WiSlow: A Wi-Fi Network Performance Troubleshooting Tool for End Users

Kyung-Hwa Kim*, Hyunwoo Nam† and Henning Schulzrinne*
*Department of Computer Science, Columbia University, New York, NY
†Department of Electrical Engineering, Columbia University, New York, NY

***Abstract*—Slow Internet connectivity is often caused by poor Wi-Fi performance. The main reasons of such performance degradation include channel contention and non-Wi-Fi interference. Although these problem sources can be easily removed in many cases once they are discovered, it is difficult for end users to identify the sources of such interference.**

We investigated the characteristics of different sources that can degrade Wi-Fi performance, and developed WiSlow, a software tool that diagnoses the root causes of poor Wi-Fi performance using user-level network probes, and leveraging peer collaboration to identify the physical location of these causes. WiSlow uses two main methods: packet loss analysis and 802.11 ACK number analysis. The accuracy of WiSlow exceeds 90% when the sources are close to Wi-Fi devices. Also, our experiment proves that the collaborative approach is feasible for determining the relative location of an interfering device.

I. Introduction

Today, it is common for households to put together home networks with a private wireless router (access point) that supports multiple wireless devices. However, the increasing usage of wireless networks inevitably results in more contention and interference, which causes unsatisfactory Wi-Fi performance. Furthermore, non-Wi-Fi devices such as microwave ovens, cordless phones, and baby monitors severely interfere with Wi-Fi networks because these devices operate on the same 2.4 GHz spectrum as 802.11b/g [1]. Although these problem sources can be easily removed in many cases (e.g., by relocating the interfering device, choosing a different channel, or moving to the 5 GHz band), it is difficult for technically non-savvy users to even notice the existence of channel contention or interference caused by non-Wi-Fi devices (non-Wi-Fi interference). Instead, properly working routers or service providers are frequently misidentified as the culprit while the actual root cause remains unidentified. However, isolating the root causes of poor Wi-Fi performance is nontrivial, even for a network expert, because they show very similar symptoms at the user level, and special devices are required in order to investigate the lower layers of the protocol stack.

We introduce WiSlow ("Why is my Wi-Fi slow?"), a software tool that diagnoses the root causes of poor Wi-Fi performance with user-level network probes and leverages peer collaboration to identify their physical locations. In other words, the goal of this tool is to report the problem source to users such as "It appears that a baby monitor located close to your router is interfering with your Wi-Fi network." We focus on building software that does not require any additional spectrum analysis hardware (unlike, e.g., WiSpy [2], AirSleuth [3], or AirMaestro [4]). In addition, WiSlow does not depend on a specific network adapter such as the Atheros chipsets, which were used to achieve similar goals in other studies [5], [6]. These features enable WiSlow to run on common end-user machines.

978-1-4799-3360-0/14/$31.00 © 2014 IEEE

First, we investigate behaviors of 802.11 networks such as retries, frame check sequence (FCS) errors, packet loss, and bit rate adaption, which can be observed on ordinary operating systems. Our experimental results show that the statistical patterns of the above variables vary depending on the problem sources. For example, with the interference that caused by non-Wi-Fi devices, we observed a greater number of retried packets, fewer FCS errors, and larger variations in the bit rates compared to channel contention. Correlating these variables, we can categorize the sources of performance problems into several distinct groups. In addition, the non-Wi-Fi devices such as baby monitors, cordless phones, and microwave ovens show different patterns when the number of UDP packets and 802.11 ACKs are plotted over time. Based on these observations, we developed two methods: packet loss analysis and 802.11 ACK pattern analysis. These methods successfully distinguish channel contention from non-Wi-Fi interference and infer the product type of the interfering device. We believe that this technology will be useful to end users since it can inform them of what needs to be done in order to improve the performance of their networks—whether to change the Wi-Fi channel or remove a device that is emitting the interference.

In non-Wi-Fi interference scenarios, another goal is to identify the physical location of the source of interference. Although it is difficult to pinpoint the exact physical location of the source without a spectrum analyzer or additional support of wireless access points (APs), we could infer the relative location of the problem source by collaborating with other end users connected to the same wireless network. WiSlow collects probing results from peers and determines whether others observe the interference. If all the machines observe the same interference, it is highly likely that the problematic source is close to the wireless AP. However, if only one of the peers observes the interference, the source is likely to be located close to that peer. Our experimental results clearly show that this approach is feasible.

In summary, WiSlow (i) distinguishes channel contention from non-Wi-Fi interference, (ii) infers the product type of the interfering device (e.g., a microwave oven, cordless phone, or baby monitor) by analyzing network packets, and finally (iii) points out the approximate location of the source of interference by exploiting user collaboration. We evaluated WiSlow with various interference sources and it showed quite high diagnostic accuracy. It also proved that our approach locating the interference source is feasible.

The remainder of this paper is structured as follows. In Section II, we describe the common sources of Wi-Fi performance degradation. In Section III, we discuss the restrictions of an end user's environment and how WiSlow attempts to overcome them. Section IV explains the detailed methods of WiSlow and Section VI evaluates our approach.

II. Background

Common sources that cause Wi-Fi performance degradation include:

- **Wi-Fi channel contention:** degradation due to a channel crowded by multiple Wi-Fi devices that compete to transmit data through an AP. It also includes interference due to nearby APs that are using the same channel or adjacent channels.
- **Non-Wi-Fi interference:** interference due to non-Wi-Fi devices that use the same 2.4 GHz spectrum as the 802.11b/g networks. The devices include microwave ovens, cordless phones, baby monitors, and Bluetooth devices.
- **Weak signal:** when the signal is not strong enough due to distance or obstacles, packets can be lost or corrupted.

Although the extent varies, all the above sources result in severe performance degradation—some of them even drop the TCP/UDP throughput to almost zero [5]. In this study, we focus on Wi-Fi channel contention and common non-Wi-Fi interference sources.

III. Challenges

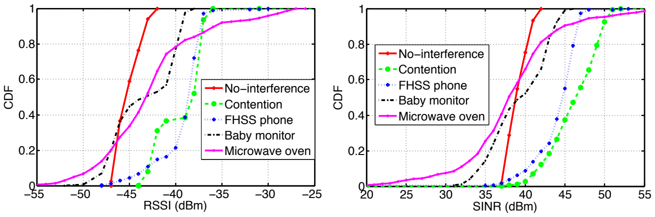
In this section, we describe the reasons why analyzing wireless networks is difficult for end users.

