Clearing the RF Smog: Making 802.11 Robust to Cross-Technology Interference

Shyamnath Gollakota [†] Fadel Adib[†] Dina Katabi[†] Srinivasan Seshan[⋆]
[†]Massachusetts Institute of Technology [⋆]Carnegie Mellon University
{gshyam, fadel, dina}@csail.mit.edu srini@cs.cmu.edu

ABSTRACT

Recent studies show that high-power cross-technology interference is becoming a major problem in today's 802.11 networks. Devices like baby monitors and cordless phones can cause a wireless LAN to lose connectivity. The existing approach for dealing with such high-power interferers makes the 802.11 network switch to a different channel; yet the ISM band is becoming increasingly crowded with diverse technologies, and hence many 802.11 access points may not find an interference-free channel.

This paper presents TIMO, a MIMO design that enables 802.11n to communicate in the presence of high-power cross-technology interference. Unlike existing MIMO designs, however, which require all concurrent transmissions to belong to the same technology, TIMO can exploit MIMO capabilities to decode in the presence of a signal from a different technology, hence enabling diverse technologies to share the same frequency band. We implement a prototype of TIMO in GNURadio-USRP2 and show that it enables 802.11n to communicate in the presence of interference from baby monitors, cordless phones, and microwave ovens, transforming scenarios with a complete loss of connectivity to operational networks.

Categories and Subject Descriptors C.2.2 [**Computer Systems Organization**]: Computer-Communications Networks

General Terms Algorithms, Design, Performance, Experimentation

Keywords Cognitive MIMO, Cross-Technology Interference

1. INTRODUCTION

Cross-technology interference is emerging as a major problem for 802.11 networks. Independent studies in 2010 by the Farpoint Group [8], BandSpeed [17], and Miercom [10] all show that high-power interferers like baby monitors and cordless phones can cause 802.11n networks to experience a complete loss of connectivity. Other studies from Ofcom [7], Jupiter Research [1], and Cisco [14] report that such interferers are responsible for more than half of the problems reported in customer networks. Today's high-power non-WiFi sources in the ISM band include surveillance cameras, baby monitors, microwave ovens, digital and analog cordless phones, and outdoor microwave links. Some of these technologies transmit in a frequency band as wide as 802.11, and all of them emit power that is comparable or higher than 802.11 devices [17]. Further, the number and diversity of such interferers is likely to increase over time due to the proliferation of new technologies in the ISM band.

Traditional solutions that increase resilience to interference by making 802.11 fall down to a lower bit rate are ineffective against high-power cross-technology interference. As a result, the most common solution today is to hop away to an 802.11 channel that does not suffer from interference [6, 38, 31, 32]. However, the ISM band is becoming increasingly crowded, making it difficult to find an interference-free channel. The lack of interference-free channels has led WiFi device manufacturers [6, 11, 3] and researchers [29] to develop signal classifiers that inform the 802.11 user about the root cause of the problem (e.g., Bluetooth, microwave, baby monitor). However, these classifiers put the burden of addressing the problem on the user and cannot solve the problem on their own.

In this paper, we ask whether it is possible to use the MIMO capability inherent to 802.11n to address high-power cross-technology interference. MIMO achieves most of its throughput gains by enabling multiple concurrent streams (e.g., packets). Current MIMO decoding, however, fails if any of these concurrent streams belongs to a different technology. Nonetheless, if MIMO can be made to work across technologies, a 3×3 802.11n transmitter can then treat the signal from a baby monitor or microwave as one stream and still deliver two concurrent streams to its receiver.

The challenge in harnessing MIMO across different technologies stems from the fact that MIMO decoding hinges on estimating the channel between all transmit and receive antennas. These estimates rely on understanding the signal structure and assume a known preamble. Hence, it has been infeasible to use MIMO across different and potentially unknown technologies.

We present TIMO,[1] an 802.11n receiver design robust to high-power cross-technology interference. TIMO introduces a MIMO technique that enables a receiver to decode a signal of interest, even when the channel from other concurrent transmissions is unknown. The intuition underlying TIMO is best explained via an example. Consider a pair of 2-antenna 802.11n nodes that want to communicate in the presence of a high-power unknown interferer. Let $s(t)$ be the signal of interest and $i(t)$ the interference signal. The 802.11n receiver node will receive the following signals on its two antennas:[2]

$$\begin{aligned} y_1(t) &= h_i i(t) + h_s s(t) &\quad (1)\\ y_2(t) &= h'_i i(t) + h'_s s(t), &\quad (2)\end{aligned}$$

where h_i and h'_i are the channels from the interferer to the 802.11n

SIGCOMM'11, August 15–19, 2011, Toronto, Ontario, Canada.

---

[1] Technology Independent Multi-Output (TIMO) receiver design.

[2] The equations here are for single-tap channels. Subsequent sections extend these equations to multi-tap channels.

receiver, and h_s and h'_s are the channels from the 802.11n sender to the 802.11n receiver. The 802.11n receiver has to solve these equations to obtain its signal of interest $s(t)$. It knows the received samples, $y_1(t)$ and $y_2(t)$, and the channels from its transmitter, h_s and h'_s, which can be computed in the presence of interference (see §6.4). The receiver, however, cannot compute the channels from the interferer, h_i and h'_i, because it does not know the interferer's signal structure or preamble. Hence, it is left with two equations in three unknowns $(s(t), h_i i(t)$, and $h'_i i(t))$,[3] which it cannot solve.

Note that the receiver can cancel the interference if it knows the interferer's channel ratio $\frac{h_i}{h'_i}$. In particular, the receiver can rewrite equations 1 and 2 to express the signal of interest as:

$$s(t) \quad = \quad \frac{y_1(t) - \beta y_2(t)}{h_s - \beta h'_s} \qquad \text{for } \beta = \frac{h_i}{h'_i}. \tag{3}$$

The only unknown in the above equation is β. Thus, though the 802.11n receiver cannot compute the exact channels of the interferer, it can still cancel its interference using only its channel ratio.

Still, how do we obtain this ratio given no support from the interferer? The receiver can obtain this ratio as follows: Say that for some time instance $t = t_0$, our transmitter sends a known symbol $s(t_0)$. Our receiver can then substitute in equations 1 and 2 to obtain:

$$\frac{h_i}{h'_i} = \frac{y_1(t_0) - h_s s(t_0)}{y_2(t_0) - h'_s s(t_0)}, \tag{4}$$

where all terms are known except for the ratio $\frac{h_i}{h'_i}$. In §6, we develop this idea further and eliminate the need for having the transmitter send a known symbol, which makes the scheme applicable to existing 802.11n frames. We further generalize the solution to address scenarios in which different frequencies have different interferers, or the interferer hops across frequencies.

A MIMO transmitter can also encode its signal using interference nulling [36] so that it does not interfere with a concurrent transmission from a competing technology. However, using a similar computation, we show that it is necessary to obtain the ratio $\frac{h_{s1}}{h_{s2}}$, where h_{s1} and h_{s2} are the channels from the MIMO transmitter to the receiver of the competing technology. These channels can only be estimated if the receiving node transmits data at some point, i.e., if the competing technology uses bidirectional communication, e.g., a cordless phone. If this constraint is met, however, TIMO can be used not only to protect 802.11n networks from high-power interference, but also as a cognitive mechanism that enables MIMO-based nodes to peacefully coexist in the same frequency band with bidirectional non-MIMO nodes from a different technology. In this case, the simpler non-MIMO nodes just transmit bidirectionally, and the more complex MIMO nodes take on the burden of preventing interference. This approach can lead to a new form of spectrum sharing in which different technologies do not necessarily have to find unoccupied bands and, in crowded environments, could instead occupy the same band thereby increasing spectral efficiency.

We have built a prototype of TIMO using 2-antenna USRP2 radios [13]. We have evaluated our design in the presence of interference from three technologies: a microwave oven, an analog baby monitor, and a DSSS cordless phone. We first use commercial 802.11n cards and iperf [33] to transmit in the presence of these interferers. We find that, in our testbed, the cordless phone and the baby monitor prevent 802.11 from establishing any connection, reducing its throughput to zero. The microwave, on the other hand, results in a throughput reduction of 35–90%. We replace the commercial 802.11n cards with our USRP2 nodes and repeat the experiment with and without TIMO. We find that in the absence of TIMO, when the USRP2 nodes are less than 31 feet away from the cordless phone or the baby monitors, they cannot deliver any packets. In contrast, in the presence of TIMO, and for the same locations, their throughput increases to 13-23 Mb/s. We also implement cross-technology interference nulling and show that it enables a MIMO node to significantly reduce the packet loss at the receiver of a competing technology, with the reduction in packet loss being as high as 14x in some locations.

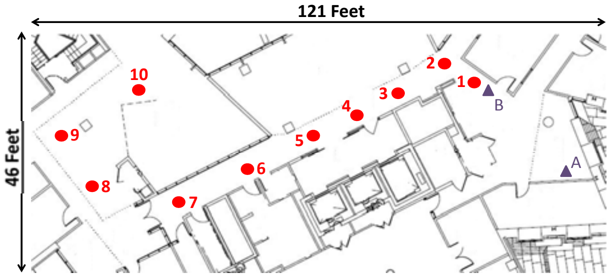


Figure 1—Testbed. An 802.11n transmitter located at A is communicating with an 802.11n receiver at B. The interferer is placed in one of the locations 1 to 10.

2. IMPACT OF CROSS-TECHNOLOGY INTERFERENCE ON 802.11N

We study the interaction between high-power interferers and 802.11n and compare against the interaction between a low power interferer, Bluetooth, and 802.11n. We focus on three high-power technologies that are prevalent in today's environments [7]: DSSS cordless phones, baby monitors, and microwave ovens.

