause the other to become worse. For example, we can raise the carrier sense threshold, increasing spatial reuse, but also increasing loss [15, 25]. Similarly, we can use an RTS/CTS mechanism to eliminate hidden terminals, but this can hurt performance in practice by decreasing spatial reuse. As a result, RTS in 802.11 NICs is typically disabled by default, even though when present, hidden terminals can reduce 802.11 throughput by 40% [21].

In this paper, we explore whether it is possible to sidestep the performance tradeoff between hidden and exposed terminals, by designing receivers to recover multiple simultaneous signals. In this way, hidden terminals are less of a problem, allowing exposed terminals to be more easily tolerated as transmitters can choose a more aggressive sensing threshold. We simplify the design space by restricting ourselves to successive interference cancellation (SIC), also called successive decoding [23, ch. 7]. With successive cancellation, one sender's packet is decoded from a received collision and then the signal is subtracted, allowing the receiver to decode weaker simultaneous signals.

Successive interference cancellation has been shown to improve bandwidth utilization in cellular networks [1]. These deployments, unlike wireless LANs, are designed with centralized control that they exploit to great advantage. Continuous closed-loop communications between cell phones and the towers allows the system to synchronize the clocks of all devices in the system. During association, and adjustable at


*MobiCom'08,* September 14–19, 2008, San Francisco, California, USA.

any time (e.g., during call setup), the towers determine the best transmit power, coding rate, and/or spreading codes to enable the disambiguation of different uplink sends. Time division, in conjunction with synchronized clocks, implies that frames transmission are aligned. And with dedicated spectrum, no devices beyond the control of the centralized infrastructure are present to cause co-channel interference.

In contrast, wireless LANs operate in the chaotic unlicensed bands. Networks in separate administrative domains compete with no centralized control over power or rates, no time or frequency synchronization between devices, and in general employ no techniques to facilitate the disambiguation of distinct transmitters beyond (with mixed success) attempting to prevent concurrent sends with carrier sense.

In this work, we adapt SIC to work in these uncontrolled network scenarios, to show that successive interference cancellation can help reduce the inherent tradeoff post by carrier sense in wireless LANs. To make our discussion concrete, we design and implement a simple SIC receiver for the IEEE 802.15.4 (ZigBee) physical layer. A key idea in our design is that the receiver uses measured properties of the signal to recover and then subtract each packet in turn. Because of the limits of the software radio system we used, we cannot directly measure our implementation against carrier sense, but we use measurements to estimate the relative performance of the different approaches when pairs of links compete to send. With a carrier sense threshold set to eliminate concurrent sends, SIC reduces the packet loss rate from 14% to 8% due to improved handling of hidden terminals. Conversely, disabling carrier sense entirely reduces performance for only 7% of all link pairs, and raises the median combined delivery rate across both links by a factor of 1.8 due to improved spatial reuse.

The rest of the paper is organized as follows. Section 2 provides a quick primer on detection of wireless signals, and explains our design for a SIC receiver. Section 3 presents our software radio implementation of SIC for ZigBee nodes, as well as the two single-packet detectors we use as a baseline. Section 4 explains our experimental setup and methodology. In Section 5 we evaluate SIC performance in our network, and in Section 6 we use these measurements to predict the system effects of using SIC in a real network. Finally, Section 7 explains related work before we conclude in Section 8.

# 2. SUCCESSIVE INTERFERENCE CANCELLATION

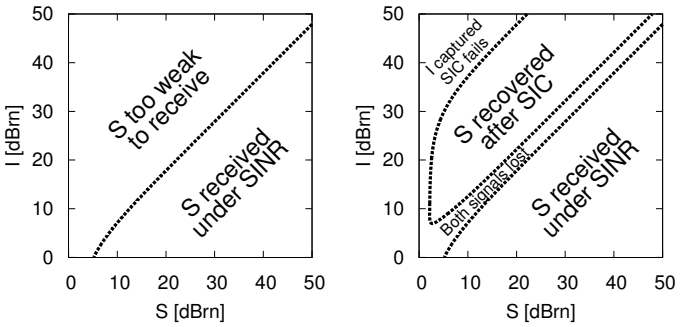
We begin with background on successive interference cancellation (SIC) and how we apply it to the setting of wireless LAN receivers.

## 2.1 Decoding signals — the SINR model

In communications theory, the *signal to interference-and-noise* ratio (SINR) of a received transmission determines a device's ability to recover the data it contains. SINR is defined as the ratio between the signal power $S$[1] of the desired transmission to the combined power of all interfering transmitters $I = \sum_i I_i$ plus noise $N$:

$$\text{SINR}_S = \frac{S}{I + N} \quad (1)$$

[1]We abuse notation slightly and use $S$ both for the signal $S(t)$ and for its signal power




(a) Capacity under the SINR model (b) Capacity with SIC

**Figure 1: These figures show the capacity regions for ZigBee under the SINR model (a) and with a SIC detector (b) that can reduce a captured signal by 30 dB. We assume a Gaussian noise model and a 3 dB inefficiency in the receiver implementation, and plot the conditions for 85% packet reception.**

Reception succeeds when the SINR is large, i.e. the signal strength overcomes both electromagnetic noise and the combined power of all interfering transmitters.

Formally, we model the bit error rate (BER) experienced by a receiver as a function of the SINR. A larger SINR implies a stronger signal and fewer bit errors, and allows for a higher supported data rate. In packet radio, we are more concerned with the packet error rate (PER), calculable directly from the link BER and packet length. For a given upper bound on the PER, we can calculate a threshold $\tau$ such that if the SINR of a signal exceeds $\tau$, then a packet will be received successfully with that bounded error probability.

For the ZigBee PHY we use in this work, the threshold $\tau$ is about 0.5 dB for a PER of 1% [19], incorporating signals gains from redundancy in the ZigBee codewords and the properties of the physical modulation scheme. This result follows directly from the BER curves for the modulation scheme employed by ZigBee under a standard Gaussian noise model. However, this ideal result neglects the realities of implementing practical receivers. Thus Figure 1 (a) shows the capacity region (85% delivery) for ZigBee packets under Gaussian noise, and allowing for a receiver implementation with a 3 dB loss of efficiency. Commercial wireless hardware commonly includes various inefficiencies, trading slightly worse error performance for reduced implementation complexity, less silicon, lower power consumption, and/or cheaper components.

## 2.2 Interference cancellation theory

The fundamental idea behind interference cancellation techniques is that, unlike noise, interfering signals have structure determined by the data that they carry. Interference cancellation techniques exploit this structure to mitigate the harmful effects of interference, improving the effective SINR of a signal and reducing bit errors.

Consider a received signal $R$ containing two additive transmissions $S_1$ and $S_2$ as well as thermal noise:

$$R(t) = S_1(t) + S_2(t) + N(t). \quad (2)$$

In this case, assume that $S_1$ is stronger than $S_2$ such that $S_2$ has insufficient SINR for recovery,

$$\text{SINR}_2 = \frac{S_2}{S_1 + N} < \tau. \quad (3)$$

However, if we exactly knew the signal $S_1$, we could simply subtract it from $R$. This would eliminate its contribution to the denominator of $\text{SINR}_2$ and potentially allow $S_2$'s reception. In practice, however, we instead have an approximate model for $S_1$, $\tilde{S}_1$, which we can subtract from $R$. In the resulting signal, $R - \tilde{S}_1$, the interference power of $S_1$ is reduced to the error in approximation, from which $S_2$ can be decoded if the approximation is sufficiently accurate.

$$\text{SINR}_2[R - \tilde{S}_1] = \frac{S_2}{(S_1 - \tilde{S}_1) + N} > \tau \quad (4)$$

*Successive interference cancellation* (SIC) [23, Ch. 7] is one version of this technique that exploits the well known *capture effect*, a natural consequence of the SINR model in which a strong interfering signal is successfully received during a collision. SIC uses the bits successfully recovered from the signal to generate the approximation that it then subtracts. We show the 85% reception region for SIC with respect to the signal $S$ in Figure 1 (b). Here we assume a SIC implementation that reduces the interfering signal strength by 30 dB, eliminating 99.9% of its energy, and allowing the signal S to be received over a larger region.