A. *Inaccurate RSSI and SINR measurements*

Received signal strength indication (RSSI) and Signal-to-interference-plus-noise ratio (SINR) are generally considered to be the key factors that indicate the quality of a wireless link. However, according to Vlavianos et al. [7], RSSI inaccurately captures the link quality and it is difficult to accurately compute SINR with commodity wireless cards. We also observed a similar result when monitoring RSSI and SINR values during our experiments. We placed various types of interference sources close to the AP and measured the values on a general client machine[1]. In Figure 1a, RSSI values with a baby monitor were higher than those obtained from a no-interference environment, which should be reversed when the measured UDP throughput is considered. In Figure 1b, the SINR values with a cordless phone were also higher than those obtained from a no-interference case. Furthermore, these results varied for each experiment. Based on this observation, we conclude that RSSI and SINR values captured by a general end-user's wireless card do not correctly represent the level of interference.

B. *No specific network adapter or driver*

We do not make any assumptions about the specific network adapters or drivers that end users may have. Some Atheros chipsets, which are widely used in research studies, support a *spectral scan* that provides a spectrum analysis of multiple frequency ranges. Rayanchu et al. developed Airshark [5] and WiFiNet [6] leveraging this feature to distinguish non-Wi-Fi interferers using a commodity network card without specialized hardware. Although this approach achieved quite high accuracy in identifying the interfering devices, to the best of our knowledge, only a few chipsets (e.g., Atheros) currently provide this feature. In addition, we failed to discover references to this feature for any operating system (OS) other than Linux. Since there are hundreds of products that use a different chipset and/or OS, it is impractical to assume that a general end user has this specific setup. Therefore, we focus instead on analyzing the quality of a link observing user-accessible packets such as UDP and 802.11 packets. Because the mechanisms of these protocols are not significantly different for many Wi-Fi devices, we believe that WiSlow's user-level approach can help a wider range of end users.



(a) RSSI measurement

(b) SINR measurement

Fig. 1: The CDFs of RSSI and SINR values

C. *Lack of monitoring data*

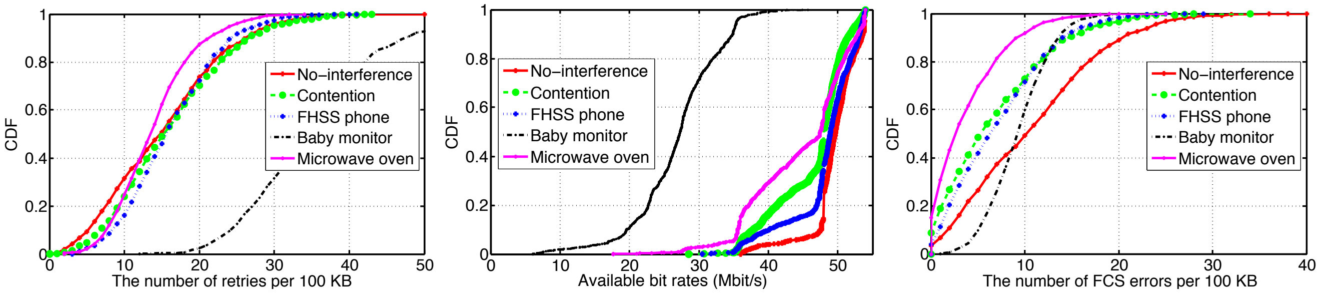
Another restriction in the end-user environment is the lack of a monitoring history. If we assume that we have been monitoring the machine up to the moment when a performance problem happens, the diagnosis will be easier because we can obtain several important clues such as the average quality of the link, the time when the problem started, and whether it has happened in the recent past. However, although the overhead of network monitoring is not heavy on modern machines, it is difficult to expect that end users will continuously run such a tool. The more common scenario is that a user launches a troubleshooting tool like WiSlow and requests a diagnostic only after he/she has noticed a severe performance problem. Therefore, we need to design the tool assuming little or no previous monitoring data. In the next section, we explain how WiSlow estimates the problem source without knowing the baseline quality of the network.

IV. WiSlow

In this section, we elaborate on the details of probing methods. First, to investigate the behavior of Wi-Fi networks in each problem scenario, we artificially inject problems while transmitting UDP packets between a client (laptop) and an AP. We capture every packet on the client machine, and then trace the transport layer (UDP), the 802.11 MAC layer, and some user-accessible 802.11 PHY layer information to ascertain each problematic scenario's interference levels and characteristics.

To capture 802.11 packets, WiSlow uses monitor mode of wireless adapters. It provides the Radiotap [8] header, which is a standard for 802.11 frame information. The headers are used to extract the lower layer information such as FCS errors and bit rates. Sniffing the wireless packets is supported by most Linux and all Mac OS X machines without additional drivers or kernel modification. Therefore, if we can successfully characterize each performance-degrading source by probing the transmitted packets, the same probes will enable WiSlow to identify the problem sources on most platforms. However,

[1] We used a MacBook Pro 2013 (network card: AirPort Extreme, chipset: Broadcom BCM43 series) in this measurement



(a) The CDF of number of 802.11 retries

(b) The CDF of available bit rates

(c) The CDF of FCS errors

Fig. 2: 802.11 statistics with various interference sources

it is not always possible to capture wireless packets on some types of OS, e.g., Microsoft Windows [9]. Instead, Windows provides several APIs that report 802.11 packet statistics to user applications. Those APIs enable WiSlow to run on Windows because they provide all the information that WiSlow must extract from the 802.11 packets.

In the following sections, we explain WiSlow's two main diagnostic methods: packet loss analysis and 802.11 ACK pattern analysis.

A. *Method 1: packet loss analysis*

First, we found that each problem source varies in their packet loss characteristics, represented by three statistics: 1) the number of 802.11 retries, 2) the available bit rates, and 3) the number of FCS errors. In each experiment, we measured these values on a client laptop while downloading UDP packets from an AP. The values were recorded for each 100 KB of UDP packets received. We repeated this experiment for different scenarios including channel contention and non-Wi-Fi interference. To simulate channel contention, we set up several laptops sending bulk UDP packets to the AP. To generate non-Wi-Fi interference, we placed each interfering device (baby monitors, microwave ovens, and cordless phones) close to the AP (about 20 cm) and measured the effect on the client placed at various distances from the AP. (In this study, we did not consider the combined interference of multiple devices.) Note that the client downloaded 100 MB of UDP packets for each experiment to collect a statistically meaningful amount of samples, but when actually probing on an end user's machine, WiSlow only needs to transmit 5 MBytes of UDP packets to identify the root cause, which takes a reasonable amount of time (10–30 s).