Experimental Setup: We use the Netgear N-300 USB-adapter and the Netgear N-300 router as the 802.11n client and AP respectively. Both devices support 2 × 2 MIMO. We place the AP and the client at positions A and B in Fig. 1. In each run, we place the interferer at one of the marked locations in Fig. 1. Our experiments include line-of-sight and non-line-of-sight situations, and show scenarios in which the interferer is within one foot of the 802.11n client as well as 90 feet away from it. We run iperf on the two 802.11n devices with the 802.11n client acting as the iperf server. The AP sends UDP packets for 2 minutes and logs the average throughput observed every 500 ms. In each location, we compute the observed 802.11n throughput first when the interferer is turned OFF and next when it is ON. Additionally, we use a USRP2 software radio to monitor a 25 MHz bandwidth. The USRP2 simply logs the time signal which we process offline to obtain the time and frequency characteristics of each interferer.

2.1 Digital Cordless Phone

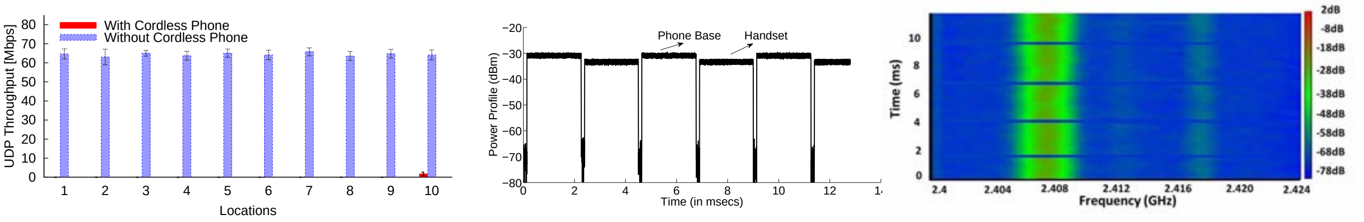
We experiment with the Uniden TRU 4465-2 DSSS cordless handset system. The phone base and handset communicate using digital spread spectrum in the 2.4 GHz range. In each experiment, we fix the 802.11n AP and client at locations A and B and place both the cordless handset and the phone base at one of the locations in the testbed, 5 cm away from each other.

Fig. 2(a1) shows the 802.11n throughput with and without interference from the cordless phone. The figure shows that in the presence of the cordless phone, the 802.11n client and AP could not establish a connection and hence experienced zero throughput.

We next examine the time and frequency profile of the cordless phone to understand why 802.11 lost connectivity. Fig. 2(a2) plots the power profile of the phone as a function of time. The phone base and handset use Time-Division Duplexing (TDD) to communicate in the same frequency band. The handset transmits in the first time slot, followed immediately by a transmission from the phone base.

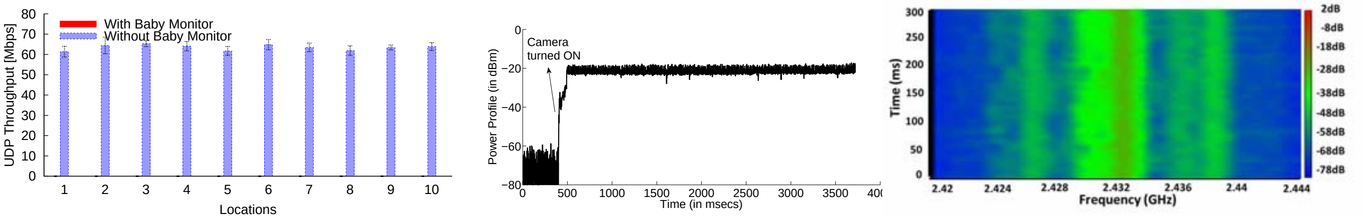
---

[3] We can lump $i(t)$ with the channel variable because we are not interested in decoding the symbols of the interferer.



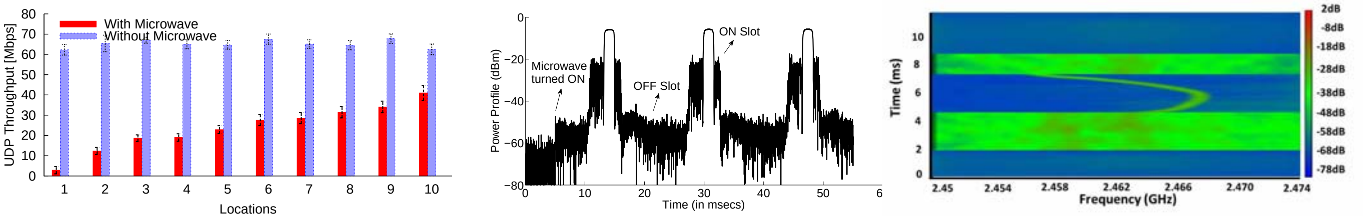
(a1) 802.11n UDP Throughput (a2) Time Profile (a3) Spectrogram

(a) DSSS Cordless Phone



(b1) 802.11n UDP Throughput (b2) Time Profile (b3) Spectrogram

(b) Baby Monitor



(c1) 802.11n UDP Throughput (c2) Time Profile (c3) Spectrogram

(c) Microwave Oven

Figure 2—Characteristics of High Power Interferers in the ISM Band.

Since these devices continuously transmit, the channel is never free. Thus, an 802.11n node that carrier senses the medium never gets the opportunity to transmit. Furthermore, since the phone transmits at about 25 mW [12], which is comparable to an 802.11 laptop, its interference continues even at distances as far as 90 feet.

The phone's spectrogram depicted in Fig. 2(a3) shows that the phone occupies about a 3-4 MHz wide band. Typically, the phone picks one channel out of 35 radio channels in the 2.407-2.478 GHz range. It stays on that channel as long as it does not experience persistent interference.

2.2 Baby Monitor

We experiment with the C-501 wireless monitoring toolkit, which has two units: a 2.4 GHz wireless camera that supports up to 4 different channels (i.e., 2.414 GHz, 2.432 GHz, 2.450 GHz and 2.468 GHz), and a wireless video receiver. For every interferer location, we measure the 802.11n throughput with the camera ON and OFF, and plot the results in Fig. 2(b1). The figure shows that the 802.11n client and AP could not establish a connection and, hence, could not exchange any packets for all tested locations.

We plot the time and frequency profile of the camera in Fig. 2(b2) and Fig. 2(b3). The frequency profile shows that the baby monitor occupies a relatively wide channel of 16 MHz. Further, the time profile shows that the camera transmits continuously, thus hogging the medium completely. These observations, compounded with the fact that the camera transmits at a fairly high power of 200 mW [2], explain the inability of 802.11n to obtain any throughput.

2.3 Microwave Ovens

We use the SHARP R-310CW microwave oven. Fig. 2(c1) shows the observed 802.11n average throughput for different placements of the microwave. The figure shows that when the microwave is one foot away (in location 1), 802.11n suffers a throughput reduction of 90%. The 802.11n throughput improves as the microwave is moved away from the AP and its client, and the throughput loss decreases to 35% at the farthest location from the 802.11 client.

To understand this behavior, we plot the microwave's power profile over time in Fig. 2(b2). The figure shows that the microwave exhibits a periodic ON-OFF pattern, where an ON period lasts for about 10 ms and an OFF period lasts for 6 ms. In addition, the microwave also exhibits a continuous low interference, as evident from the 10 dB increase in the noise level after the microwave was turned on. The microwave time profile explains its impact on 802.11n. Specifically, at distant locations in our testbed, 802.11n transmits during the OFF periods but refrains from transmitting during the ON periods because it senses the medium as occupied. As a result, the throughput loss in such locations is about 35%. In contrast, at close distances, the 10 dB increase in the noise level generated by the microwave creates substantial interference for 802.11n causing most packets to be dropped even during the OFF periods.

2.4 Frequency Hopping Bluetooth

Finally, we evaluate the interference generated by Bluetooth devices. Bluetooth uses frequency hopping across a 79 MHz band in the 2.402-2.480 GHz range, occupying 1 MHz at any point in time. The most common devices use class 2 Bluetooth which transmits at a relatively low power of 2.5 mW [5].

For each interferer location, we transfer a 100 MB file between two Google Nexus One phones. We plot in Fig. 3 the throughput obtained by our 802.11n devices, in the presence and absence of the Bluetooth traffic. The figure shows that except in location 1, which

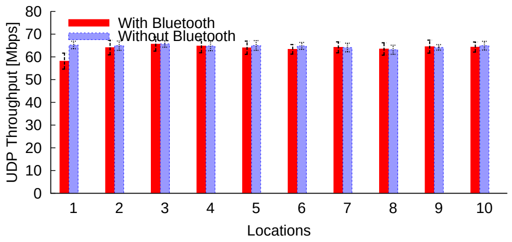


Figure 3—The impact of Bluetooth interference on 802.11n.

is one foot away from the 802.11n client, the Bluetooth exchange has no observable impact on the throughput of the 802.11n devices.

2.5 Summary

The above empirical study shows the following:

- High-power cross-technology interference can completely throttle 802.11n. Furthermore, loss of connectivity can occur even when the interferer is in a non-line-of-sight position and separated by 90 feet.
- While 802.11 and low-power interferers (e.g., Bluetooth) have managed a form of coexistence where both devices stay operational, coexistence with high-power devices (e.g., cordless phones, baby monitors, microwave, etc.) is lacking. Furthermore, the typical outcome of the interaction between 802.11n and a high-power interferer is that 802.11n either suffers a complete loss of connectivity or a significant throughput reduction. In §9 we show that even if carrier sense is deactivated, 802.11n continues to lose connectivity for many of the interferer's locations.
- Frequency isolation is increasingly difficult. Multiple of the studied interferers occupy relatively wideband channels of 16–25 MHz (e.g., camera and microwave). Moreover, these devices can occupy any band in the 802.11 spectrum. For example, both the cordless phone and the baby monitor have multiple channels that together cover almost the whole frequency range of 802.11.
- Finally, the characteristics of an interferer may change in time and frequency. The interferer may have ON-OFF periods, may move from one frequency to another, or change the width of the channel it occupies, like a microwave. This emphasizes the need for an agile solution that can quickly adapt to changes in the interference signal.