## 2.3 SIC receiver processing

At a high level, our SIC receiver performs the following steps: (1) detect a collision, (2) identify and decode the strongest signal, (3) develop a data dependent model of the signal, (4) use the received data and the model to cancel the signal, and (5) iterate to decode the packet(s) that remain. In a cellular implementation of SIC, steps (1)-(3) would be vastly simpler than for wireless LANs. Further, note that in wireless LANs SIC, and collisions in general, raise some MAC level issues that we defer until Section 3.4.

**Detecting a collision.** One way to determine when collisions happen is to determine when packets begin and end. We do this by scanning for sharp changes in the amplitude variation of the incoming signal. When the energy jumps to indicate that a stronger transmission has started, the detector will synchronize with and begin modeling the new, stronger signal. We detect the end of a packet by a corresponding decrease in signal strength.

**Decoding the strongest signal.** In order for SIC to work, we require that one signal has SINR sufficient that it can be received. If this holds, then provided we can detect the beginnings and ends of packets, we can use standard single-packet reception techniques to decode this strongest signal. It is important to note here that many conventional single-packet receivers *do not* re-synchronize on a strong interfering transmission if synchronization has already been achieved for a preceding transmission. A consequence of this is that a receiver that does not implement such re-synchronization can drop an interfering transmission even with a large SINR that is sufficient for decoding. For a conventional detector, this is acceptable if carrier sense is used to suppress competing traffic. For comparison, we provide an implementation of both a resynchronizing and a non-resynchronizing detector.

**Modeling a captured signal.** To model a received transmission, we need to understand the wireless channel. At the transmitter, the signal constructed to carry the data is defined by the standard (e.g., IEEE 802.15.4 [10] for ZigBee), but becomes distorted in various ways. Filtering at both the sender and receiver prevents bandwidth leakage and combats intersymbol interference. Physical propagation introduces signal attenuation — energy decays with distance — and multipath fading, in which multiple copies of the signal interfere after taking different physical paths. And, the crystal oscillator generating the carrier, a radio wave of frequency $f_c$ which carries the transmitted data, is a physical object with imperfections that result in a small offset between $f_c$ at the transmitter and at the receiver.

Rather than explicitly model complex channel effects, we approximate them as an unknown but deterministic function of the data (as we assume distortions to be constant over one packet length) that is implicit in the received signal $R(t)$. In practice, $R(t)$ for one data symbol depends not only on the data symbol transmitted at time $t$, but also the preceding and succeeding symbols due to multipath interference and filters which deliberately blend symbols to lessen spectral usage. Thus to build the model, we average together the received waveforms wherever the packet contains the same window of data. Provided the data is quasi-random, interfering transmissions average out and become noise.

Previous work, including that of the authors [7], built simple explicit channel models by calculating the frequency offset and ignoring other effects. While this approach is easy to implement, we found it limited in accuracy, especially as symbol time decreases and multipath plays a larger role. Further, we found extending the model to account for all channel effects to be overly involved, e.g., there are different fading models (e.g., Rice, Rayleigh, and Nakagami) for different environments (indoor, outdoor, urban)m that would be quite difficult to track in practice.

**Canceling the strong interferer.** The final step in SIC is to use the models and data to cancel out the signal from the strong interferer. To do so, we iterate through the packet data, and align the phase of the model with the received samples. Finally, we subtract the waves, canceling the interfering signal.

# 3. IMPLEMENTATION OF SIC ZIGBEE RECEIVER

In this section we describe our implementation of SIC as well as of two detectors designed for a single-packet environment against which we compared our design. We begin with information on the software radio platform we use (the USRP) and the physical layer we build (ZigBee).

## 3.1 The USRP platform

We implemented the ZigBee physical layer on the Universal Software Radio Peripheral (USRP) [4], a hardware platform for software radio that interoperates with the GNU Radio [5] software library. Simply, the USRP digitizes received waves after converting them down from the carrier frequency $f_c$ (2.425 GHz in our experiments) to baseband (0 Hz) for processing in a general-purpose computer. We implemented SIC and single-packet receiver processing, as described below, using a combination of standard and custom-built GNU Radio processing blocks.

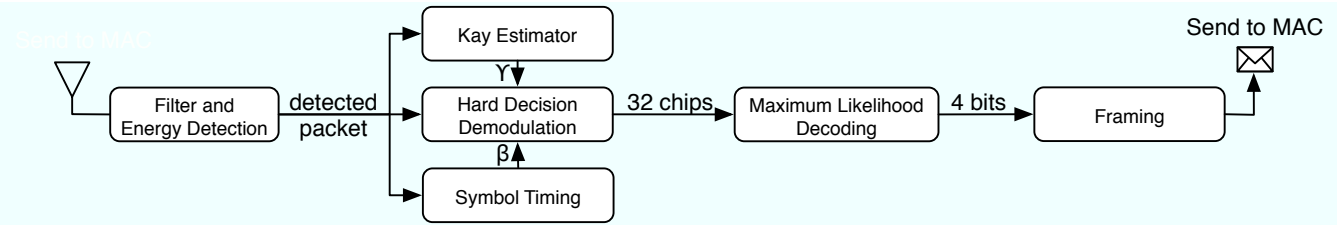



**Figure 2: A block diagram of our conventional, single-packet ZigBee detector. The incoming signal is filtered by a channel select filter to remove noise outside the desired 3 MHz of spectrum. An energy detection block detects bursts with power 2 dB larger than the noise floor and isolates these transmissions for decoding. Next, a Kay estimator calculates the frequency offset $\gamma$ and then symbol timing $\beta$ — the offset in the $4\times$ oversampled signal that is best to demodulate the data — is recovered. The demodulation block uses $\beta$ and $\gamma$ to make hard decisions about the incoming chip sequence, which is then framed into codewords and using maximum likelihood estimation decoded as bits. Finally, the bits are framed into packets and passed up the network stack.**