- **Retry and available bit rate:** Since an 802.11 retry and bit rate reduction are both initiated by a packet loss, their temporal changes are closely correlated; when a packet loss occurs, the bit rate decreases by the 802.11 rate adaptation algorithm [10]. The probability of packet loss then decreases due to the reduced bit rate, which lowers the number of retries. After that, the bit rate gradually increases again owing to the reduced packet loss, which leads to a higher probability of packet loss and retries. In other words, if contention or interference exists, it causes packet losses, and then the bit rate and the number of retried packets repeatedly fluctuate during the subsequent data transmission. Because of this fluctuation, the measured statistics of retries and bit rates do not represent the characteristics of interference sources correctly. Figure 2a and 2b shows that the cumulative distribution functions (CDFs) of the values do not distinguish each device except the baby monitor.
- **Frame check sequence errors:** Another variable that we trace is the number of FCS errors per byte. In our experiments, we counted the number of FCS errors per 100 KB of data. Intuitively, it can be predicted that non-Wi-Fi interference introduces more FCS errors than channel contention or a no-interference environment. This is because the packet corruptions are likely to occur more frequently when a medium is noisy. However, in our experiment, it turned out that a large number of FCS errors are not necessarily correlated with severe interference. On the contrary, we often observed that fewer FCS errors occur in a severe interference environment (e.g., interference caused by a baby monitor) than in a no-interference environment (Figure 2c). This paradox can be explained by the low bit rates in the interference case, which implies that a smaller number of bits are transmitted in the same bandwidth. Consequently, the number of FCS errors per byte alone is not sufficient to characterize interference sources.

*1) **Packet loss estimation**:* As we stated above, although the number of retries, bit rate, and FCS errors are affected by the current state of the wireless network, they often show very different statistics for each experiment set. We conjecture several reasons; the environment is not exactly the same in every experiment, the occurrence of packet loss is probabilistic rather than deterministic, and the individual variables fluctuate over time, affecting each other and leading to different statistics for a certain period of time. Therefore, it would be more reasonable to compare the combinations of these statistics together instead of investigating each variable individually.

There are two cases that can cause a retry. First, a packet was not delivered, i.e., it was lost. Second, a packet was delivered but it had an FCS error. We can estimate the number of packets lost (the first case) by subtracting the number of FCS errors from the number of retries (Eq. 1).

$$N_{PacketLoss} = N_{Retries} - N_{FCSerrors} \tag{1}$$

We found that this estimated number of packet losses represents the level of interference more reliably than the individual statistics of retries, bit rates and FCS errors. In other words, the number of packet losses provides relatively consistent results in repeated experiments, while the others varied for each experiment. Figure 3 shows that the CDF of the estimated number of packet loss clearly distinguishes each device compared to the CDFs in Figure 2. It can be seen that a baby monitor causes the most severe amount of packet loss while cordless phones cause a relatively small amount of packet loss. Since baby monitors send video and audio data at the same time, they use more bandwidth than cordless

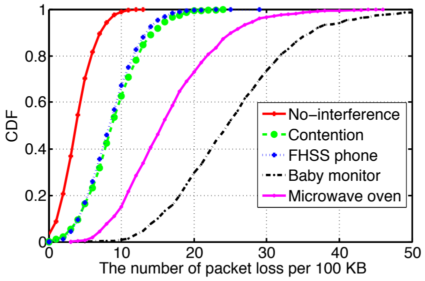
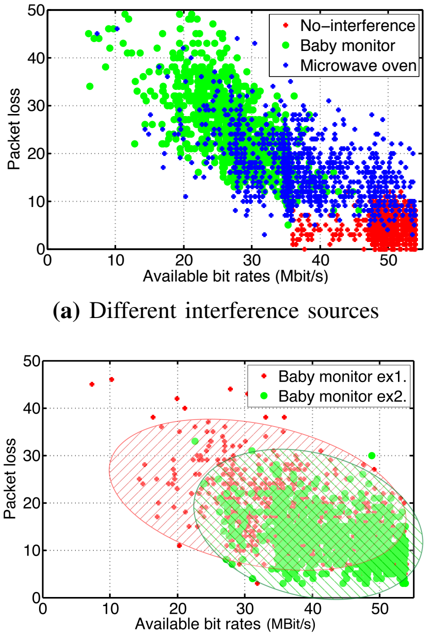


Fig. 3: The CDFs of the estimated packet loss



(a) Different interference sources

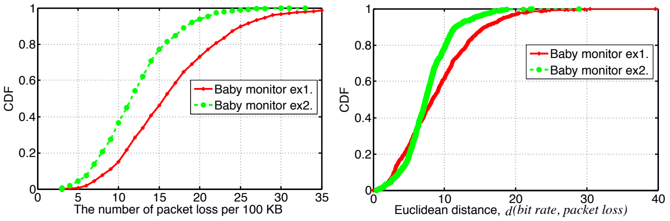
(b) The same device, a baby monitor, in different environments

Fig. 4: The distribution of the correlation of bit rates and the estimated packet loss

phones that send audio only, thus causing more interference. Channel contention shows less packet loss because of the 802.11 collision-avoidance functions such as random back-off and RTS/CTS that force each client to occupy the medium in separate time slots. In this case, the degradation of throughput is caused by the shared medium rather than noise from other sources.

Furthermore, we found that the correlation between bit rate and the estimated number of packet losses shows clearer differences among various problem sources. In Figure 4a, the majority of the samples from a clean environment are distributed in a healthy zone (higher bit rate and lower packet loss) while the samples of baby monitors and microwave ovens are widely dispersed. WiSlow uses the correlation of these two variables to distinguish the level of interference.