3. MIMO AND OFDM BACKGROUND

Consider the 2×2 MIMO system in Fig. 4. Say the sender transmits stream $s_1(t)$ on the first antenna, and $s_2(t)$ on the second antenna. The wireless channel linearly combines the signal samples corresponding to the two streams. Therefore, the receiver receives the following linear combinations on its two antennas:

$$\begin{aligned} y_1(t) &= h_{11}s_1(t) + h_{21}s_2(t) && (5) \\ y_2(t) &= h_{12}s_1(t) + h_{22}s_2(t), && (6) \end{aligned}$$

where h_{ij} is a complex number whose magnitude and angle refer to the attenuation and delay along the path from the i^{th} antenna on the sender to the j^{th} antenna on the receiver, as shown in Fig. 4. If the receiver knows the channel coefficients, h_{ij}, it can solve the above two linear equations to obtain the two unknowns, $s_1(t)$ and $s_2(t)$, and decode the two transmitted streams.

To enable the receiver to estimate the channel coefficients, h_{ij}, a MIMO sender starts each frame by transmitting a known preamble from each of its antennas, one after the other. The receiver uses its knowledge of the transmitted preamble and the received signal samples to compute the channel coefficients, which it uses to decode the rest of the bits in the frame.

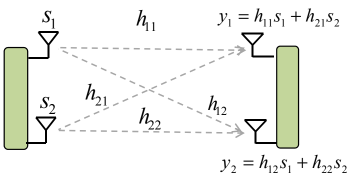


Figure 4—Decoding in a standard 2-by-2 MIMO system.

The above model assumes a narrowband channel, whose bandwidth is limited to a few MHz. In wideband channels, different frequencies may experience different channels. Thus, the channel function cannot be expressed as a single complex number; it has to be expressed as a complex filter, and the multiplication becomes a convolution:

$$\begin{aligned} y_1(t) &= \mathbf{h}_{11} * s_1(t) + \mathbf{h}_{21} * s_2(t) \\ y_2(t) &= \mathbf{h}_{12} * s_1(t) + \mathbf{h}_{22} * s_2(t), \end{aligned}$$

Modern wireless technologies like 802.11a/g/n, WiMax, and LTE handle such wide channels by operating on the signal in the frequency domain using OFDM. OFDM divides the channel frequency spectrum into many narrow subbands called OFDM subcarriers. The receiver takes an FFT of the received signal and operates on individual OFDM subcarriers, as if they were narrowband channels, i.e., the receiver applies the model in Eqs. 5 and 6 to the frequency domain signal, and decodes the transmitted symbols.

In 802.11, there are 64 OFDM subcarriers, four of which are called pilots that have a known symbol pattern to allow the receiver track the channel [24]. Additionally, 48 subcarriers are used to transmit data and the rest are unused for distortion reasons.

4. PROBLEM DOMAIN

TIMO deals with high power cross-technology interference in 802.11n networks. We focus on typical situations that arise in the operation of 802.11 networks. In particular,

- TIMO tackles scenarios in which the interferer is a single antenna device. This is typically the case for current 802.11 interferers, like baby monitors, microwave ovens, cordless phones, surveillance cameras, etc.
- TIMO applies to scenarios in which the interfering signal lasts more than a few seconds. This constraint does not necessarily mean that the interferer transmits continuously for that duration. For example, a microwave signal that lasts for a few seconds satisfies our constraint despite having OFF periods.
- TIMO applies to scenarios where, in the absence of an interferer, the 802.11n receiver can use MIMO multiplexing, i.e., it can receive multiple concurrent streams at some bitrate. If the 802.11n receiver cannot multiplex streams from the same technology, it cannot be made to multiplex streams from different technologies.
- TIMO can address environments with multiple concurrent interferers, as long as the interferers are in different frequencies (i.e., different 802.11 OFDM subcarriers). We believe this to be the common case in today's networks because the presence of multiple high-power interferers in the same band will cause them to interfere with each other, and is likely to prevent the proper operation of the device.

5. TIMO

TIMO extends the MIMO design to operate across diverse wireless technologies that may differ in modulation, coding, packet format, etc. It develops two primitives: The first primitive enables a

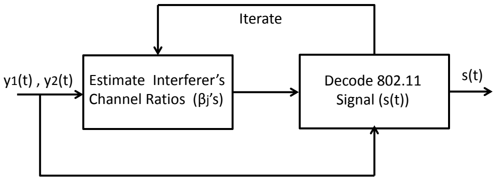


Figure 5—Flowchart of the different components.

MIMO 802.11n pair to exchange packets in the presence of an unknown interference signal, as if the unknown interference were a single-antenna 802.11 transmission. For example, an 802.11n AP-Client pair may use this primitive to correctly decode packets in the presence of the ON periods of a microwave oven. The second primitive enables a MIMO node to transmit in the presence of an unknown bi-directional technology without hampering reception at the receiver of the unknown technology. For example, an 802.11n node may use this primitive to transmit in the presence of a cordless phone without hampering the phone's operation. The next few sections describe these two primitives in detail.

6. DECODING IN THE PRESENCE OF CROSS-TECHNOLOGY INTERFERENCE

Consider a scenario in which two 802.11n nodes want to communicate in the presence of high-power cross-technology interference. For clarity, we will explain the design in the context of a 2-antenna 802.11n receiver decoding a single 802.11n transmission, in the presence of an interferer. The results extend to any number of antennas as we explain in the appendix.

In this case, the signal at the 2-antenna 802.11n receiver is the sum of the signal of interest, $s(t)$, and the interference signal, $i(t)$, after convolving them with their respective channels to the receiver:

$$y_1(t) = \mathbf{h}_i * i(t) + \mathbf{h}_s * s(t) \tag{7}$$

$$y_2(t) = \mathbf{h'}_i * i(t) + \mathbf{h'}_s * s(t), \tag{8}$$

where $\mathbf{h}_i$ and $\mathbf{h'}_i$ are the channel functions of the interference signal, and $\mathbf{h}_s$ and $\mathbf{h'}_s$ are channel functions of the signal of interest. We will explain TIMO's decoding algorithm assuming the receiver knows the channel of the signal of interest. In §6.4, we explain how the receiver obtains this channel in the presence of interference.

Since the signal of interest (i.e., that of 802.11n) is an OFDM signal, the receiver processes its input in the frequency domain by taking an FFT. Thus, for each OFDM subcarrier, j, the receiver obtains the following equations:

$$Y_{1j} = H_{ij}I_j + H_{sj}S_j \tag{9}$$

$$Y_{2j} = H'_{ij}I_j + H'_{sj}S_j, \tag{10}$$

where the terms in the above equations are the frequency version of the terms in Eqs. 7 and 8, for a particular OFDM subcarrier. Thus, the receiver can express the signal of interest as:

$$S_j = \frac{Y_{1j} - \beta_j Y_{2j}}{H_{sj} - \beta_j H'_{sj}} \quad \text{for } \beta_j = \frac{H_{ij}}{H'_{ij}}. \tag{11}$$

All terms in Eq. 11 are known at the receiver, except for β_j. The objective of the receiver is to figure out β_j in each subcarrier, and use it to decode the signal of interest, S_j, in that subcarrier.

A TIMO receiver has three main components shown in Fig. 5. 1) An algorithm for computing the interferer's channel ratio in an OFDM subcarrier without knowing the interferer's preamble or signal structure. 2) A decoder that allows the receiver to decode the signal of interest given the interferer's channel ratio in every OFDM subcarrier. 3) An iteration mechanism that reduces the noise in the computation of channel ratios, hence increasing SNR. The following sections describe these components.

6.1 Computing the Interferer's Channel Ratio

A simplistic approach for computing the ratio $\beta_j = \frac{H_{ij}}{H'_{ij}}$ would rely on that the signal S_j in the OFDM pilots is known to the receiver. Thus, if one assumes β_j is the same for all OFDM subcarriers, one can simply substitute the signal S_j, where j is a pilot subcarrier, in Eq. 11, and use that equation to compute the ratio β. The receiver then uses this ratio to compute signal values in other OFDM subcarriers that contain data symbols. However, the assumption that the interferer channel ratio is the same in all OFDM subcarriers is typically invalid for several reasons. First, there might be multiple interferers each of them operating in a different frequency band. For example, the interfering signal may be a combination of two cordless phone signals each occupying upto 4 MHz and overlapping with a different set of 802.11n OFDM subcarriers. Second, there might be an interferer that hops across the OFDM subcarriers, but does not always occupy all subcarriers. This is the case for the narrowband signal during the microwave ON period. Finally, the interferer may have a relatively wideband channel, like the baby monitor which can span upto 16 MHz. In this case, the channel of the interferer may differ across the OFDM subcarriers due to multipath and hence the channel ratio also changes across the subcarriers.

Thus, the receiver should compute the interferer's channel ratio for each OFDM subcarrier independently. Since most OFDM subcarriers carry data and contain no known patterns, the receiver has to compute this ratio without any known symbols.