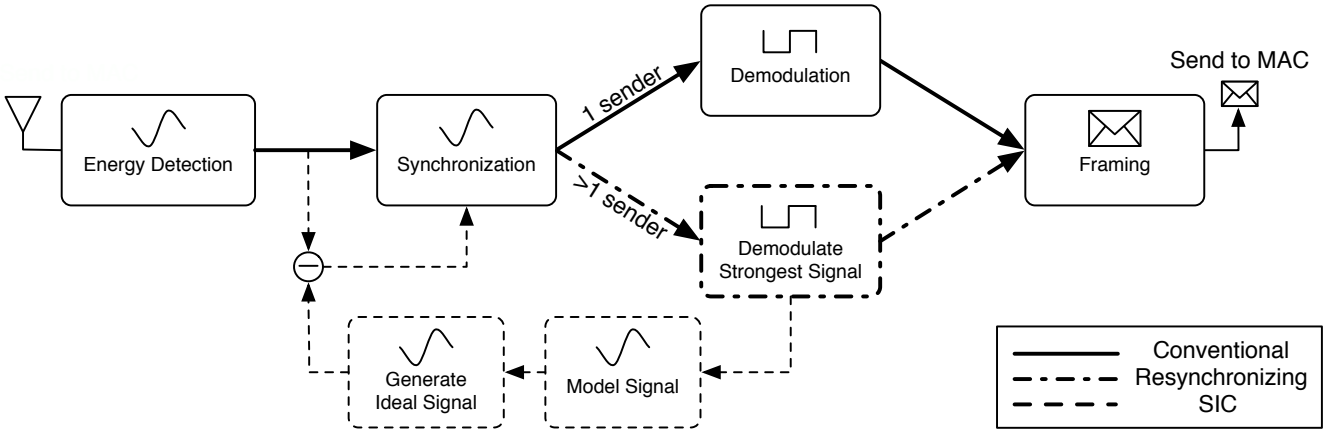



**Figure 3: Block diagram of successive interference cancellation. Filtering, energy detection, and synchronization occur as in the conventional detector. After synchronization and decoding of the headers for the leading packet, the calculated transmission length is used to determine whether one or multiple senders are present. In the case of a collision (dashed lines), the strongest signal is demodulated, including re-synchronizing if it is the later packet. This signal is then modeled to generate an ideal copy of the signal from the decoded bits. This approximation is then canceled from the original signal, and the result (the remaining transmissions, plus approximation error) is fed back into the detector.**

## 3.2 IEEE 802.15.4, the ZigBee PHY

The IEEE 802.15.4 standard [10] provides a physical layer for sensor networks and other wireless personal area networks. At 2.4 GHz, it sends up to 128 byte packets at a low 250 kbps rate and uses direct sequence spread spectrum (DSSS) to map every group of 4 bits to one of 16 sequences of 32 bits (or "chips") known as ZigBee *codewords*. ZigBee uses offset quadrature phase shift keying (O-QPSK) modulation, which maps pairs of bits from the data into four different electromagnetic waveforms, known as symbols. Thus a single byte of data is coded into two ZigBee codewords of 32 bits each each, and these 64 bits are mapped into 32 O-QPSK symbols of two bits each.

The true bitrate of the ZigBee PHY is 2 Mbps. This is well suited to the 8 MHz sampling limit on our software radios with $4\times$ oversampling as commonly used in wireless receivers. (Note that while $2\times$ oversampling is the required Nyquist rate, real world conditions and implementations typically require more frequent sampling.) Low rate 802.11 modes share many features with ZigBee including comparable data rates, (250 Kbps versus 1 and 2 Mbps) and the use of DSSS (ZigBee's spreading factor is 8 whereas 802.11 uses 11 chips per bit).

## 3.3 Our receivers

We implemented three different detection algorithms to model different types of receivers. We implemented two single-packet receivers representing the performance of commercial hardware today, and a receiver using SIC.

### *3.3.1 A conventional single-packet detector*

A block diagram of our conventional single-packet ZigBee receiver is shown in Figure 2. First, a filter known as a channel selector isolates the 3 MHz wide frequency band occupied by a ZigBee channel. Then an energy detection block, with threshold about 2 dB above the noise floor detects incoming transmissions of sufficient strength. The receiver then synchronizes on the preamble, a predetermined sequence of bits that indicates the start of a ZigBee packet. During

synchronization, the detector uses a Kay estimator [20] to determine the frequency offset between the transmitter and receiver and uses simple methods to determine the correct sampling interval (i.e., which offset in the 4× oversampled signal best contains the signal) and the boundaries between the 32-chip ZigBee codewords. Finally, the receiver demodulates the chips in the rest of the packet and then uses the minimum Hamming distance (number of bits that differ) between the received chip sequence and the set of 16 codewords to decode that sequence to the most likely codeword.

We use hard decoding to determine the individual chips in isolation rather than considering the group of 32 symbols together. This entails a penalty of around 3 dB, effectively increasing the noise floor, but we use the same procedure in our SIC implementation.

Finally, the conventional receiver synchronizes on the preamble of a ZigBee transmission and then uses the calculated parameters to decode the remainder of the packet. This implies that a stronger interrupting transmission may not be recovered even with sufficient SINR, behavior that is observed in commercial WiFi devices (notably, the Prism hardware [6]).

#### 3.3.2 A resynchronizing detector

To improve upon the performance of the conventional detector, with the aim of performance that obeyed the SINR model as closely as possible, we also implemented a detector with the ability to resynchronize on the second packet in a collision. This type of capture reflects the empirical behavior of Atheros chipsets.

There are many ways to implement a single-packet detector that will resynchronize and capture a strong packet that begins second during a collision; a key goal in our design was to facilitate the reuse of the same blocks in the SIC receiver. This sharing provides for a direct comparison between the two algorithms, isolating the impact of using SIC from the side effects of other design and implementation decisions. Thus, our resynchronizing detector implements only the base case of a two packet collision as this is sufficient for our experiments. It also operates on the entire captured collision signal. Both of these implementation restrictions can be relaxed in future work.

After the channel select filter and energy detection, synchronization and timing of a detected energy burst (which may be a single packet or may be a collision) occurs as in the conventional detector presented above. Once the headers of the first packet in the burst are decoded, the calculated transmission length is used to determine whether one or multiple senders are present. In the single-sender case, the conventional receiver will decode the packet as normal.

During a collision, we first separate the energy burst into transmission regions for the two interfering senders. We can use the above calculated transmission length of the leading packet for the first sender, and determine the beginning of the second transmission via a sharp increase in the variance of the received signal amplitude (as in [14]). We next determine the amplitude of each received signal. Unless the packets overlap completely, the isolated transmission regions at the beginning and end of the energy burst can be used to determine the individual signal power levels; other methods exist [14] that could be used to determine the signal strengths if packets overlap completely.

After identification of the stronger signal, and resynchronization if it arrived second, we demodulate the stronger signal using the same methods as before.

#### 3.3.3 Our SIC receiver

We present a block diagram of our SIC receiver in Figure 3. To provide a basis of comparison, our implementation reuses the same blocks as the resynchronizing single-packet detector, adding only those elements necessary to implement SIC.

After demodulating the stronger signal as specified above, the SIC receiver builds a model of that signal as a function of the underlying bits. To construct this model, we use the averaging techniques described in Section 2.3. In practice, we found that for ZigBee in our environment, the current sample $R(t)$ depends only upon 3 consecutive symbols. As ZigBee encodes two bits per symbol, these three symbols include 6 bits and induce 64 different models. More advanced physical layers transmit more bits per symbol — for instance 802.11g [9] sends 288 bits at once — but SIC algorithms have been proposed and simulated for these systems as well [12]. We then use these data-dependent models and the recovered bits to approximate the signal and then subtract the generated waveform, leaving only the approximation error and the second, weaker signal.

Finally, we re-synchronize and demodulate the second signal with our standard detector, and pass the demodulated bits up the network stack.

### 3.4 A SIC MAC

Our SIC receiver has implications for the MAC layer because it supports the reception of multiple packets at once. We discuss these issues here to show how SIC fits as part of a complete wireless LAN.