As described above, the problem sources each have their own distribution patterns on the scatter plot. However, an end user cannot infer a root cause by simply matching the measured statistics with the results of our experiments. This is because the measurement of a wireless network is highly affected by the client's own environment such as a distance from the AP, signal power, or fading (multi-path and shadowing). In other words, even though they have the same type of problem, the statistics of the measured metrics can vary depending on each end user's own situation. Note that this is the reason why simple measurements such as the higher-layer throughput (e.g., TCP or UDP) or number of 802.11 retries are not enough to identify the level of interference and the type of interferers. We found that *even if the underlying environment changes, the extent of the area over which a set of samples (correlated packet loss and bit rate) are dispersed remains similar if the problem source is the same.* Figure 4b shows that even though the two groups of samples from discrete environments are distributed on different spots on the coordinate plane, their extent is similar. Thus, we first quantify how widely the samples are dispersed by calculating the Euclidean distances between each sample and the mean ($\sqrt{(M_x - S_x)^2 + (M_y - S_y)^2}, mean = (M_x, M_y), sample = (S_x, S_y)$). Figure 5 compares the CDFs obtained from two experiments that were conducted with the same baby monitor in two discrete environments. The CDFs of packet loss estimation (Figure 5a) show different distributions while the CDFs of the Euclidean distances between the samples and the mean show similar distribution (Figure 5b).



(a) The estimated number of packet loss

(b) The Euclidean distance between each sample and the mean

Fig. 5: The CDFs obtained from two experiments with the same baby monitor in different environment.

Therefore, WiSlow can use the CDFs of the Euclidean distances to identify the root causes of network interference. We prepare these CDFs of each problem source in advance, which are obtained from our experiments. Then, WiSlow traces the wireless packets on an end user's machine, generates a CDF of the distances, and compares it to the pre-obtained CDFs of each problem source. For the convenience of identification, we group the problem sources into three groups by the shape of the CDFs: no interferers (group 1), light interferers (group 2), and heavy interferers (group 3). Each group has its representative CDFs that are determined by multiple experiments (Figure 6). In our data sets, group 1 indicates a no-interference environment, group 2 includes channel contention and cordless phones that use frequency-hopping spread spectrum (FHSS), and group 3 contains microwave ovens and baby monitors. WiSlow examines which representative CDF is the most similar one to the CDF measured on the user's machine. To compare the CDFs, WiSlow uses the two-sample Kolmogorov-Smirnov test (K-S test), a widely used statistical method that tests whether two empirical CDFs obtained from separate experiments have the same distribution [11]. If the p-value of this test is close to 1, the two CDFs are likely to come from the same distribution, however, if the p-value is close to 0, they are likely to come from different distributions. Since the K-S test not only considers the average and variance of the samples but also takes into account the shape of the CDFs, it best fits the purpose of WiSlow where it is used to pick the most similar distribution from multiple data sets. Our evaluation proves that

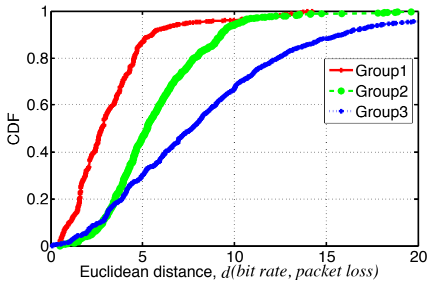
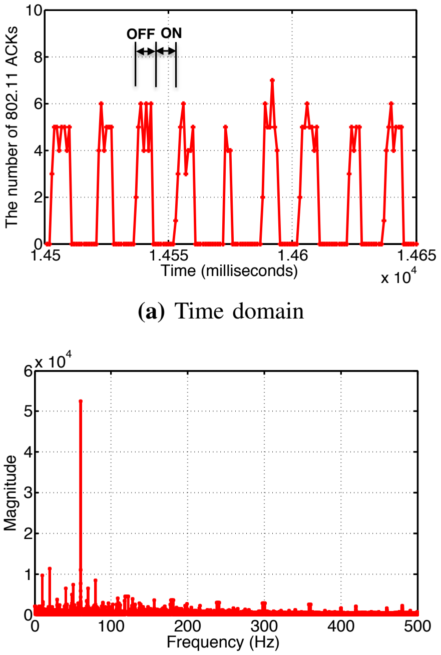


Fig. 6: Three groups categorized by the packet loss analysis: 1) a no-interference environment, 2) contention and FHSS cordless phones, and 3) microwave ovens and baby monitors



(a) Time domain

(b) Frequency domain

Fig. 7: The number of 802.11 ACKs with interference of a microwave oven

the approach explained above successfully distinguishes these groups, minimizing the impact of the end user's underlying environment.

B. Method 2: 802.11 ACK pattern analysis

The first method is able to determine which type of loss pattern a problem source has. However, because multiple problem sources are categorized into each group, we need another method that further narrows down the root causes. In this section, we explain the second method, designed to distinguish several detailed characteristics of non-Wi-Fi devices such as frequency hopping and duty cycle.

WiSlow sends bulk UDP packets to the AP and counts the received 802.11 ACKs to check the quality of a wireless link within a given period. In order to detect patterns on the scale of milliseconds, we use a very small size of UDP packets (12 bytes) that reduces potential delays such as propagation and processing delays, and we transmit as many UDP packets as possible to reduce the intervals between samples. As a result, we received 0–7 ACKs per millisecond.

In the following sections, we describe the results of the above method when performed with various non-Wi-Fi interferers, and we explain how WiSlow identifies the devices based on the results.

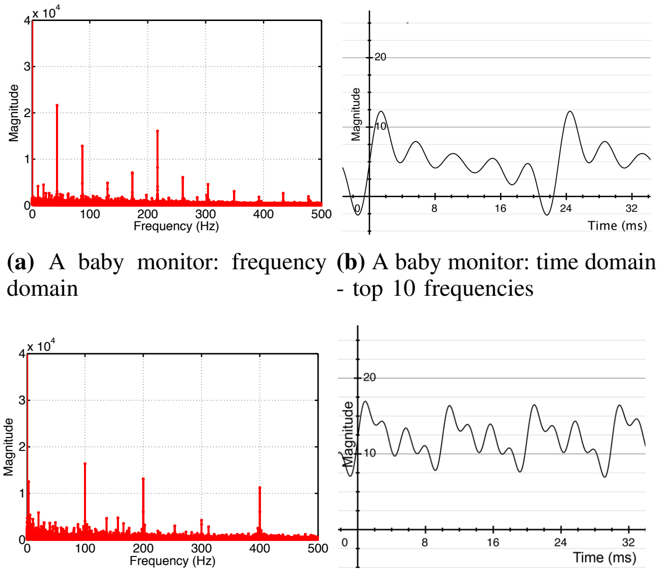
*1) **Duty cycle (microwave ovens):*** Microwave ovens generate severe interference in almost every channel of the 2.4 GHz band. We identify this heavy interferer using its duty cycle, which is the ratio of the active duration to the pulse period. It is known that the duty cycle of microwave ovens is 50% and the dwell time is 16.6 ms (60 Hz)[2] [12]. This implies that it stays in the ON mode (producing microwaves) for the first 8.3 ms and the OFF mode for the next 8.3 ms. This feature can be observed by various means such as using a spectrum analyzer [2] or by signal measurement [5].