Below we use Eqs. 9 and 10 to obtain a closed form expression for the interferer's channel ratio in each OFDM subcarrier. To do so, we first eliminate the contribution from the signal of interest S_j, by multiplying Eq. 10 with $\frac{H_{sj}}{H'_{sj}}$ and subtracting it from Eq. 9:

$$Y_{1j} - \frac{H_{sj}}{H'_{sj}}Y_{2j} = \left(\frac{H_{ij}}{H'_{ij}} - \frac{H_{sj}}{H'_{sj}}\right)H'_{ij}I_j$$

Next, we multiply the resulting equation with the conjugate of Y_{2j}, and take the expectation:

$$\begin{aligned}
E[(Y_{1j} - \frac{H_{sj}}{H'_{sj}}Y_{2j})Y^*_{2j}] &= \left(\frac{H_{ij}}{H'_{ij}} - \frac{H_{sj}}{H'_{sj}}\right)E[H'_{ij}I_jY^*_{2j}] \\
&= \left(\frac{H_{ij}}{H'_{ij}} - \frac{H_{sj}}{H'_{sj}}\right)E[H'_{ij}I_j(H'^*_{ij}I^*_j + H'^*_{sj}S^*_j)] \\
&= \left(\frac{H_{ij}}{H'_{ij}} - \frac{H_{sj}}{H'_{sj}}\right)(E[|H'_{ij}I_j|^2] + H'^*_{sj}H'_{ij}E[I_jS^*_j]) \\
&= \left(\frac{H_{ij}}{H'_{ij}} - \frac{H_{sj}}{H'_{sj}}\right)E[|H'_{ij}I_j|^2] \\
&= \left(\beta_j - \frac{H_{sj}}{H'_{sj}}\right)P'_{Ij}, \qquad (12)
\end{aligned}$$

where $|x|^2 = xx^*$ denotes the square of the amplitude of the complex number x, and $E[I_jS^*_j] = 0$ because the signal of interest is independent from the interference signal and hence their correlation is zero. Also $P'_{Ij} = E[|H'_{ij}I_j|^2]$ is the received interference power in OFDM subcarrier j on the second antenna of the 802.11n receiver.

Eq. 12 has two unknown β_j and P'_{Ij}. Thus, if the receiver knows the interferer's received power, P'_{Ij}, it can solve Eq. 12 to obtain the desired ratio. To compute P'_{Ij} , the receiver takes Eq. 10, multiplies it by its conjugate, and then computes the expectation, i.e.:

$$\begin{aligned}
E[Y_{2j}Y^*_{2j}] &= E[(H'_{ij}I_j + H'_sS_j)(H'_{ij}I_j + H'_sS_j)^*] \\
&= E[|H'_{ij}I_j|^2] + E[|H'_sS_j|^2] \\
&= P'_{Ij} + P'_{Sj}, \qquad (13)
\end{aligned}$$

where P'_{Sj} is the power of the signal of interest on the second antenna in the jth OFDM subcarrier. Again, to reach Eq. 13 we have exploited the fact that the interference signal and the signal of interest are independent of each other.

We can solve Eq. 12 and Eq. 13 together to obtain the ratio:

$$\beta_j = \frac{H_{ij}}{H'_{ij}} = \frac{E[(Y_{1j} - \frac{H_{sj}}{H'_{sj}} Y_{2j}) Y^*_{2j}]}{E[|Y_{2j}|^2] - P'_{Sj}} + \frac{H_{sj}}{H'_{sj}}. \tag{14}$$

This equation enables the 802.11 receiver to compute the interferer's channel ratio without any known symbols, simply by substituting the power and the channel ratio for $s(t)$.

It is important to note that the above derivation exploits that expectations can be computed by taking averages. The accuracy of this estimate increases as one averages over more signal symbols. In §6.3 we will discuss how we can obtain a good accuracy without averaging over many symbols.

6.2 Decoding the Signal of Interest

Once the 802.11n receiver has an estimate of the interferer's channel ratio, β_j, in each OFDM subcarrier, it proceeds to decode its own signal of interest. One way to decode would be to substitute β_j in Eq. 11 to compute S_j in the frequency domain. This approach works well when the interferer is a narrowband signal, like a cordless phone. However, it has low accuracy in scenarios the interferer has a relatively wideband channel, like a baby monitor that spans 16 MHz. This is because wideband signals suffer from multipath effects; i.e., the signal travels from the sender to the receiver along multiple paths with different delays. A wideband receiver receives the combination of multiple copies of the same signal with different relative delays. This leads to inter-symbol interference (ISI), which mathematically is equivalent to convolving the time-domain signal with the channel on the traversed paths.

To deal with ISI, an OFDM transmitter inserts a cyclic prefix between consecutive symbols. The receiver discards the cyclic prefix and takes the remaining signal, thus eliminating any interference from adjacent symbols. This, however, does not work when we have a wideband interferer like the baby monitor. First, its signal may not have a cyclic prefix. Second, even if it does, as noted by past work on concurrent 802.11n transmissions [35], it is unlikely that the cyclic prefixes of the two devices are synchronized, in which case the receiver cannot discard a single cyclic prefix that eliminates ISI for both the devices.

The above discussion means that in the frequency domain, the interferer's signal, I_j, will experience ISI which would add noise. As a result, Eqs. 9 and 10 have additional noise terms due to ISI. While this is not a problem for the channel ratio estimation since one can average across more samples to obtain an accurate estimate of β_j; this additional noise would reduce the SNR for the signal of interest and, hence, affect its throughput.

The solution to the ISI problem is, however, simple. The 802.11n receiver needs to decode the signal of interest $s(t)$ by eliminating interference in the time domain. Here, ISI is simply a convolution with a filter, which can be removed by applying the inverse filter (i.e., an equalizer). Thus, we consider again the initial time domain Eqs. 7 and 8 which describe the signal at the 802.11n receiver:

$$\begin{aligned} y_1(t) &= \mathbf{h}_i * i(t) + \mathbf{h}_s * s(t) \qquad (15) \\ y_2(t) &= \mathbf{h'}_i * i(t) + \mathbf{h'}_s * s(t), \qquad (16) \end{aligned}$$

We want to find a filter, **h**, such that:

$$\mathbf{h} * \mathbf{h'}_i = \mathbf{h}_i$$

Given such a filter, the receiver can convolve **h** with Eq. 16 and subtract the resulting equation from Eq. 15 to eliminate $i(t)$ and obtain an equation in $s(t)$, which it can decode using a standard 802.11 decoder.[4]

The above filter can be represented in the frequency domain as:

$$H_j H'_{ij} = H_{ij} \Rightarrow H_j = \frac{H_{ij}}{H'_{ij}} = \beta_j$$

Thus, we can compute the desired filter **h** by taking the IFFT of the interferer channel ratios, β_j's, computed in §6.1.

To summarize, the 802.11n receiver first moves the received signal to the frequency domain where it computes the interferer channel ratios using Eq. 14 while averaging over multiple samples to reduce the ISI and noise. Then, it transforms the interferer channel ratio into a time domain filter by taking an IFFT. Finally, it uses the filter to eliminate interference in the time domain. The receiver can now take this interference-free signal and decode its signal of interest using a standard 802.11 decoder.

6.3 Iterating to Increase Accuracy

The algorithm in §6.1 computes expectations by taking averages over multiple OFDM symbols. A packet, however, may not have enough OFDM symbols to obtain a highly accurate estimate. Also averaging over multiple packets will reduce TIMO's ability to deal with a dynamic interferer. Thus, in this section we are interested in obtaining an accurate estimate of the interferer's channel ratio, β_j, using only a few OFDM symbols.

To increase the accuracy of the estimate without much averaging, the receiver iterates over the following two steps:

Initialization: The receiver obtains a rough estimate of β_j by averaging over a limited number of OFDM symbols.

Step 1: The receiver uses its estimate of β_j to obtain the signal, $s(t)$, as in §6.2. The receiver then decodes $s(t)$ using the standard decoder to obtain the transmitted bits.

Step 2: The receiver re-modulates the decoded bits to obtain an estimate of $s(t)$, which we call $\hat{s}(t)$. The receiver convolves $\hat{s}(t)$ with the channel functions and subtracts the results from $y_1(t)$ and $y_2(t)$. Thus, we obtain the following:

$$\begin{aligned} \hat{y}_1(t) &= \mathbf{h}_i * i(t) + \mathbf{h}_s * (s(t) - \hat{s}(t)) \\ \hat{y}_2(t) &= \mathbf{h'}_i * i(t) + \mathbf{h'}_s * (s(t) - \hat{s}(t)). \end{aligned}$$

The receiver then obtains a new estimate for β_j while treating $(s(t) - \hat{s}(t))$ as the new signal of interest.

After iterating between Step 1 and 2 for two or three times, the receiver obtains an accurate estimate of the interferer's channel ratio β_j, which it uses to decode signal $s(t)$.

The reason why the above algorithm works is that in each iteration, the signal of interest used in Step 2, $(s(t) - \hat{s}(t))$, has a smaller magnitude. Since, in Step 2, the receiver is focused on estimating the interferer's ratio, the signal of interest plays the role of noise; reducing this signal's magnitude increases the accuracy of the ratio estimate. This higher accuracy in the ratio β_j percolates to the estimate of $s(t)$ in Step 1. Consequently, the decoded bits are more accurate and lead to even smaller difference between $\hat{s}(t)$ and $s(t)$, and hence an even more accurate β_j.

6.4 Estimating the 802.11n Channel Functions

So far, we have assumed that the 802.11n receiver knows the channel of the signal of interest, H_{sj} and H'_{sj}. To compute this channel we distinguish between two cases. First, the signal of interest

---

[4] As described in §3, such a decoder would apply FFT and decode in the frequency domain.

starts before the interference in which case the receiver can use the 802.11 preamble to compute the channel, as usual. Second, the interference signal starts before the signal of interest. In this case, the receiver can easily compute the interferer's channel ratio $\beta_j = \frac{H_{ij}}{H'_{ij}}$ by taking the ratio of the signals it receives on its two antennas $Y_{1j} = H_{ij}I_j$ and $Y_{2j} = H'_{ij}I_j$. Once the receiver knows the interferer's channel ratio, it computes the equalization filter described in §6.2 and uses it to eliminate the interference signal. The receiver can then use the 802.11n preamble to compute the channel as usual.