**Link layer ACKs.** Link layer ACKs can be problematic in a network with collisions. This is because ACKs are typically defined to occur in a small time-window after transmissions, potentially interfering with a conflicting packet still in progress. Another design is to allow a deferred ACK. Even without this, if only one of the colliding packets is intended for a node, then the node can safely ACK the packet. This ACK may cause a further collision — with the packet for another node that is still underway. Fortunately, this other node can use a SIC receiver to resolve the ACK collision, and will benefit from the fact that nearly all the bits of the incoming ACK packet are well known. Thus we expect that two links with different receivers will be able to concurrently receive and ACK packets using SIC. Note that in our implementation, the weaker signal may not be decoded until well after it completes — a topic we turn to next.

**Buffering and latency.** Our implementation buffers an entire packet to form a data-model. The amount of buffering will depend on how much time is needed to form a data-dependent model for cancellation. This may be significantly less than a complete packet, and once the model is built only a few symbols need to be buffered. It is also likely that previous transmissions from a node can be used to model its transmissions, at least for recent transmissions and slowly changing channels, e.g., stationary nodes. We plan to explore this in the future.

**SIC and increased control.** Senders in 802.11 wireless LANs typically have more control than in ZigBee; they can choose from multiple rates and power levels through coor-

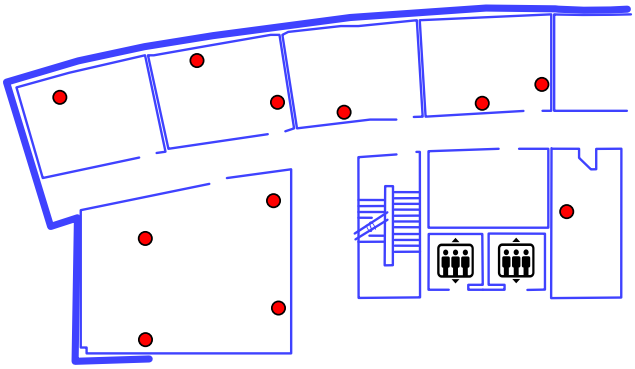


**Figure 4: Our ZigBee USRP testbed. We deployed eleven nodes over six rooms in a university environment. There are both lossy and robust links and substantial RF barriers including elevators and an electrical closet.**

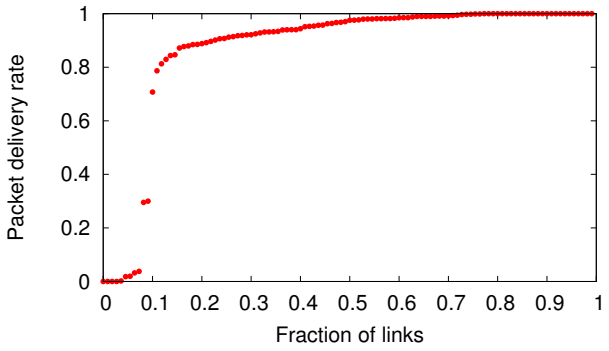



**Figure 5: Delivery rates for all 110 unidirectional links in the testbed. We use the sharp transition at 75% delivery to label a link as "good" for consideration in further experiments.**

dination (implicitly through e.g. ACKs, or explicitly) with receivers. In these situations, opportunities for capture and the subsequent application of SIC may decrease. But theoretical work shows that SIC can increase multiuser wireless LAN capacity [22]. Additionally, CMAP [24] demonstrated that the necessary condition for the use of SIC, capture, does occur in a real 802.11a deployment; SIC would complement this system by adding more sets of nodes that are exposed, rather than hidden terminals. Thus successive interference cancellation can benefit wireless LANs, even when power and rate control are applied.

**SIC as part of a broader solution.** Since SIC cannot resolve all collisions in practice, e.g., the regions of no coverage shown in Figure 1 (b), some MAC-layer method must be added to handle these corner cases. And although the principles underlying SIC apply with more than two senders, it will be increasingly difficult to recover the third or fourth concurrent collision as the signal diminishes and residual noise accumulates. Instead, we view SIC as part of a solution that involves carrier sense or other MAC scheduling mechanisms. Given that contention tends to be low in practice [17], SIC provides a margin of robustness that supports fast and loose scheduling. Determining the best MAC to complement SIC is an interesting issue for future work.

# 4. EXPERIMENTAL SETUP

In this section we describe our testbed and our experimental methodology. We deployed an unplanned, ad hoc testbed of USRP nodes with our ZigBee receivers to understand the benefits of interference cancellation. Section 5 presents our baseline measurements of SIC effectiveness, and Section 6 uses these measurements to estimate the network performance when links contend.

## 4.1 USRP testbed

To analyze the effects of our interference cancellation technique in a dense wireless environment, we deployed a testbed of 11 USRP nodes (Figure 4). Four nodes are located in a lab, six in nearby offices, and the last in a machine room. Metal elevators provide transient path loss on some links.

The USRPs communicated on ZigBee channel 15 centered at 2.425 GHz, situated centrally between WiFi channels 1 and 6. Six 802.11b/g access points (three each on channels 1 and 6, including the floors above and below) are nearby and more are distributed throughout our building and adjacent buildings. Note that despite the 12/13 MHz separation, the spectral mask for 802.11g is less restrictive and requires only 20 dB of attenuation as compared with 30 dB in 802.11b; thus 802.11 nodes can provide a potentially significant amount of co-channel interference. We ran our experiments at night to minimize this effect.

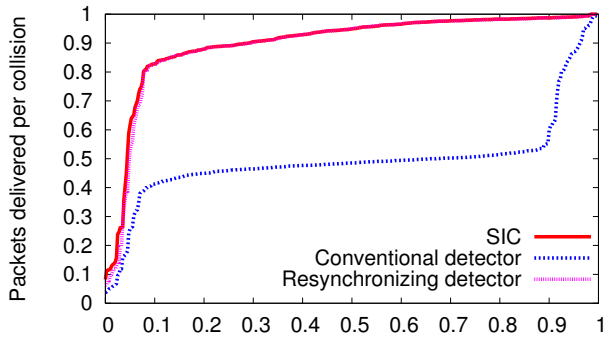
## 4.2 Experimental methodology

Our ideal experiment would evaluate SIC in an end-to-end way, with online receivers using real traffic models to understand the effects of SIC on network performance. However, software radio platforms such as the USRP do not readily support these experiments because of high I/O latency that prevents the rapid transmission of link layer ACKs, and general-purpose frameworks that are less efficient than the DSP implementation of commercial NICs. Instead, we focus on packet reception experiments that evaluate the core capabilities of our USRP-based ZigBee and SIC receivers. We then use these experiments to emulate the performance of a network comprised of receivers and carrier-sense senders.

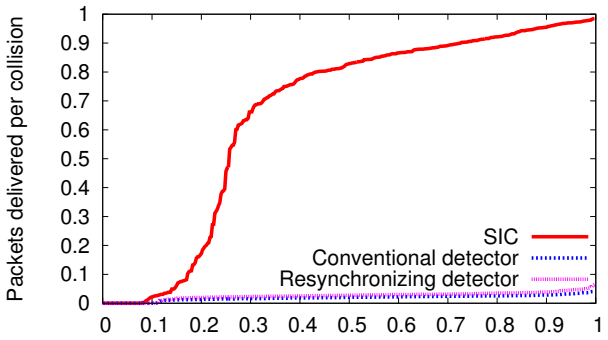
**Link quality.** We measured the quality of the 110 unidirectional links in the network by calculating the delivery rate of 1,000 packets sent from each sender and received by the conventional ZigBee receiver. The CDF of delivery rates is shown in Figure 5. We see that most links reliably deliver most of their packets, with a sharp decline to the 8.2% of links that deliver less than 10% of packets. We used the transition point of this curve to label a link as "good" if it delivers more than 75% of its packets and define $L$ to be this set of links.