Our hypothesis was that a user-level probe could also detect this *on-off* pattern if the network packets were monitored on a millisecond timescale because the packets would be lost only when the interferer was active (*on* mode). To validate this assumption, we implemented the above method and plotted the number of successfully received 802.11 ACKs per millisecond. As a result, a clearly perceptible waveform with a 50% duty cycle is observed (Figure 7a); the number of ACKs is over five for the first 8 ms and zero during the next 8 ms. This pattern repeats while the microwave oven is running. This result becomes clearer when it is converted to the frequency domain (Figure 7b) using a fast Fourier transform (FFT). The highest peak is at 60 Hz, which means the cycle is 16.6 ms. This number is exactly the same as the known duty cycle of microwave ovens.

Consequently, if a perceptible cycle is detected from this probing method and the period matches a well-known value, WiSlow determines that the current interference is due to a particular type of device (e.g., 60 Hz for microwave ovens).

*2) **Frequency hopping (baby monitors and cordless phones):*** The duty cycle of typical audio and video transmitters such as baby monitors is known to be 100%. It means that they send and receive data constantly, implying that they continuously interfere with Wi-Fi networks without any *off* period. Therefore, intuitively, we do not expect to observe similar ACK patterns as those observed in the microwave oven experiment. However, when converting the plot from the time domain to the frequency domain, we observe another notable pattern. Figure 8a shows that there are multiple high peaks set apart by a specific interval, i.e., 43 Hz (occurring at 43, 86, 129, and 172 Hz). This is in contrast to the microwave ovens that showed only one significant peak at 60 Hz (Figure 7b). We conjecture that these peaks are caused by frequency hopping; a frequency hopper switches its frequency periodically, and interference occurs when it hops to a nearby frequency of the current Wi-Fi channel. Since the frequency-hopping chooses the next frequency using a pseudorandom sequence, it creates diverse pulses with different magnitudes, that are randomly positioned in the ACK number plot. For clarity, we plot a quantized time-domain graph (Figure 8b) that is converted back from the frequency-domain graph. We used the 10 highest frequencies from Figure 8a. In the time-domain graph, the number of ACKs (y-axis) fluctuates periodically, however, note that the heights of the peaks vary. The possible explanation is as follows: the number of ACKs is large when the device hops

[2] This frequency could be 50 Hz in other countries (e.g., Europe and most of Asia) where 50 Hz AC power is used.



(a) A baby monitor: frequency domain **(b)** A baby monitor: time domain - top 10 frequencies

(c) An FHSS cordless phone: frequency domain **(d)** An FHSS cordless phone: time domain - top 10 frequencies

Fig. 8: The number of 802.11 ACKs per 100 KB of UDP packets with a baby monitor and a cordless phone

far from the current Wi-Fi channel, and it is relatively small when it hops to a nearby frequency. If the device hops into the exact range of the Wi-Fi channel, the number of 802.11 ACKs drops almost to zero. In other words, there are multiple levels of interference, which depend on how closely in frequency the device hops to the frequency used by the Wi-Fi channel. These multiple levels of interference create several pulses that have different magnitudes and frequencies. Finally, because the hopping interval of the device is fixed, the frequencies of the created pulses are synchronized such that the periods of the cycles are multiples of a specific value.

The FHSS cordless phone, which also uses the frequency hopping technique, showed a similar result – multiple peaks with a fixed interval, 100 Hz (Figure 8). This verifies that our method is suitable to identify frequency hopping devices.

Consequently, we can distinguish frequency-hopping devices by determining whether the number of 802.11 ACKs has multiple high peaks with a certain interval in the frequency domain. We check this by linear regression of the peak frequencies; if the correlation coefficient is greater than 0.99, we consider it to be a frequency-hopping device.

*3) **Fixed frequency (analog cordless phones):*** Since typical analog cordless phones use a fixed frequency, they usually interfere only with a small number of channels. (The analog phones we tested only interfered with Channel 1.) Because they do not change frequency, severe interference occurs if the current Wi-Fi channel overlaps with the frequency of the phone. In addition, their duty cycle is close to 100%, which implies that no ACK cycle exists. In our experiments, the UDP throughput stayed very low and no explicit ACK cycle (no hopping) was observed, as expected. Therefore, WiSlow concludes that an analog cordless phone is the interferer if there is a heavy interference pattern but no explicit ACK cycle or duty cycle is detected. Then, we can inform the user that switching the Wi-Fi channel can improve the performance in this case because this kind of device is likely to affect only a few channels.

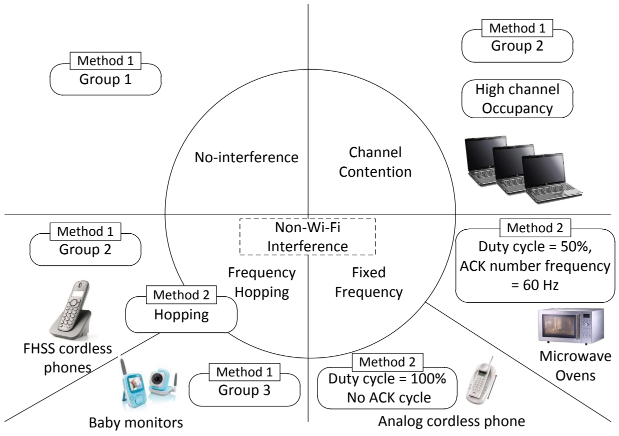


Fig. 9: The classification of problem sources by WiSlow's methods

*4) **Bluetooth:*** Bluetooth is another widely used wireless standard that operates in the 2.4 GHz spectrum. Hopping within the entire 2.4 GHz band, it interferes with every channel of an 802.11 network. However, algorithms such as Adapted Frequency Hopping (AFH), which is used to automatically avoid busy channels, mitigate this interference. Consequently, Bluetooth inconsiderably affects the performance of 802.11 networks. In a measurement by Rayanchu et al. [5], Bluetooth was shown to degrade the UDP throughput by about 10% as a worst case. Since we also verified that Bluetooth does not interfere much with 802.11g networks based on our experimental result, we excluded Bluetooth in our identification algorithm.