Two points are worth noting: First, while it is easy to compute the interferer's channel ratios when the interferer is alone on the medium, this does not eliminate the need to continue tracking the interferer's channel ratio using the algorithm in §6.1. In particular, the channel ratio may change as the interferer moves to a different frequency, as in the narrowband phase of a microwave signal, or it might change for a mobile interferer, as with the cordless phone.

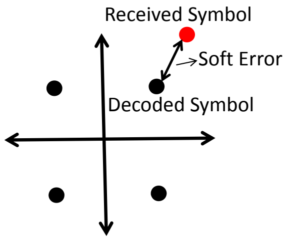
Second, the above scheme will miss in scenarios in which the interference and the 802.11n signal starts during the same OFDM symbol. This event has a low probability, and the resulting packet loss is minor in comparison to the packet loss observed without TIMO. When such an event occurs the packet will be retransmitted by its sender as usual.

6.5 Finding the Interference Boundaries

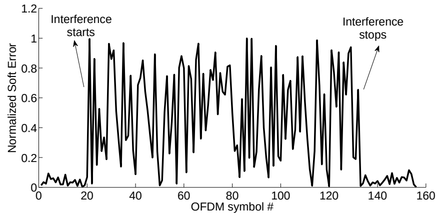
Estimating the interferer's channel ratio, β_j, using Eq. 14 requires the 802.11n receiver to compute the expectations by taking averages over multiple OFDM symbols. This averaging, however, needs to be done only over symbols that are affected by interference. Thus, the 802.11n receiver needs to determine where, in a packet, interference starts and where it stops. The question of identifying the sequence of symbols affected by interference has been addressed in few recent systems, like PPR [25] and SoftRate [37]. Our approach follows the same principles. Specifically, when the interference signal starts, it causes a dramatic increase in decoding errors. As shown in Fig. 6(a), these errors appear at the PHY layer as large differences between the received symbol and the nearest constellation points in the I and Q diagram. We refer to these differences as soft errors. Thus, for each OFDM subcarrier, the 802.11n receiver computes the soft-error, and normalizes it by the minimum distance of the constellation. As shown in Fig. 6(b), when the interferer starts, the soft errors jumps; when it ends, they go back to their low values. In our implementation we consider a jump that is higher than doubling the errors as a potential interferer, i.e., interference above 3 dB. This means that we might miss low power interferers, but such interferers can be dealt with using traditional methods like reducing the bit rate.

6.6 Putting it together

A TIMO receiver first performs packet detection as usual by looking for jumps in received power (using standard window detection algorithms [24]). Then, the receiver computes the 802.11 preamble cross-correlation, in a manner similar to current 802.11. If the cross-correlation stays low, the receiver works under the assumption that the signal of interest may start later. Hence, it computes the channel ratios for the signal though it is not its signal of interest. On the other hand, if the cross-correlation spikes, the receiver identifies the packet as a signal of interest. It continues decoding the packet using a standard 802.11 decoder [15]. If the packet does not pass the checksum test, the receiver computes the soft-errors as described in §6.5. If the soft-errors jump by over 3 dB, the receiver initiates the channel ratio estimation algorithm. Specifically, for each OFDM bin, the TIMO decoder starts at the symbol where the soft errors jump and proceeds to compute the interference channel ratios in an iterative manner as described in §6.3. Once the channel ratios are estimated for each OFDM subcarrier, the receiver uses the decoder in §6.2 to decode its signal of interest.



(a) Soft-errors in a 4QAM Constellation.



(b) Soft-errors with Interference.

Figure 6—Soft errors increase in the presence of interference.

6.7 Complexity

While past work that deals with cross-technology interference [6, 34] typically employs different mechanisms for different technologies, TIMO is technology agnostic and hence its complexity stays constant as the number of technologies in the ISM band increases. Further, the components used in TIMO such as correlation, equalization and projection, are also used in MIMO receivers (though for a different purpose), and hence are amenable to hardware implementations.

7. ENSURING THE INTERFERER CAN DECODE

A MIMO transmitter can also encode its signal to prevent interference to a competing transmission from a different technology. Specifically, let $i(t)$ be the competing signal and $s_1(t)$ and $s_2(t)$ the two streams that a 2-antenna 802.11n node transmits. The receiver of the competing signal receives the following:

$$z(t) = h_i i(t) + h_{s1}s_1(t) + h_{s2}s_2(t), \tag{17}$$

where h_i refers to the channel from its transmitter and h_{s1} and h_{s2} are the channels from the 2-antenna 802.11n transmitter. The 802.11n transmitter can cancel its signal at the receiver of the competing technology by ensuring that the signals it transmits on its two antennas satisfy $s_2(t) = -\frac{h_{s1}}{h_{s2}}s_1(t)$. Such a technique is typically referred to as interference nulling [36].[5]

We note that nulling does not require the knowledge of the exact channels to the receiver. It is sufficient to know the channel ratios to null the signal at some receiver. This is crucial since for cross-technology scenarios, it is hard to estimate the exact channel.

But how does the 802.11n transmitter compute the channel ratio to the interferer's receiver? If the interfering technology is bidirectional in the frequency of interest, then our 802.11n nodes can use the interference caused by the receiver's response to compute the channel ratio from the receiver to itself. This can be done by leveraging the algorithm in §6.1. The required ratio for nulling, however, refers to the channels in the opposite direction, i.e., from our 802.11n transmitter to the interfering receiver. To deal with this

---

[5] Note that having the 802.11n transmitter perform interference nulling does not require any modification to decoding at the 802.11n receiver.

issue, TIMO exploits that wireless channels exhibit reciprocity, i.e., the channel function in the forward and backward direction is the same. Reciprocity is a known property that has been validated empirically by multiple studies [21, 39, 28].[6] Using reciprocity one can compute the required channel ratio. Once the ratio is computed, the transmitter can perform interference nulling. We note that since it is hard to synchronize wideband cross-technology interferers with 802.11, to avoid ISI we perform nulling by using a time-domain equalizer similar to §6.2.

Thus, interference nulling combined with our algorithm for estimating the interferer's channel ratio provide a new primitive that enables a MIMO node to transmit in the presence of a different technology without hampering reception of that technology. This primitive, however, requires the competing technology to be bidirectional, i.e., the competing receiver acks the signal or transmits its own messages, like a cordless phone.

If the technology is bidirectional, then the MIMO transmitter can learn the channel ratio to the communicating node pair, using the interference they create. The MIMO transmitter then alternates between nulling its signal at the two communicating nodes. For example, in the case of a cordless phone, the 802.11 transmitter has to switch between nulling its signal at the handset and nulling its signal at the base. In the case of the cordless phone, the switching time is constant, and for the tested phone it is 2.25 ms. Even if the switching time is not constant, as long as the pattern of the interference is persistent (e.g., one data packet, followed by one ack), the MIMO node can monitor the medium and immediately switch every time the medium goes idle.

On the other hand, if the receiver of the competing technology is not bidirectional, an 802.11n device has no way to compute its channel ratio, and hence cannot cancel its signal at the receiver of the competing technology. The impact of such interference will depend on the competing technology. For example, interference does not hamper a microwave oven function. Also, analog devices (e.g., an analog camera) have some level of resistance to interference which causes smooth degradation in their signal, and while they suffer from interference, they can still function if the interferer is not in close proximity (see §9).

In general, our objective is to create a form of coexistence between 802.11n and high-power interferers that approaches the coexistence it enjoys with low-power devices like Bluetooth, where the two technologies may interferer if they are in close proximity but the interference is limited and does not cause either device to become completely dysfunctional. Unidirectional devices which do not sense the medium or use any feedback from their receiver tend to show some level of resistance to interference. Hence, even if the 802.11n node did not cancel its interference at their receiver, they can still support some level of coexistence, as long as 802.11n can protect itself from their interference.

8. IMPLEMENTATION

We have built a prototype of TIMO using the USRP2 radio platform and the GNURadio software package. A 2×2 MIMO system is built using two USRP2 radio-boards connected via an external clock [9]. Each USRP2 is configured to span a 10 MHz channel by setting both the interpolation rate and decimation rate to 10. The resulting MIMO node runs a PHY layer similar to that of 802.11n, i.e., it has 64 OFDM subcarriers, a modulation choice of BPSK, 4QAM, 16QAM, or 64QAM, and punctured convolution codes with standard 802.11 code rates [15]. Since we operate at half the 802.11 bandwidth, the possible bit rates span 3 to 27 Mbps.

We modify the receiver MIMO decoding algorithm to incorporate TIMO (summarized in §6.6). We also implemented interference nulling at the MIMO transmitters. To work with cross-technology interference, the transmitter first computes the channel ratios and then uses them for nulling (as described in §7).

9. PERFORMANCE EVALUATION

We evaluate TIMO with three high-power interferers: a DSSS cordless phone, a microwave oven, and a baby monitor.

9.1 Cordless Phone

Again, we use the Uniden TRU 4465-2 cordless phone as the interferer. We also use the same testbed in Fig. 1.

Addressing Cross-Technology Interference: We first evaluate TIMO's ability to help 802.11n nodes operate in the presence of high power cross-technology interference. We place two USRP-based 802.11n nodes in locations A and B in Fig. 1. In each run, we place the cordless phone system in one of the 10 interferer locations in Fig. 1. We transfer a 20 MB file between the 802.11n pair at the best bitrate for the channel in the presence of interference from the cordless phone. This rate is determined by initially trying all the possible bitrates and choosing the one which yields the highest throughput for the rest of the run. The 802.11 receiver logs the received samples and processes them both with and without TIMO.