First, note that there is not all-to-all connectivity in our testbed. Hidden terminals play an important role in wireless networks, and our network contains hidden terminals even for a minimum carrier sense threshold. Secondly, we only measure SIC performance for reliable links in the set $L$; SIC improves the network under collisions but does not make single links better.

**Comparing receivers.** In each of our experiments, receiving nodes record the raw received signal samples to disk for later processing by our single-packet and SIC ZigBee re-



(a) Fraction of triples, stronger transmitter



(b) Fraction of triples, weaker transmitter

**Figure 6: Cumulative distribution function (CDF) of packet delivery rates for the stronger (left) and weaker transmitters at a receiver $R$ for 400 $(A, B, R)$ triples. Synchronization on the first, weaker transmission causes the conventional detector to drop half of strong packets; both the resynchronizing detector and SIC recover most of these transmissions. Unlike the conventional and resynchronizing detectors, SIC delivers the weaker packet in a collision with a median probability of 83%.**

ceivers. This allows us to obtain a direct comparison of the three receivers on exactly the same received data. It gives us reproducibility and mitigates concerns of variability in the results due to slight differences in the workload or wireless channels.

# 5. SIC RECEIVER PERFORMANCE

The key feature of SIC is the ability of a receiver to recover a weak packet during a collision. In this section we evaluate this baseline feature in an experiment in which we sent traffic shaped to cause collisions and exercise SIC.

## 5.1 Sender workload

To determine whether SIC can successfully recover signals under interference, and whether it applies to most pairs of nodes in our unplanned network or required careful topology tuning, we ran the following simple microbenchmark. Note that this microbenchmark is not intended to be indicative of typical behavior, but rather to measure SIC. We chose pairs of USRP nodes as transmitters and sent 1,000 of the largest sized ZigBee packets (6 header bytes plus 128-byte payloads) with carrier sense disabled. Note that 128 bytes is equivalent in time to 512 bytes at 1 Mbps or 1 024 bytes at 2 Mbps, and is longer than any packet of a higher rate. One sender emitted packets at 50% load, delaying for one packet interval between sends. The other sender delivered packets with an interpacket spacing of 1.1 packet lengths (about 48% load), with small random jitter added. This workload forces a high number of two-packet collisions with uniform random amounts of overlap between the two packets. For each of the 55 pairs of senders, the other nine nodes acted as receivers and logged the raw received complex samples output by the USRP to disk. These traces were later replayed to our implementations of two single-packet ZigBee detectors (Sections 3.3.1 and 3.3.2) and our SIC detector (Section 3.3.3).

## 5.2 Delivery under interference

For each pair of transmitters $A$ and $B$, and receiver $R$, we consider the stronger sender at $R$ to be the transmitter with the most packets decoded correctly. Note that for a given pair of transmitters, $A$ may be the stronger transmitter at one receiver while $B$ is stronger at another. Also be aware that while our definition of "stronger," higher packet delivery, will correlate with a larger signal strength, some individual transmissions from the "stronger" transmitter may have a lower signal strength due to varying fades. We define

$$T = \{(A, B, R) \mid (A, R) \in L \text{ and } (B, R) \in L\}$$

to be the set of triples in which both links are reliable. Of the 495 unique triples in the network, 400 contain two reliable links. Figure 6 plots the distribution of packet delivery rates for the stronger (a) and weaker (b) transmitters.

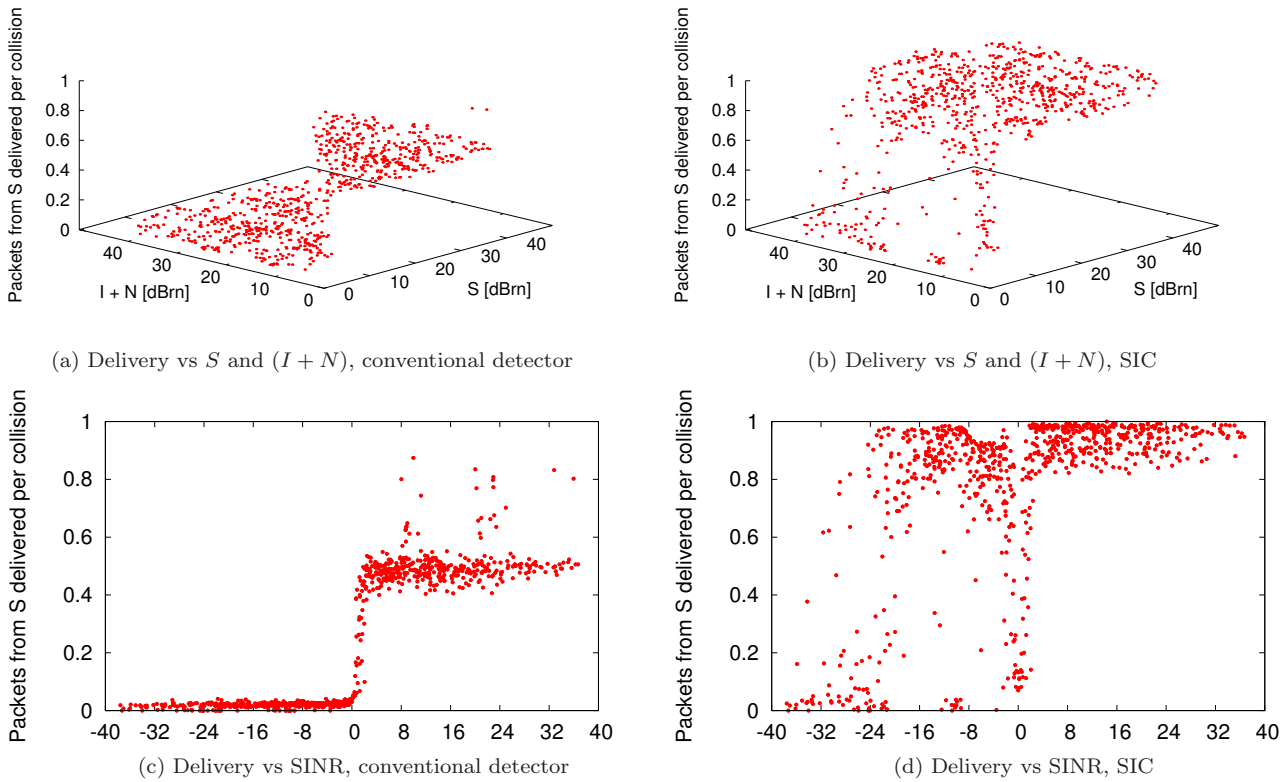
Figure 6 (a) shows that both SIC and the resynchronizing detector remain reliable in the presence of an interferer for 93% of triples. Slight differences between the two can be attributed to the gain of SIC when the normally stronger receiver becomes weaker due to fading. The conventional detector, as expected, typically drops half of the stronger transmitter's packets when it synchronizes on the first, weaker transmission. In some cases, the stronger signal dominates such that the weaker transmission is lost and greater than 50% of stronger packets to be received. This occurs for about 8% of triples.

Figure 6 (b) shows that SIC greatly increases the ability of a receiver to recover the weaker transmission over both single-packet detectors. The *weaker* link remains reliable (delivers 75% of its packets) in 63% of triples. In comparison, the conventional and resynchronizing receivers recovered a median of 2-3% of these weaker packets — only when the overlap of the colliding packets was minimal.