C. *Classification*

WiSlow takes into account the combination of the results from the first method (packet loss analysis) and the second method (ACK pattern analysis) to identify the device type precisely. For example, the result of the first method is Group 3 and that of the second method is frequency-hopping, we consider the problem source to be a baby monitor. In addition, WiSlow looks into the source and destination addresses of the captured 802.11 packets in order to examine the channel occupancy rate. If the channel is highly occupied by other clients or nearby APs, but WiSlow does not detect any non-Wi-Fi interference, it considers the root cause to be channel contention. Figure 9 describes the classification algorithm that WiSlow uses to conclude the root cause. Currently, we aim to provide the best estimate of suspicious devices that we are aware of, but we believe that more types of devices can be covered easily once they are characterized in a similar manner.

V. Locating Interfering Devices

In this section, we describe a method to determine the physical locations of interfering devices. A number of research studies on indoor location tracking have attempted to pinpoint the location of laptops or smartphones through various methods [13]–[15]. While these studies focus on locating *client devices* using signal information such as RSSI and SINR values, we focus on locating *interference sources* using multiple collaborative end-user devices. Compared to locating Wi-Fi devices, there are several difficulties in locating non-Wi-Fi devices for end users. First, it is impossible to obtain measurement data such as RSSI and throughput from such devices (e.g., microwave ovens neither monitor signals, nor communicate with Wi-Fi devices). Second, owing to the limited capability of the hardware, end-user devices cannot detect signals emitted from the devices precisely. To overcome these circumstances, we leverage multiple Wi-Fi devices; a probing client (end-user machine) requests cooperative clients to perform a WiSlow diagnostics as described in previous sections. It then receives the diagnostic result containing the type of detected device and its *interference strength* from each client. We calculate the *interference strength* using the magnitude of the particular ACK number frequency that was used to detect the device, as described in the previous section (Method 2). For example, the *interference strength* of a microwave oven can be determined based on a magnitude of 60 Hz in the FFT of the ACK number analysis. In the case of an FHSS device, it can be determined by the sum of the magnitudes of the multiple frequencies caused by the frequency hopping pattern. After collecting the strength values from the clients, we use the same method of obtaining the center of mass to find the location of the interference. If the *interference strength* detected by a particular client is greater than the *interference strength* detected by other clients, it means that the interference source is closer to that client. Therefore, *interference strength* can be considered equivalent to the mass in the formula of the center of mass. WiSlow first obtains the coordinates of cooperative clients based on the input from end users and calculates the coordinates of the interference source using the following formula.

$$M_i = \sum_{k=1}^{m} f_i(kx), \qquad \boldsymbol{R} = \frac{1}{\sum\limits_{i=1}^{n} M_i} \sum_{i=1}^{n} M_i \boldsymbol{r_i} \tag{2}$$

M_i is the strength of interference on the ith client and f_i denotes the function of the measured magnitudes for each frequency, kx, where x is the smallest frequency caused by the interfering device. The coordinate of the interference source, $\boldsymbol{R}$, can be calculated based on the sum of each client's weighted (M_i) coordinates ($\boldsymbol{r_i}$).

VI. Evaluation

In this section, we describe the accuracy of WiSlow in identifying the root cause of a Wi-Fi performance problem. First, we placed a laptop 8 m away from an AP, where Wi-Fi performance is not affected by weak signal strength. Then, we located the interfering devices between them, one at a time. We repeated the experiments altering the distance between the interfering device and the AP. We ran WiSlow on the laptop 15 times each at six different locations (a total of 90 measurements for each interfering device) and counted the number of times that WiSlow correctly diagnosed the root cause. First, without considering the type of the non-Wi-Fi device, we tested the capability of WiSlow to distinguish between no-interference, channel contention, and non-Wi-Fi interference.

We evaluate the *diagnostic accuracy* and the *false positive* rate (type-I error) of WiSlow for each problem source. The diagnostic accuracy of a problem source $\mathtt{P}$ is the ratio of the number of correct diagnostics to the total number of experiments in which $\mathtt{P}$ is injected as a problem source. The false positive rate of $\mathtt{P}$ is the ratio of the number of cases that the cause is misidentified as $\mathtt{P}$ to the total number of experiments in which $\mathtt{P}$ is not actually the cause.

Injected Problem	**Distance from the AP**	**Accuracy**	**False Positive**
No interference	-	100.0%	14.1%
Channel contention	-	92.2%	1.5%
Non-Wi-Fi interference (baby monitor, cordless phone, and microwave oven)	0.0 m	100.0%	3.9%
	0.5 m	97.8%	
	1.0 m	82.2%	
	1.5 m	82.2%	
	2.0 m	73.3%	
	2.5 m	68.9%	

TABLE I: The accuracy of WiSlow for distinguishing between a clean environment, channel contention, and non-Wi-Fi interference

Table I shows that WiSlow successfully distinguishes them with high accuracy (over 90% for no-interference and channel contention). In the non-Wi-Fi interference case, the accuracy was also over 90% when the interfering device was close to the AP; however, it notably decreased when the distance between the AP and the device increased. We found that this inaccuracy was mostly caused by the FHSS cordless phones. In the following sections, we explain the reason for this inaccuracy and the method WiSlow employed to reduce it.

A. Identifying the root cause

Table II shows the detailed diagnostic results of identifying each type of non-Wi-Fi device. First, WiSlow could clearly detect interference caused by a microwave oven regardless of the distance (average 98%). In our extra experiments, WiSlow could detect the duty cycle of the microwave oven even when located relatively far from the AP and laptop (11 m and 16 m). However, in these cases, since the microwave oven did not severely interfere with the Wi-Fi network, we do not elaborate further on the results in the present paper.

Second, the diagnostic accuracy of detecting baby monitors was also very high when it was close to the AP. However, it dropped to under 6.7% when the distance was greater than 1 m (Table II). In most cases, it was misidentified as a FHSS cordless phone, which contributed the high false positive rate of this device (24.8%). This result occurred because these two devices have the same characteristic (frequency hopping), and WiSlow considers their level of interference to distinguish them. In other words, if a baby monitor is far from a Wi-Fi device and causes less interference, it can mislead WiSlow's identification. The accuracy of detecting FHSS cordless phones was also low when it was not close to the AP (6.7% at 2.5 m). However, this was because the cordless phone caused insignificant interference at this spot; the average UDP throughput was 13.28 Mb/s at 2.5 m (the average throughput with no interference was 14 Mb/s in the same environment). With this small interference, WiSlow did not observe the expected hopping patterns. As a result, the majority of incorrect diagnostic results were *no interference*, which explains its high false positive rate (14.1%) shown in Table I.