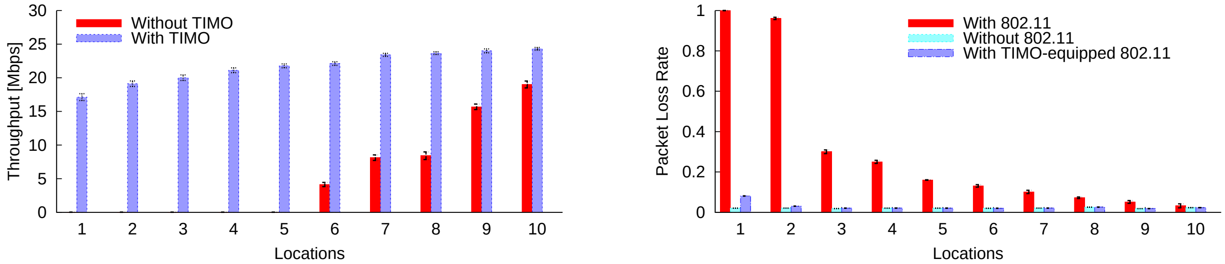
Note that in contrast to the experiments done with commercial 802.11n nodes, the USRP implementation of 802.11n does not use carrier sense. Carrier sense is hard to implement in software due to its strict timing requirements. This constraint, however, can be beneficial. In particular, the lack of carrier sense provides insight into whether the throughput loss of commercial 802.11n is due to the nodes sensing the phone's signal and abstaining from transmitting, or due to their packets being corrupted by interference.

Fig. 7(a) plots the throughput of the 802.11 MIMO nodes in the presence of the phone signal, with and without TIMO. The figure reveals the following:

- Without TIMO, interference from the cordless phone causes the 802.11 nodes to completely lose connectivity in half of the testbed locations. This loss of connectivity occurs even though the nodes have deactivated carrier sense and are using the best bit rate for the channel. This means that the interference in these locations is too high even for the lowest bit rate supported by 802.11. This loss in connectivity can be attributed to the fact that the phone system transmits continuously at a high power. Hence, the 802.11 packets are always subject to strong interference. As the interferer moves away from the 802.11 USRP-based nodes, their throughput improves because of reduced interference.
- In contrast, with TIMO, the 802.11 nodes never experience disconnectivity. Also, their throughput becomes much higher and close to optimal (24.5Mbps) at most locations. The throughput decreases slightly as the phone moves closer to the 802.11 receiver in location B because of residual interference, but continues to be 78% of the optimal throughput even when the phone is one foot away from the 802.11 receiver. These results indicate that TIMO is successful at exploiting MIMO capability to address 802.11 cross-technology interference.
- Comparing the throughput of the USRP-based 802.11n implementation to that of commercial 802.11n in §2 shows that while carrier sense contributed to the loss of connectivity particularly

---

[6] To use it in our system, one needs to calibrate the effect of the hardware before applying reciprocity. This calibration, however, is done once for the hardware. Furthermore, an 802.11n transmitter can perform this task without the help of any other node because it merely involves taking the difference between the two transmit chains attached to its two antennas.



(a) 802.11 throughput with and without TIMO in the presence of interference from a DSSS phone.

(b) Packet loss at the DSSS phone with and without TIMO.

Figure 7—Interference from a DSSS Cordless Phone: Figure (a) shows that TIMO significantly improves the throughput of 802.11 USRP2-based nodes in the presence of interference from a DSSS phone. Figure (b) shows that if 802.11 nodes transmit concurrently with a DSSS cordless phone, they can cause the phone a dramatic packet loss at close distances. TIMO, however, enables such nodes to transmit concurrently with the phone without hampering its performance.

when the interferers are in locations 6–10, it is not the main reason since even though the USRP nodes do not implement carrier sense, they still lose connectivity in 50% of the locations.

Transmitting without Harming the Competing Technology: Next, we evaluate TIMO's ability to allow 802.11n to transmit concurrently with a cordless phone in the same frequency band, but without harming the phone's transmission. The commercial phone does not give us access to packets, making it hard to evaluate the impact of TIMO's interference nulling. Instead, we implement the phone's physical layer in GNURadio and experiment with a USRP-based DSSS phone. We try to match the physical layer description of the Uniden phone. In particular, the transmitter feeds digital bits to a scrambler, differential encoder, and a spread spectrum module. The spread spectrum module sends bits at a data rate of 1.366 Mbps over FSK modulation. The receiver computes the correlation with the spreading code and outputs the data bits. For every packet we use the CRC to detect if it was correctly received.

We place the USRP nodes that perform the role of the phone base and handset at location A and B in the testbed. We then place a 802.11 USRP transmitter at each of locations 1 to 10 in the testbed, and let it transmit at the same time as the USRP phone. The 802.11 USRP transmitter uses TIMO to null its signal at the phone.

The 802.11 transmitter has to alternate between nulling its signal at the phone base and the handset. Since the Uniden phone packets have a fixed duration of 2.25 ms [12], this switching can easily happen on 802.11 hardware. However, due to the software nature of GNURadio, it is hard to alternate with the phone system at a granularity of about 2.25 ms. Thus, in our experiments, we increase the inter-packet time and the packet duration to 20 ms, which allows us to alternate with the phone system in software.

Each run of the experiment has three parts. First, the phone handset and base exchange packets without any interference from the 802.11n transmitter. Next, the handset and base exchange packets with interference from the 802.11 node but without TIMO. Finally, the handset and base exchange packets concurrently with the 802.11n node which uses TIMO.

Fig. 7(b) shows the packet loss rate at the handset for the above three cases. The figure shows three main trends.

- In comparison with 802.11n, the DSSS phone is more resilient to cross-technology interference. This is due to its use of FSK combined with a high redundancy DSSS code. Despite this resilience, without TIMO, the phone suffers a high loss rate at locations close to the 802.11 nodes.
- In contrast, TIMO significantly reduces the loss rate at the handset across all the locations. Further, in locations 2-10 the loss rate is almost as low as that without any interference. We note that this is true even for locations where the interferer is closer to the handset than the base is to the handset (locations 2-4). Thus, we conclude that TIMO can help 802.11 and DSSS phones coexist.
- Finally, when the 802.11 interferer is less than a foot from the handset (location 1), the packet loss rate is higher than that without interference. This is because, in practice, it is difficult to completely eliminate interference using interference nulling. The residual interference may cause an increase in packet loss rate at such close distances. However, even at location 1, while TIMO did not completely eliminate interference, it still dramatically reduces the packet losses by more than 14x, from 100% to about 6-7%.

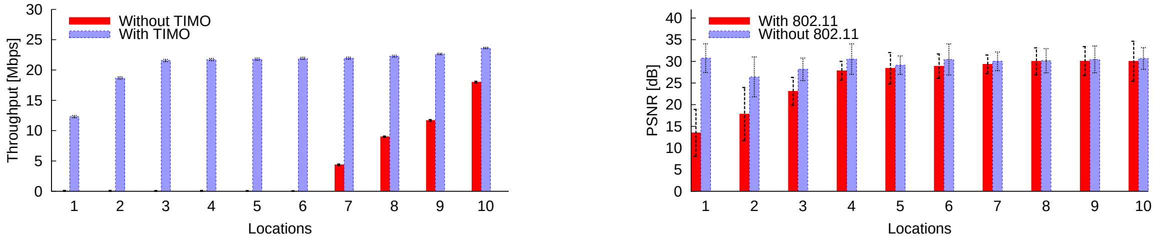
9.2 Baby Monitor

Next, we evaluate TIMO with a baby monitor.

Impact of baby monitors on 802.11n: To evaluate this, we repeat the previous experiment after replacing the microwave with the C-501 baby monitor. For every interferer location, we run the system with and without TIMO, and plot the results in Fig. 8(a). The figure shows that TIMO significantly increases the throughput in the presence of interference from the tested baby monitor. In particular, without TIMO the 802.11 nodes experience complete disconnectivity for 60% of locations of the baby monitor. In contrast, with TIMO no scenario causes disconnectivity and the overall throughput is significantly higher. We note that in comparison to the performance of commercial 802.11n nodes, the USRP-based 802.11n implementation does not use carrier sense, and hence was able to transmit and obtain some throughput in scenarios where the commercial 802.11n nodes refrained from transmitting due to carrier sense.

Impact of 802.11n transmissions on baby monitors: Communication in the baby monitor system is one-way. The camera continuously broadcasts the analog video. A monitor in range of the device receives the signal, decodes it and displays it on its screen. Given no signal from the video receiver, TIMO is limited in its ability to protect the transmitted video. Thus, we would like to check how the camera is affected by interference from our 802.11 implementation (which use the same power level as a laptop, i.e., about 30 mW).

To do so, we place the camera and its video receiver in locations A and B in the testbed. We move the 802.11-USRP node across the various interferer locations, and at each location, we ensure it interferes with the camera's transmission. We compare the received video quality with and without interference from 802.11. We measure video quality using PSNR, which is a standard video metric. A PSNR of less than 20 dB is hard to watch, whereas PSNRs in the range 25–30 dB are good. The PSNR can be computed only with respect to the original video. However, the camera does not provide us access to the original video before transmission over the wireless medium. To obtain a video baseline, we focus the camera on



(a) 802.11 throughput with and without TIMO in the presence of interference from a baby monitor.

(b) Camera PSNR. (above 20 dB is watchable; above 25 is good [40]).