Our microbenchmarks demonstrate that SIC can allow a receiver to reliably recover packets under interference, even when the interfering signal is stronger. That is, SIC improves the robustness of a receiver to hidden terminals without requiring other changes to the MAC or physical layer.

## 5.3 Delivery vs SINR

Figure 7 shows packet delivery rates as a function of signal, interference, and noise power for the SIC and conventional receiver implementations. We normalize power levels with respect to noise. These four figures plot 800 total points,




(a) Delivery vs $S$ and $(I + N)$, conventional detector

(b) Delivery vs $S$ and $(I + N)$, SIC

(c) Delivery vs SINR, conventional detector

(d) Delivery vs SINR, SIC

**Figure 7: Delivery rate of $S$ as a function of its signal power and the combined power of interference $I$ and noise $N$. The four figures plot 800 total points, two each for the 400 triples in $T$, alternatively treating $A$ and $B$ in turn as $S$ and $I$. Figures (a) and (b), showing the relative strengths of $S$ and $(I + N)$, demonstrate that packet delivery in our network qualitatively matches the expected theoretical behavior displayed in Figure 1. Plots (c) and (d) show the delivery rate as a function of SINR, which is simply $S$ - $(I + N)$ with these quantities in decibels. The conventional algorithm recovers packets only when the SINR is positive, while SIC can reliably recover packets on links with large negative SINR by modeling and canceling the signal. The resynchronizing detector matches the conventional detector for SINR $< 0$ and SIC for SINR $> 0$.**

two each for the 400 triples in $T$ where we alternately treat $A$ and $B$ as $S$ and $I$. We omit the resynchronizing detector as it closely matches the conventional detector for SINR $<$ 0 and SIC for SINR $> 0$.

Figure 7 (a) and (b) show delivery as a function of the relative strengths of the signal and the interference, and corroborate the theoretical results displayed in Figure 1. Parts (c) and (d) show delivery as a function of SINR, which simply equals $S$-$(I + N)$ with these quantities in decibels. The conventional detector delivers packets only when the SINR is sufficiently large and in most cases, only when the desired signal is transmitted first. SIC can reliably recover packets at SINR values as low as -30 dB. Note that with SIC, the delivery rate is not captured entirely by the relative signal powers (SINR); the absolute signal powers in (b) affect delivery as well because $S$ must be sufficiently strong to overcome the residual error in the canceled $I$.

# 6. SIC NETWORK PERFORMANCE

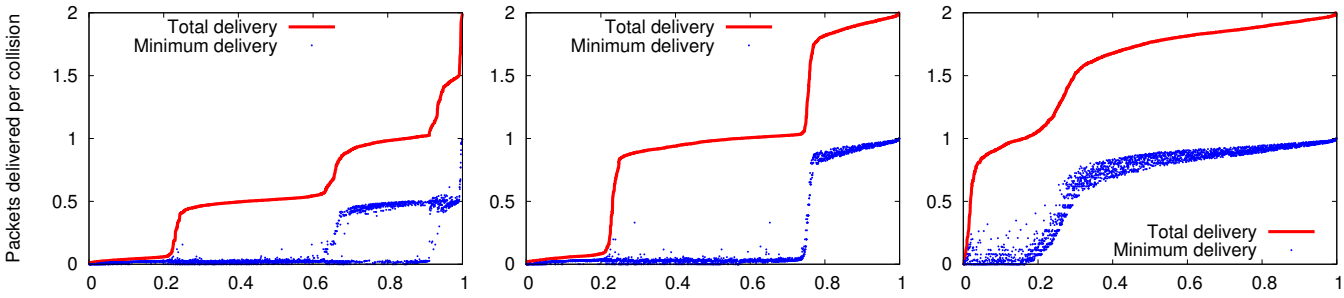
In the previous section, we showed how SIC improves delivery when there are collisions. However, our experiments do not directly show the effects of SIC on the network because they do not include carrier sense and analyze delivery from the point of view of a single receiver. Here, we use our measurements to estimate the effects of SIC for different carrier sense thresholds. We consider spatial reuse, packet delivery, and fairness for competing pairs of links in our testbed. We find that SIC achieves overall network performance gains with minimal negative consequences for individual links.

## 6.1 Potential for spatial reuse

Network designers strive for spatial reuse because allowing multiple links to send simultaneously improves the capacity of the network. To explore the properties of spatial reuse for pairs of reliable unidirectional links in our network we define

$$Q = \{(\ell_1, \ell_2) \text{ with distinct nodes } \mid \ell_1 \text{ and } \ell_2 \in L\}$$

to be the set of quadruples of nodes forming two interfering, reliable unidirectional links in the network. In our testbed there are 3 120 pairs of reliable links (of 3 960 pairs total). If $A \rightarrow R$ and $B \rightarrow S$ are the two competing links, then we quantify spatial reuse with the results from the previous section, calculating the delivery ratios from $A \rightarrow R$ while $B$ interferes and vice versa.




(a) Fraction of pairs of links conventional detector

(b) Fraction of pairs of links resynchronizing detector

(c) Fraction of pairs of links SIC

**Figure 8: Comparison of packet delivery rates for the 3 120 pairs of reliable interfering links. Separately for each detector, we sort link pairs by the total packets delivered and also plot the delivery of the worse performing link. The median delivery rate for SIC improves by a factor of 1.8 the median delivery rate of a resynchronizing detector. Additionally, spatial reuse benefits around 70% of link pairs using SIC, compared with about 25% without; thus SIC in effect turns 45% of competing links from hidden terminals into links that can send together in this network.**

Figure 8 shows the distribution of packet delivery ratios for the two links using the two single-packet ZigBee receivers and using our SIC algorithm. We sort the quadruples by the sum of the link delivery rates and also plot the reception rate on the link that delivers fewer packets of the pair.

Figure 8 (a) shows the complex behavior of a conventional receiver in our testbed. Collisions starve both links 22% of the time, when neither node experiences capture, receiving fewer than 10% of packets. There is an intermediate region where one link can capture packets but the other cannot, and a full 85% of pairs of links average fewer than 1 packet delivered per collision. This implies that spatial reuse can only benefit the last 15% of link pairs, when both nodes experience capture and one link pair is sufficiently isolated that it does not even synchronize on the interferer's packets. Overall, the median rate is about 0.5 packets delivered by the conventional detector per collision, i.e. the throughput of each link is degraded by at least half during collisions and on average by a factor of 4.

In Figure 8 (b), we see that resynchronization greatly helps the network. The same 22% of link pairs are still starved, but now the stronger link is reliable in most of the remaining cases. Spatial reuse would benefit 25% of pairs for the delivery rate exceeds by at least ten percent 1 packet per collision. The median rate is 0.94 packets delivered per collision, and thus the network performs in aggregate about as well as if only a single node sent. However, in 74% of cases, the weaker link nearly starves with less than 10% of packets delivered.

In contrast, Figure 8 (c) shows that SIC greatly increases the ability of a receiver to perform under interference, and provides a much larger potential for spatial reuse in our network. With SIC there are barely any pairs of links for which collisions cause both links to starve and in fact for 61% of link pairs the *worse* performing link delivers at least 3/4 of its packets, meeting our threshold for reliability. And now 77% of links have spatial reuse. Compared with the previous two cases, the use of SIC increases the number of link pairs that benefit with spatial reuse by more than a factor of 5 over the conventional detector (15%) and by a factor of 3 with resynchronization (25%). Finally, the median rate is about 1.8 packets delivered per collision — when receivers in our network use SIC, for most pairs of links both links work almost as well together as they do separately.