The low accuracy of detecting baby monitors and FHSS cordless phones can be improved if we take into account their specific ACK number frequency values, which were discussed in Section IV-B. Recall that the ACK number frequencies of the baby monitor were a multiple of 43 Hz, and those of the FHSS cordless phone were a multiple of 100 Hz. When WiSlow is adapted to consider these specific values, the detection accuracy increases dramatically. Table III shows that the accuracy was 100% most of the time, except when

Non-Wi-Fi Interference	Distance from the AP	Avg. Throughput	Diagnostic Accuracy	False Positive
Microwave oven	0.0 m	7.54 Mb/s	100 %	0.4 %
	0.5 m	8.52 Mb/s	100 %	
	1.0 m	8.96 Mb/s	100 %	
	1.5 m	9.33 Mb/s	100 %	
	2.0 m	9.30 Mb/s	100 %	
	2.5 m	8.91 Mb/s	93.3 %	
Baby monitor	0.0 m	0.51 Mb/s	100 %	1.1 %
	0.5 m	3.16 Mb/s	73.3 %	
	1.0 m	4.79 Mb/s	6.7 %	
	1.5 m	4.49 Mb/s	6.7 %	
	2.0 m	4.81 Mb/s	6.7 %	
	2.5 m	5.17 Mb/s	0.0 %	
FHSS Cordless phone	0.0 m	6.76 Mb/s	80.0 %	24.8 %
	0.5 m	9.65 Mb/s	86.7 %	
	1.0 m	10.02 Mb/s	40.0 %	
	1.5 m	10.05 Mb/s	40.0 %	
	2.0 m	12.44 Mb/s	13.3 %	
	2.5 m	13.28 Mb/s	6.7 %	

TABLE II: The accuracy of WiSlow for identifying non-Wi-Fi devices

Non-Wi-Fi Interference	Distance from the AP	Diagnostic Accuracy
Baby monitor	0.0 m	100%
	0.5 m	100%
	1.0 m	100%
	1.5 m	100%
	2.0 m	100%
	2.5 m	100%
Cordless phone	0.0 m	100%
	0.5 m	100%
	1.0 m	100%
	1.5 m	66.7%
	2.0 m	26.7%
	2.5 m	6.7%

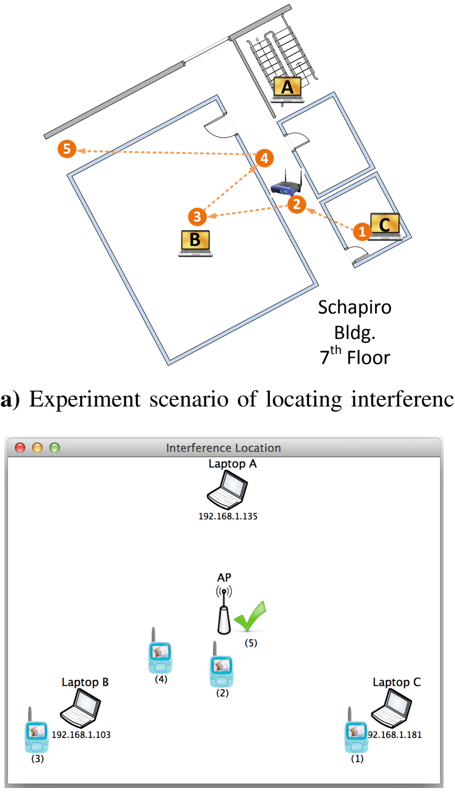
TABLE III: The accuracy of WiSlow for identifying baby monitors and cordless phones

the FHSS cordless phone was placed at locations farther than 1.5 m[3]. However, the disadvantage of this approach is that WiSlow needs to learn the ACK number frequency value of the particular product in advance because the pattern depends on each model. It appears to be impractical to collect the patterns from every product. However, we found that different models of the same type of product likely have common characteristics. For example, we tested four FHSS cordless phones produced by two different manufacturers[4], and each one showed the same ACK number frequencies (multiples of 100 Hz). Therefore, we believe that collecting a small amount of information can cover the majority of devices if they follow the industry standards or use similar technologies.

In conclusion, WiSlow successfully detected the root cause of Wi-Fi performance degradation with a high probability (over 90%) in most cases, although it frequently misidentified the type of certain non-Wi-Fi interfering devices when they were not located near the Wi-Fi device. However, this inaccuracy can be removed if we take into account the pre-obtained ACK number pattern of each device.

B. *Locating interfering devices*

We set up three laptops and one 802.11g AP in a building at Columbia University. We placed a baby monitor between them and changed its location over time. Figure 10a illustrates our experimental scenario. The circled numbers indicate the movement path of the baby monitor. We ran WiSlow each time the location was changed. Figure 10b shows an actual real-time screenshot of WiSlow detecting the location of the baby monitor. For the first location, laptops A and B reported no interference, but laptop C detected the baby monitor successfully. For the second location, the three laptops all detected the baby monitor and reported similar *interference strengths* because the interference source was close to the AP, and thus the entire wireless network was affected by the baby monitor. In this particular case, WiSlow could infer that the problem source was likely to be a device placed near the AP. For the third location, only laptop B detected the baby monitor, and thus WiSlow placed the baby monitor icon close to laptop B. For the fourth location, the three laptops all detected the baby monitor, but the measured *interference strengths* were distinct. Therefore, based on Equation 2, WiSlow pointed the location of the baby monitor as being relatively close to laptop B. For the last spot, since none of the laptops detected any interference, only a green check icon was displayed, which indicates that the state of the network is good.



(a) Experiment scenario of locating interference

(b) A real-time result of WiSlow

Fig. 10: Locating the interference source

This experiment proves that our approach is feasible for finding the relative location of an interfering device. Although WiSlow shows errors of several meters in pinpointing a location, we believe that this level of error is not critical for a home network environment.

VII. Related Work

Airshark [5] uses a commodity Wi-Fi network adapter to identify the source of interference. It leverages a spectral scan to obtain signal information from multiple frequency ranges. It identifies the interference sources very accurately (over 95%) by analyzing the spectrum data using various methods. However, we suppose that it would be difficult to

[3] These inaccuracies can be ignored because the throughput shows there was actually *no interference* even though the cordless phone was active

[4] Motorola and Panasonic

apply this approach for typical end users because collecting high-resolution signal samples across the spectrum is impossible if the network card does not support this functionality. WiFiNet [6] identifies the impact of non-Wi-Fi interference and finds its location using observations from multiple APs that are running Airshark. However, this approach seems difficult to be used in a common home network environment that has a single AP. In contrast, WiSlow focuses on identifying the location of the interference source by cooperating end users.