Figure 8—Interference from a Baby Monitor: Figure (a) shows that TIMO significantly improves the throughput of 802.11 nodes in the presence of interference from a baby monitor. Figure (b) shows that while TIMO cannot cancel its signal at the camera's receiver because it use a unidirectional communication, the impact of interference on the camera's signal is watchable in all locations but the two closest to the 802.11 nodes.

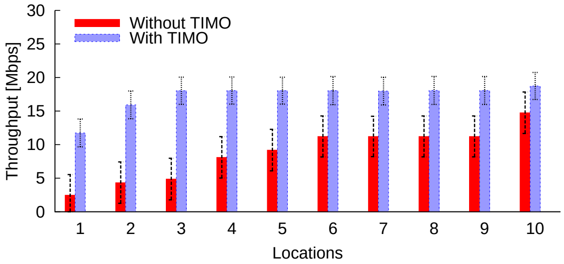


Figure 9—802.11 throughput with interference from a Microwave Oven: The figure shows that TIMO increases resilience to microwave interference.

a static image for all experiments, and make it transmit the same frame 1000 times. Then, we take the average pixel value in these 1000 versions of the same frame and consider this to be the ground truth. All experiments are run with the camera focused on the same picture so that they can be compared with this ground truth.

Fig. 8(b) shows the PSNR of the received video both with and without interference from our USRP-based 802.11 implementation. The figure shows that at the closest two locations, which are less than 6 feet away from the 802.11 interferer, the video is not watchable. However, for the rest of the locations, the video quality stays watchable. Further, for seven out of the ten testbed locations, the video PSNR hardly changes from its value without interference. This is expected because devices that blast the medium without checking for interference or without any feedback tend to be relatively resilient to some level of interference.

We note that since the monitoring system is uni-directional, TIMO cannot cancel its signal at potential video receivers; hence, we observe that interference degrades the monitoring system's performance at nearby locations. However, in contrast to the current mode of operation, where 802.11 loses connectivity in most locations due to interference, TIMO is an improvement over the status quo because it reduces the range of interference to close-by locations. This moves the system to a scenario where the two technologies enjoy some level of coexistence, which despite being far from optimal, is more acceptable than the current situation.

9.3 Microwave Oven

We evaluate TIMO's performance in the presence of interference from the microwave oven used in the experiments in §2. We repeat the experiment we conducted with the cordless phone, where we place the USRP-based 802.11 devices in locations A and B, and let them exchange traffic with the microwave on and off. We perform the experiment for each of the ten interferer locations in the testbed. In each run, the 802.11 transmitter uses the best bitrate as in §9.1.

Fig. 9 shows the average throughput and standard deviation, with and without TIMO. Without TIMO, the performance of the USRP2 nodes is relatively similar to that of the commercial 802.11n nodes. Specifically, at short distances, the throughput is very low due to increased interference. As the microwave is moved away, the nodes start getting packet through during the OFF periods of the microwave. In contrast, TIMO significantly increases resilience to interference from the microwave, allowing the 802.11 USRP node to deliver packets efficiently even during the ON periods of the microwave. Microwave ovens leak significantly high power during the ON periods, which could reach 1 Watt [17]. The results show that TIMO is effective even with such high-power interferers.

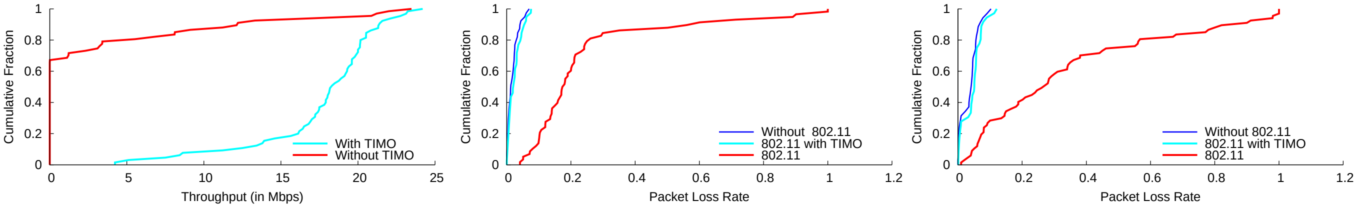
TIMO's approach is based on treating cross technology interference as if it were a stream from a single-antenna node of the same technology. Residential microwave ovens are equipped with a cavity magnetron which radiates energy in the 2.4 GHz range. Since they have only one magnetron radiating energy, theory concludes that they act as a single antenna device [34]. Our results confirm theoretical conclusions and show that TIMO can successfully treat a microwave as a single-antenna interferer.

9.4 Multiple Interferers

This experiment includes three node pairs with different transmission technologies: our 2×2 802.11n implementation, our DSSS phone implementation, and a GNURadio ZigBee implementation. The 802.11n devices occupy a 10 MHz channel, the DSSS phone occupies a 4 MHz channel, and the ZigBee devices occupy 5 MHz. The center frequencies of these devices are picked such that the phone interferes with the first half of the 802.11 channel, whereas the ZigBee device interferes with the second half. We place these six nodes randomly at the marked locations in Fig. 1. We make the three pairs transmit concurrently, and we repeat each run with and without TIMO. As before, we make the inter-packet arrival and the packet duration for the cordless phone and ZigBee nodes 20 ms, to allow for a software implementation.

Fig. 10(a) plots the CDF of 802.11 throughput with and without TIMO. The figure shows that without TIMO, about 67% of the locations cannot get any packets through and the average throughput is low. In contrast, with TIMO no locations suffer disconnectivity and the average throughput increases significantly.

Fig. 10(b) and 10(c) plot the packet loss rate of the competing technologies: the DSSS phone and ZigBee. The figure shows that if 802.11n transmits concurrently, without TIMO, these technologies can suffer significant packet loss. However, if 802.11n employs TIMO, then its interference increases loss rates by less than 0.5%, which is negligible. Thus, TIMO can help diverse technologies coexist in the same frequency band while placing the burden of interference prevention on high-end MIMO nodes instead of low-end single antenna systems.



(a) 802.11 Throughput. (b) Zigbee Throughput. (c) DSSS Phone Throughput.

Figure 10—TIMO with Multiple Interferers. The figure shows the throughput CDFs for three technologies that are transmitting concurrently in overlapping frequencies: 802.11n, DSSS phone, and ZigBee.

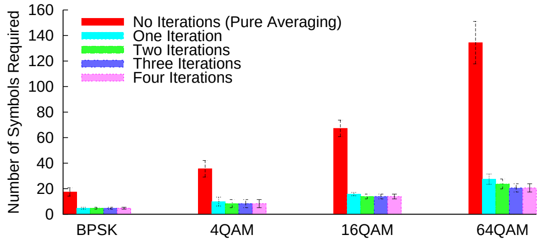


Figure 11—Tradeoff Between the Number of Averaged Symbols and the Number of Iterations: With three iterations, TIMO can achieve the same accuracy as a baseline that knows the structure and the preamble of the interferer, while maintaining the averaged symbols less than 22 for all modulations.

10. MICRO BENCHMARKS

Finally, we zoom in on the components of TIMO to examine the tradeoff between averaging over a larger number of symbols and applying the same algorithm iteratively over a smaller number of symbols.

We transfer a 20 MB file between two 2 × 2 802.11 USRP2 nodes. A third USRP2 node plays the role of an unknown interfering technology, and transmits a signal unknown to the 802.11 USRP2 nodes. We run the experiment for random placement of the three nodes in various locations in Fig. 1. We want to compute the amount of averaging and the number of iterations that TIMO needs to obtain an accurate estimate of the interferer's channel ratio. To obtain a ground truth of the channel ratios, we provide a baseline receiver with the full knowledge of the transmitted interference signal so that it can use the whole signal as if it were a preamble, and compute a very accurate estimate of the interferer's channel. We compute this estimate over periods of 1 ms each, which is significantly lower than the coherence time for indoor static channels at 2.4 GHz. For each run, we process the signal using the baseline receiver and TIMO.

Fig. 11 plots the number of symbols that TIMO needs to average over to obtain an estimate of the channel ratio that is within 3% of the value obtained with the baseline. The figure shows the results for the four modulations in 802.11 (BPSK, 4QAM, 16QAM and 64QAM). The plots reveal the following trends.

- The iterative algorithm yields a significant reduction in the number of symbols required to average over to obtain an accurate estimate of the interferer's channel ratio.
- Across all modulation schemes, two to three iterations are sufficient, and the return from more iterations is negligible. The reason why there is a ceiling for the iteration gain is that iterating does not provide more information; it only provides a better estimation using the collected information. After some point, the algorithm becomes limited by the intrinsic noise in the collected measurements.
- Given three iterations, TIMO needs to average over less than 22 symbols even at the highest modulation scheme.

11. RELATED WORK

Wireless interference has been the topic of much recent research. Work in this area falls under two broad categories:

(a) Interference Across Technologies: One can identify three main approaches within this category. The first approach attempts to eliminate interference by isolating the signals in time, frequency or space. The most common isolation approach is to employ frequency-based isolation, such as OFDM subcarrier suppression [30, 32, 23], variable channel width [19], or other fine grained frequency fragmentation techniques [38, 18, 31]. TIMO, on the other hand, enables independent technologies to share the same frequencies without interfering with each other. Directional antennas may also be used to provide spatial isolation and reduce interference. However, directional antennas are difficult to use in indoor scenarios where the signal tends to bounce off walls and furniture and scatter around [36]. In contrast, TIMO works in scattering environments and applies even when the two receivers are in the same direction.

The second approach uses mitigation schemes to modify transmissions to be more resilient to interference (e.g. by using coding or by lowering the bit rates). Mitigation proposals like PPR [25] and MIXIT [27], though designed and evaluated for the same technology, can work across technologies. These schemes however assume interference is fairly transient and limited to some bytes in each packet. In contrast, TIMO can deal with persistent interference.