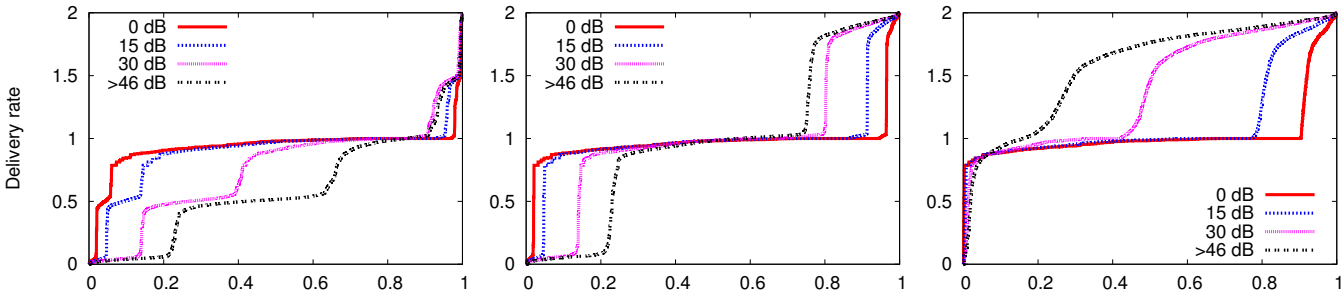
## 6.2 Spatial reuse via carrier sense

We investigate the properties of SIC under carrier sense as the most straightforward way to provide spatial reuse. For these results, we assume all nodes use a uniform carrier sense threshold, and use measured RSSI values between nodes to determine whether a sender will defer to an ongoing transmission from another node. From this information, we extrapolate the number of packets delivered per transmission as a function of the sensing threshold.

**Measuring RSSI.** For each of the 110 sender/receiver pairs ($A$,$B$) in our network, nine of the experiments we analyzed in Section 4 contained transmissions from $A$ to $B$. We computed the signal power at the receiver of every packet where the CRC matched, took the median for each trial, and used the median value over all nine trials as the pairwise, unidirectional RSSI. Over the nine trials, the worst-case difference between the median RSSI and the maximum or minimum measured RSSI was within 15%, a difference of less than 0.6 dB. The $90^{th}$ percentile difference was 0.4 dB (9%) and the $80^{th}$ percentile was an error of 0.2 dB (5%). Thus, RSSI on the links of our network is consistent over many trials.

**Varying carrier sense.** We vary the carrier sense threshold in our topology from 0 dBrn (decibels relative to noise) to 50 dBrn. At the low end, concurrent sending is completely disabled in the network except for a few nodes for which the measured RSSI is below the noise floor. The strongest RSSI between nodes in our network is 46 dBrn, above which carrier sense is effectively turned off and all pairs of nodes interfere.

Figure 9 plots the delivery rates of two competing links under interference as a function of threshold. Figure 9 (a) shows that with the conventional detector, raising the carrier sense threshold is uniformly bad. This follows naturally from Figure 8 (a) in which we learn that our topology has extremely limited potential for spatial reuse with that receiver. The resynchronizing algorithm (Figure 9 (b)) performs no-

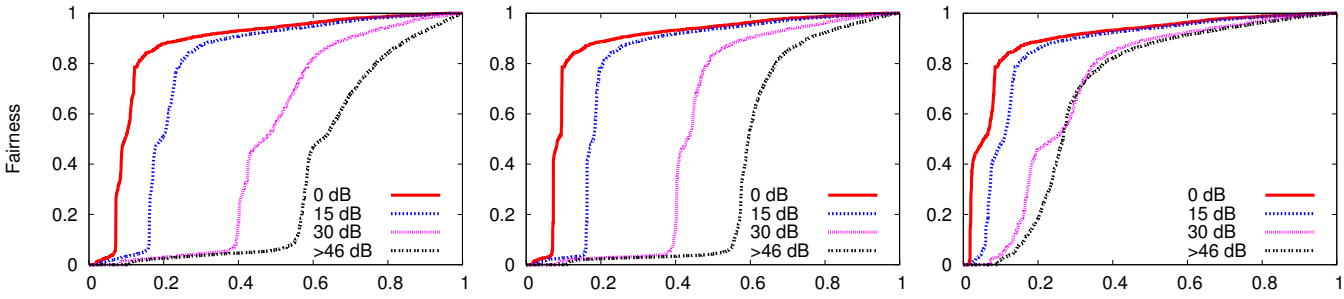



(a) Fraction of pairs of links conventional detector

(b) Fraction of pairs of links resynchronizing detector

(c) Fraction of pairs of links SIC

**Figure 9: Delivery rate as a function of the carrier sense threshold. For each ordered pair of links, we use the carrier sense threshold and our measurements to determine the number of packets delivered when one link sends first. Though some link pairs achieve spatial reuse with the conventional receiver, raising the carrier sense threshold generally degrades performance. The resynchronizing detector fares noticeably better, but many link pairs suffer from collisions without carrier sense. In contrast, with SIC increasing the threshold improves aggregate performance by 52% and decreases delivery for only 7% of links using SIC. Above 46 dB, no nodes defer and carrier sense is effectively disabled.**




(a) Fraction of pairs of links conventional detector

(b) Fraction of pairs of links resynchronizing detector

(c) Fraction of pairs of links SIC

**Figure 10: Evaluation of fair capacity sharing between the two interfering links. We define fairness as the ratio of delivery between the worse link and the better link. For any fixed carrier sense threshold, SIC is more fair than both single-packet detectors. In all receivers, including SIC, fairness degrades as carrier sense is disabled. The crossing lines in the SIC graph show a small number of cases in which increasing the threshold improves fairness when, because of asymmetric RSSI, only one of two competing senders will defer.**

ticeably better, but many link pairs still suffer as the carrier sense threshold is raised. On the other hand, in the most conservative setting with carrier sense thresholds set to suppress concurrent sends, SIC improves packet loss rate from 18% for the conventional detector (14% with resynchronization) to 8% due to better handling of hidden terminals. And Figure 9 (c) shows that, with SIC, raising the carrier sense threshold *promotes* effective spatial reuse. Aggregate performance increases by 52% over the conservative case, and only 7% of the link pairs experience reduced performance with carrier sense completely disabled.

**Fairness.** We assess the fairness of each carrier sense threshold in Figure 10. For each pair of competing links, we plot the ratio of packets delivered on the weaker link to the rate of delivery on the stronger link. Hidden terminals can exacerbate unfairness because one link may dominate another. Because SIC allows better recovery of packets under collisions, we see that for any threshold it provides better fairness than either single-packet receiver. Interestingly, the lines that cross in the SIC graph show that in a small number of cases, increasing the carrier sense threshold can improve fairness when only one of the two competing senders defers because of asymmetric RSSI.

## 7. RELATED WORK

This paper advocates the use of successive interference cancellation to combat hidden terminals and to promote spatial reuse in wireless LANs. There is a large body of related work including physical layer algorithms, theoretical analyses, MAC protocol designs, and some recent work that like ours also investigates concurrent sending in ad-hoc wireless LANs.

SIC is drawn from a wide class of multiuser detection (MUD) techniques from the study of wireless communications. While these techniques are well known in the literature [23], the majority of research deals with cellular networks (see e.g. [1] and [8]), which are marked by centralized allocation of power, rate, coding, frequency, and schedul-

ing of transmissions. Work in this field on chaotic wireless LANs is limited, focusing mainly on theoretical analyses of network capacity in small (up to 5 nodes) simulations [22].