Kanuparthy et al. [16] propose an approach similar to WiSlow in terms of using user-level information. They distinguish congestion (channel contention) from hidden terminals and low SNR by measuring the one-way delay of different packet sizes. They then investigate the delay patterns to distinguish hidden terminals from low SNR. While their approach intentionally avoids using layer-2 information, WiSlow actively exploits 802.11 information in order to obtain a more detailed identification (e.g., device type causing the interference). Spectrum MRI [17] also isolates interference problems. The authors discuss that the link occupancy and retransmission rate is different depending on the sources of interference. They measure and compare those metrics to identify Bluetooth, channel congestion and the "slow link on same AP" problem.

Sundaresan at el. [18] present a tool that identifies whether a performance bottleneck exists inside the home network or on the access link by measuring variation of packet interarrival time. It also evaluates the state of the wireless link by monitoring the bitrate and throughput on an AP. While this tool focuses on identifying where a bottleneck exists, WiSlow focuses on identifying the type of interference source within the wireless network.

VIII. Discussion and Future Work

A. 802.11n

802.11n uses both 2.4 and 5 GHz bands. Although fewer non-Wi-Fi devices are operating at 5 GHz, and thus less interference presently exists at that band, Cisco has anticipated that more devices will use the 5 GHz band in the future, and therefore a similar interference will likely occur [19]. We believe that our basic approach will also be feasible for discovering non-Wi-Fi interference sources at 5 GHz if customized to an 802.11n environment.

B. Ad-Hoc mode and mobile devices

We also tested WiSlow on an ad-hoc network using two laptops, which enables WiSlow to run independently without communicating with an AP. Since ad-hoc networks also use the same 802.11 protocol, we did not see any differences from the experiments with an AP. We expect that using WiSlow with ad-hoc networks will be especially helpful in independently discovering nearby interference sources when used with multiple mobile devices such as smartphones.

IX. Conclusion

We designed WiSlow, a Wi-Fi performance trouble shooting application, specialized to detect non-Wi-Fi interference. WiSlow distinguishes 802.11 channel contention from non-Wi-Fi interference, and identifies the type of interfering devices present. WiSlow was designed to exploit user-level probing only, which enables a software-only approach. For this purpose, we developed two novel methods that use user-accessible packet information such as UDP and 802.11 ACKs.

The accuracy of WiSlow exceeds 90% when the sources are close to a Wi-Fi device. WiSlow becomes less accurate when the devices are located farther. However, this inaccuracy can be removed if we take into account the known characteristics of each device. Also, we proved that the collaborative approach is feasible for determining the relative location of an interfering device.

X. Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1218977.

References

[1] S. Gollakota, F. Adib, D. Katabi, and S. Seshan, "Clearing the RF smog: making 802.11n robust to cross-technology interference," in *Proc. of ACM SIGCOMM*, Toronto, Ontario, Canada, Aug. 2011.

[2] "Wi-Spy," http://www.metageek.net/, [Online; accessed May 2013].

[3] "AirSleuth," http://nutsaboutnets.com/airsleuth-spectrum-analyzer/, [Online; accessed May 2013].

[4] "AirMaestro," http://www.bandspeed.com/products/products.php, [Online; accessed May 2013].

[5] S. Rayanchu, A. Patro, and S. Banerjee, "Airshark: Detecting non-WiFi RF Devices Using Commodity WiFi Hardware," in *Proc. of ACM IMC*, Berlin, Germany, Nov. 2011.

[6] S. Rayanchu, A. Patro, and S. Banerjee, "Catching Whales and Minnows Using WiFiNet: Deconstructing non-WiFi Interference Using WiFi Hardware," in *Proc. of USENIX NSDI*, San Jose, CA, USA, Apr. 2012.

[7] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, and M. Faloutsos, "Assessing link quality in IEEE 802.11 wireless networks: Which is the right metric?" in *Proc. of PIMRC*, Cannes, France, Sep. 2008.

[8] "Radiotap," http://www.radiotap.org/, [Online; accessed May 2013].

[9] "WLAN packet capture," http://wiki.wireshark.org/CaptureSetup/WLAN, [Online; accessed May 2013].

[10] S. Biaz and S. Wu, "Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison," in *Proc. of IEEE ISCC*, Marrakech, Morocco, 2008.

[11] F. J. Massey Jr, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[12] A. Kamerman and N. Erkocevic, "Microwave oven interference on wireless LANs operating in the 2.4 GHz ISM band," in *Proc. of PIMRC*, Helsinki, Finland, Sep. 1997.

[13] G. V. Zàruba, M. Huber, F. A. Kamangar, and I. Chlamtac, "Indoor location tracking using RSSI readings from a single Wi-Fi access point," *Wireless Networks*, vol. 13, no. 2, pp. 221–235, Apr. 2007.

[14] A. Ali, L. Latiff, and N. Fisal, "GPS-free indoor location tracking in mobile ad hoc network (MANET) using RSSI," in *Proc. of IEEE RFM*, Selangor, Malaysia, Oct. 2004.

[15] J. Hightower, R. Want, and G. Borriello, "SpotON: An indoor 3D location sensing technology based on RF signal strength," *Technical Report, UW CSE 00-02-02, University of Washington, Seattle, WA*, vol. 1, 2000.

[16] P. Kanuparthy, C. Dovrolis, K. Papagiannaki, S. Seshan, and P. Steenkiste, "Can user-level probing detect and diagnose common home-WLAN pathologies," *Computer Communication Review*, vol. 42, no. 1, pp. 7–15, 2012.

[17] A. Baid, S. Mathur, I. Seskar, S. Paul, A. Das, and D. Raychaudhuri, "Spectrum MRI: Towards diagnosis of multi-radio interference in the unlicensed band," in *Proc. of IEEE WCNC*, Quintana-Roo, Mexico, Mar. 2011.

[18] S. Sundaresan, Y. Grunenberger, N. Feamster, D. Papagiannaki, D. Levin, and R. Teixeira, "WTF? Locating performance problems in home networks," in *SCS Technical Report; GT-CS-13-03*, Jun. 2013.

[19] "20 Myths of Wi-Fi Interference," http://tinyurl.com/9rbe2f7, [Online; accessed July 2013].