Finally, some proposals identify the type of interference (is it ZigBee? Bluetooth?) and inform the user so he may switch off the interfering device [6, 29]. Others leverage the specific characteristic of a particular technology to design a suitable coexistence strategy [34]. Like this prior work, TIMO aims to provide coexistence of different wireless technologies. TIMO provides a single approach that works with different technologies, e.g., microwave ovens, cordless phones, etc, and applies even to unknown technologies.

(b) Interference from the Same Technology: Recent work in this category include interference cancellation [22], ZigZag [20] and analog network coding [26] which address the problem of interference from other 802.11 nodes. The closest to ours is prior work on MIMO systems which enables multiple transmitters to transmit concurrently without interference. This includes schemes like SAM [35], Interference Alignment and Cancellation [21], and beamforming systems [16]. Unlike these schemes, however, TIMO delivers a MIMO system that enables cooperation with multiple different wireless technologies.

Finally, TIMO is related to prior work on interference management in cellular networks, which uses multiple antennas to mitigate interference from nodes operating in adjacent cells [36, 4]. In contrast to this work, however, TIMO develops new algorithms that can address cross-technology interference.

12. CONCLUSION

This paper presents TIMO, a MIMO design that enables 802.11n to communicate in the presence of high-power cross-technology interference. TIMO exploits 802.11n's MIMO capability to treat a high-power signal from a different technology as if it were another stream from the same technology, hence enabling diverse technologies to share the same frequency band. We show via a proof-of-concept implementation that TIMO enables 802.11n to communicate effectively in the presence of typical interferers. Beyond 802.11n, we believe that TIMO provides the first step for a new form of coexistence, in which different technologies do not necessarily have to find unoccupied bands and could, in crowded environments, occupy the same band, thus increasing spectral efficiency.

Acknowledgments: We thank Nabeel Ahmed, Arthur Berger, Nate Kushman, Kate Lin, Hariharan Rahul, and Lili Qiu for their insightful comments. This research is supported by NSF Grants CNS-0831660 and CNS-0721857, and DARPA ITMANET.

13. REFERENCES

[1] 20 Myths of Wi-Fi Interference: Dispel Myths to Gain High-Performing and Reliable Wireless, White paper C11-44927.1-00, Cisco, 2007.

[2] 2.4GHz 4-Channel Wireless Receiver and 4 Wireless Infrared Color Cameras, Genica. www.genica.com.

[3] AirMaestro Spectrum Analysis Solution, Bandspeed. www.bandspeed.com.

[4] ArrayComm. www.arraycomm.com.

[5] Bluetooth Basics, Bluetooth SIG Inc., 2011. www.bluetooth.com.

[6] Cisco CleanAir Technology, Cisco. www.cisco.com/en/US/netsol/ns1070/index.html.

[7] Estimating the Utilisation of Key License-Exempt Spectrum Bands, Final Report REP003, Mass Consultants Ltd., Ofcom, April 2009.

[8] Evaluating Interference in Wireless LANs: Recommended Practice, White paper FPG 2010-135.1, Farpoint Group Technical Note, 2010.

[9] Fury GPS Disciplined Oscillator, Jackson Labs. www.jackson-labs.com.

[10] Miercom: Cisco CleanAir Competitive Testing, Lab Test Rerpot DR100409D, Miercom, 2010.

[11] Motorola Airdefense Solutions, Motorola. www.airdefense.net.

[12] Uniden TRU4465: Dual Handset Powermax 2.4GHz Cordless Systems, Uniden. www.uniden.com.

[13] Universal Software Radio Peripheral, Ettus Inc. www.ettus.com.

[14] Wireless RF Interference Customer Survey Result, White paper C11-609300-00, Cisco, 2010.

[15] Local and metropolitan area networks–specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. *IEEE Std 802.11*, 2009.

[16] E. Aryafar, N. Anand, T. Salonidis, and E. W. Knightly. Design and Experimental Evaluation of Multi-user Beamforming in Wireless LANs. In *Proc. ACM MobiCom*, 2010.

[17] Bandspeed. Understanding the Effects of Radio Frequency (RF) Interference on WLAN performance and Security, 2010.

[18] L. Cao, L. Yang, and H. Zheng. The Impact of Frequency-Agility on Dynamic Spectrum Sharing. In *Proc. IEEE DySPAN*, 2010.

[19] R. Chandra, R. Mahajan, T. Moscibroda, R. Raghavendra, and P. Bahl. A Case for Adapting Channel Width in Wireless Networks. In *Proc. ACM SIGCOMM*, 2008.

[20] S. Gollakota and D. Katabi. Zigzag Decoding: Combating Hidden Terminals in Wireless Networks. In *Proc. ACM SIGCOMM*, 2008.

[21] S. Gollakota, S. D. Perli, and D. Katabi. Interference Alignment and Cancellation. In *Proc. ACM SIGCOMM*, 2009.

[22] D. Halperin, J. Ammer, T. Anderson, and D. Wetherall. Interference Cancellation: Better Receivers for a New Wireless MAC. In *Proc. ACM HotNets*, 2007.

[23] Y. He, J. Fang, J. Zhang, H. Shen, K. Tan, and Y. Zhang. MPAP: Virtualization Architecture for Heterogenous Wireless APs. In *Proc. ACM SIGCOMM*, 2010.

[24] J. Heiskala and J. Terry. *OFDM Wireless LANs: A Theoretical and Practical Guide*. Sams Publishing, 2001.

[25] K. Jamieson and H. Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *Proc. ACM SIGCOMM*, 2007.

[26] S. Katti, S. Gollakota, and D. Katabi. Embracing Wireless Interference: Analog Network Coding. In *Proc. ACM SIGCOMM*, 2007.

[27] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard. Symbol-Level Network Coding for Wireless Mesh Networks. In *Proc. ACM SIGCOMM*, 2008.

[28] J. Ketchum, S. Nanda, R. Walton, S. Howard, M. Wallace, B. Bjerke, I. Medvedev, S. Abraham, A. Meylan, and S. Surineni. System Description and Operating Principles for High Throughput Enhancements to 802.11, QUALCOMM Inc., 2005.

[29] K. Lakshminarayanan, S. Sapra, S. Seshan, and P. Steenkiste. RFDump: An Architecture for Monitoring the Wireless Ether. In *Proc. CoNEXT*, 2009.

[30] S. Mishra, R. Brodersen, S. Brink, and R. Mahadevappa. Detect and Avoid: An Ultra-Wideband/WiMAX Coexistence Mechanism. *IEEE Communications Magazine*, 2007.

[31] T. Moscibroda, R. Chandra, Y. Wu, S. Sengupta, P. Bahl, and Y. Yuan. Load-Aware Spectrum Distribution in Wireless LANs. In *Proc. IEEE ICNP*, 2008.

[32] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat. Learning to Share: Narrowband-Friendly Wideband Networks. In *Proc. ACM SIGCOMM*, 2008.

[33] SourceForge. iperf.sourceforge.net.

[34] T. Taher, M. Misurac, J. LoCicero, and D. Ucci. Microwave Oven Signal Modelling. In *Proc. IEEE WCNC*, 2008.

[35] K. Tan, H. Liu, J. Fang, W. Wang, J. Zhang, M. Chen, and G. M. Voelker. SAM: Enabling Practical Spatial Multiple Access in Wireless LAN. In *Proc. ACM MobiCom*, 2009.

[36] D. Tse and P. Vishwanath. *Fundamentals of Wireless Communications*. Cambridge University Press, 2005.

[37] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-Layer Wireless Bit Rate Adaptation. In *Proc. ACM SIGCOMM*, 2009.

[38] L. Yang, W. Hou, L. Cao, B. Y. Zhao, and H. Zheng. Supporting Demanding Wireless Applications with Frequency-Agile Radios. In *Proc. USENIX NSDI*, 2010.

[39] P. Zetterberg. Experimental Investigation of TDD Reciprocity-Based Zero-Forcing Transmit Precoding. *EURASIP J. Adv. Signal Process*, 2010.

[40] H. Zhao, Y. Q. Shi, and N. Ansari. Hiding Data in Multimedia Streaming over Networks. In *Proc. CNSR*, 2010.

APPENDIX

Generalization to any number of antennas. Let M be the number of antennas at the 802.11 receiver. Say, there are K concurrent 802.11n transmissions, $s_1(t) \cdots s_K(t)$ whose channels are known at the receiver. We would like to estimate the interferer's channel in the presence of these K transmissions. Let, h_j^k be the channel coefficient of the kth transmission at the jth antenna on the receiver. Similarly, let h_j denote the channel of the interferer to the jth antenna on the receiver.

First, we note that one can always set h_1 to one. This can be done by considering the interferer to be the scaled value, $h_1 i(t)$, instead of $i(t)$. Thus, the received equation on the jth antenna is given by,

$$y_1(t) = i(t) + \sum h_1^k s_k(t)$$

$$y_j(t) = h_j i(t) + \sum h_j^k s_k(t), \forall j \neq 1$$

Now, since the channel of the interferer is given by $(1, h_1, \cdots, h_M)$, it is sufficient to find the h_is. To do this, the receiver correlates all the equations above with $y_1(t)^*$ and taking the expectation.

$$E[y_1(t) y_1(t)^*] = P_i + \sum h_1^k h_1^{k*} P_k$$

$$E[y_j(t) y_1(t)^*] = h_j P_i + \sum h_j^k h_1^{k*} P_k,$$

where P's are the corresponding powers. Since the only unknowns in the above equations are P_i and h_j's, they can be easily computed. Thus, even in the presence of K concurrent transmissions, a 802.11 receiver can estimate the channel of the interferer without knowing the preamble.