In the computer networking field, carrier sense has been the primary means of combating hidden terminals while promoting spatial reuse. Karn proposed the RTS/CTS [13] mechanism to combat both hidden and exposed terminals, but in practice it proves to be overly conservative. Different carrier sense-based clear channel assessment methods [9], tuned to the physical layer of the receiver or to properties of the network, can make carrier sense more effective in a network [2] but can lead to degradation of the system in the presence of heterogeneous hardware [3], environmental changes, or churn in the topology such as mobile nodes. More effective techniques for enhancing performance in a contentious network focus on tuning clear channel assessment thresholds, transmit power, and data rate to better achieve spatial reuse while avoiding hidden terminals [15, 25]. Vutukuru et al. [24] presented a system that identifies the conflict graph [11, 18] — links that cannot send concurrently — and modifies the carrier sense mechanism to only defer in that case. Our work with SIC is complementary to these approaches; while these techniques are designed to reduce exposed terminals while avoiding hidden terminals, the use of SIC can turn hidden terminals into nodes that can send together.

A final branch of related work closely connected to ours explores physical layers that support concurrency. Katti et al. [14] use knowledge of one signal to cancel another, but focus on this as a network coding technique in which collisions between pairs of nodes are carefully coordinated by the MAC. Their technique cancels signals above the level of raw electromagnetic signals and therefore has much worse performance at low (negative in dB) SINR. In earlier work [7], we demonstrated a multiuser detection (MUD) proof of concept that generalizes successive interference cancellation to work when the signals in a collision are close in power. This approach is in theory more powerful than SIC, but has much higher computational complexity. Finally, Li et al. [16] implemented a form of successive cancellation for superposition coding, which resolves two packets coded together and sent by a single transmitter using rate and power control. In contrast, we explore the general case of uncoordinated asynchronous colliding transmissions from distinct senders. All of these previous systems are only evaluated in a few simple and customized topologies, while we evaluate our system over all configurations on an unplanned, 11-node software radio testbed and extrapolate MAC-level effects and the tradeoff between carrier sense threshold and performance using SIC. To the best of our knowledge, ours is the first work in which successive interference cancellation techniques have been adapted for, implemented, and evaluated on wireless LANs.

# 8. CONCLUSION

Collisions are the bane of today's wireless networks. Some collisions are unavoidable, due to hidden terminals; other collisions are acceptable but prevented by carrier sense. To help address this problem, this paper presents the practical design of a successive interference cancellation technique which provides robustness in a wireless network by enabling receivers to better recover colliding packets. We prototyped a simple version of our design and found that for our unplanned, ad hoc eleven node topology, SIC promotes spatial reuse for the vast majority of competing, reliable links. We also analyze the behavior of a network with differing carrier sense thresholds and find that SIC significantly outperforms receivers implementing single-packet detection algorithms during collisions. Our measurements also show that SIC maintains fairness among most pairs of competing links. The next step forward towards understanding the potential effects of these techniques is to build an online SIC detector as part of a complete PHY/MAC design and evaluate it in a larger network under a variety of real workloads. Looking ahead to networks that employ multiple rates, the ability to run experiments with real workloads will allow investigation of how to design mechanisms to promote spatial reuse beyond simply changing carrier sense thresholds.

# 9. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their comments. This work was supported in part by the National Science Foundation (CNS-0435065), Intel, Microsoft, and the UW College of Engineering.

# 10. REFERENCES

[1] J. G. Andrews. Interference cancellation for cellular systems: A contemporary overview. *IEEE Wireless Communications*, 12(2):19–29, Apr. 2005.

[2] M. Bertocco, G. Gamba, A. Sona, and S. Vitturi. Performance measurements of CSMA/CA-based wireless sensor networks for industrial applications. In *IEEE IMTC*, 2007.

[3] A. Conti, D. Dardari, G. Pasolini, and O. Andrisano. Bluetooth and IEEE 802.11b coexistence: Analytical performance evaluation in fading channels. *IEEE JSAC*, 21(2):259–269, Feb. 2003.

[4] Ettus Research, LLC. `http://www.ettus.com`.

[5] GNU Radio Project. `http://gnuradio.org/trac`.

[6] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and mitigating the impact of RF interference on 802.11 networks. In *ACM SIGCOMM*, 2007.

[7] D. Halperin, J. Ammer, T. Anderson, and D. Wetherall. Interference cancellation: Better receivers for a new wireless MAC. In *HotNets-VI*, 2007.

[8] J. Hou, J. E. Smee, H. D. Pfister, and S. Tomasin. Implementing interference cancellation to increase the EV-DO Rev A reverse link capacity. *IEEE Communications Magazine*, 44(2):58–64, Feb. 2006.

[9] IEEE. IEEE Std. 802.11g-2003: Further higher data rate extension in the 2.4 GHz band. `http://www.ieee802.org`, 2003.

[10] IEEE. IEEE Std. 802.15.4-2006 : Wireless medium access control and physical layer specifications for low-rate wireless personal area networks. `http://www.ieee802.org`, 2003.

[11] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance,. In *ACM MobiCom*, 2003.

[12] Y. Jung and J. Kim. Symbol detection algorithm and implementation results for space-frequency OFDM transmit diversity scheme. In *AP-ASIC*, 2004.

[13] P. Karn. MACA - a new channel access method for packet radio. In *ARRL 9th Computer Networking Conf.*, 1990.

[14] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference: Analog network coding. In *ACM SIGCOMM*, 2007.

[15] T.-S. Kim, H. Lim, and J. C. Hou. Improving spatial reuse through tuning transmit power, carrier sense threshold, and data rate in multihop wireless networks. In *ACM MobiCom*, 2006.

[16] L. Li, R. Alimi, R. Ramjee, J. Shi, Y. Sun, H. Viswanathan, and Y. R. Yang. Extended abstract: Superposition coding for wireless mesh networks. In *ACM MobiCom*, 2007.

[17] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Analyzing the MAC-level behavior of wireless networks in the wild. In *ACM SIGCOMM*, 2006.

[18] D. Nicolescu. Interference map for 802.11 networks. In *IMC*, 2007.

[19] N.-J. Oh and S.-G. Lee. Building a 2.4-GHz radio transceiver using IEEE 802.15.4. *IEEE Circuits and Devices*, 21(6):43–51, Nov./Dec. 2005.

[20] J. G. Proakis. *Digital Communications*. McGraw-Hill, 2001.

[21] A. Sheth, C. Doerr, D. Grunwald, R. Han, and D. Sicker. MOJO: A distributed physical layer anomaly detection system for 802.11 WLANs. In *MobiSys*, 2006.

[22] S. Toumpis and A. J. Goldsmith. Capacity regions for wireless ad hoc networks. In *ISCTA*, 2001.

[23] S. Verdu. *Multiuser Detection*. Cambridge University Press, 1998.

[24] M. Vutukuru, K. Jamieson, and H. Balakrishnan. Harnessing exposed terminals in wireless networks. In *USENIX NSDI*, 2008.

[25] J. Zhu, X. Guo, L. L. Yang, W. S. Conner, S. Roy, and M. M. Hazra. Adapting physical carrier sensing to maximize spatial reuse in 802.11 mesh networks. *Wirel. Comm. Mob. Comput.*, 4(8):933–946, Dec. 2004.