Airshark: Detecting Non-WiFi RF Devices using Commodity WiFi Hardware

Shravan Rayanchu, Ashish Patro, Suman Banerjee
{shravan, patro, suman}@cs.wisc.edu
University of Wisconsin, Madison, USA

ABSTRACT

In this paper, we propose Airshark—a system that detects multiple non-WiFi RF devices in *real-time and using only commodity WiFi hardware*. To motivate the need for systems like Airshark, we start with measurement study that characterizes the usage and prevalence of non-WiFi devices across many locations. We then present the design and implementation of Airshark. Airshark extracts unique features using the functionality provided by a WiFi card to detect multiple non-WiFi devices including fixed frequency devices (e.g., ZigBee, analog cordless phone), frequency hoppers (e.g., Bluetooth, game controllers like Xbox), and broadband interferers (e.g., microwave ovens). Airshark has an average detection accuracy of $91-96\%$, even in the presence of multiple simultaneously active RF devices operating at a wide range of signal strengths (-80 to -30 dBm), while maintaining a low false positive rate. Through a deployment in two production WLANs, we show that Airshark can be a useful tool to the WLAN administrators in understanding non-WiFi interference.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

General Terms

Design, Experimentation, Measurement, Performance

Keywords

WiFi, 802.11, Spectrum, Non-WiFi, RF Device Detection, Interference, Wireless Network Monitoring

1. INTRODUCTION

The unlicensed wireless spectrum continues to be home for a large range of devices. Examples include cordless phones, Bluetooth headsets, various types of audio and video transmitters (security cameras and baby monitors), wireless game controllers (Xbox and Wii), various ZigBee devices (e.g., for lighting and HVAC controls), even microwave ovens, and the widely deployed WiFi Access Points (APs) and clients. Numerous anecdotal studies have demonstrated that links using WiFi, which is a dominant communication technology in this spectrum, are often affected by interference from all of these different transmitters in the environment. In Figure 1, we present results from our own experiments where a single, good quality WiFi link was interfered by different non-WiFi devices—an analog phone, a Bluetooth device, a videocam, an Xbox controller, an audio transmitter, a frequency hopping cordless phone, a microwave, and a ZigBee transmitter—when placed at different distances from the WiFi link.

The figure shows the normalized UDP throughput under interference, relative to the un-interfered WiFi link, as a function of the interfering signal strength from these different devices. While all of these devices impede WiFi performance to a certain degree, some of these devices, e.g., the videocam and the analog phone, *can totally disrupt WiFi communication* when they are close enough ($\geq$ 80% degradation at RSSI ≥ -70 dBm, and throughput drops to *zero* in some cases).

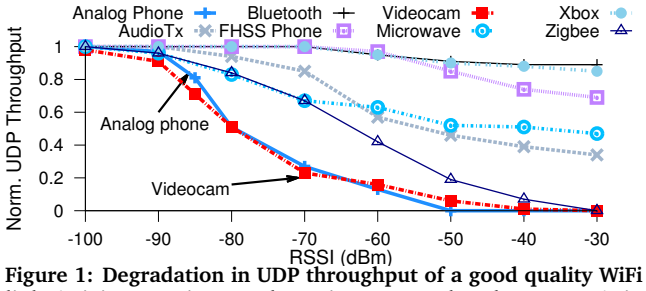

Figure 1: Degradation in UDP throughput of a good quality WiFi link (WiFi transmitter and receiver were placed 1m apart) in the presence of non-WiFi devices operating at different signal strengths.

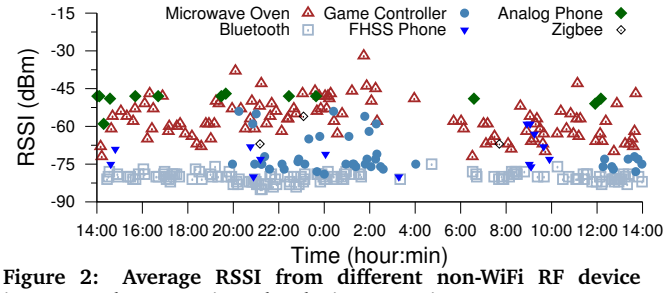

Figure 2: Average RSSI from different non-WiFi RF device instances shown against the device start times. Measurements were taken at a dorm-style apartment (location L16, dataset §2) for a 24 hour period.

IMC'11, November 2–4, 2011, Berlin, Germany.

Furthermore, our measurements across diverse home, office, and public environments, and over many weeks, show that many of these devices are routinely visible at all times of the day often at significantly high signal levels to be disruptive to WiFi links. Figure 2 shows an example of non-WiFi RF activity in a dorm-style apartment building, where some respite is observable only in the wee hours of the night.

Non-WiFi RF device detection: Traditional WiFi systems (Access Points or clients) utilize various mechanisms in the 802.11 MAC to detect and avoid interference from other WiFi sources, and are largely *oblivious* to non WiFi sources. However, with the continued growth of non-WiFi activity in this shared unlicensed band and the consequent impact on WiFi performance, hardware vendors and network administrators are increasingly exploring techniques to better detect non-WiFi sources of interference. Among commercial systems, Spectrum XT [2], AirMaestro [3], and CleanAir [4] are examples of custom hardware systems that integrate unique spectrum analyzer functionality to facilitate non-WiFi device detection. In the research community, work by Hong et. al. [17] utilizes a modified channel sounder and associated cyclostationary signal analysis to detect non-WiFi devices. RFDump [21] uses USRP GNU Radios and phase/timing analysis along with protocol specific demodulators to achieve similar goals. In this paper, we focus on techniques to detect non-WiFi RF devices but using commodity WiFi hardware instead of more sophisticated capabilities available in dedicated spectrum analyzers, expensive software radios, or any additional specialized hardware.

Using commodity WiFi hardware for non-WiFi RF device detection: Commercial spectrum analyzers or software radio platforms have specialized capability of providing "raw signal samples" for large chunks of spectrum (e.g., $80-100$ MHz) at a fine-grained sampling resolution in both time domain ($O(10^5)$ to $O(10^6)$ samples per second) and frequency domain (as low as 1 kHz bandwidth). In contrast, commodity WiFi hardware, can provide similar information albeit at a much coarser granularity. For instance, Atheros 9280 AGN cards, as available to us, can only provide RSSI measurements for an individual WiFi channel (e.g., a 20 MHz channel) at a resolution bandwidth of OFDM sub-carrier spacing (e.g., 312.5 kHz), and at a relatively coarse timescale ($O(10^3)$ to $O(10^4)$ samples per second). This capability is part of the regular 802.11 frame decoding functionality. If we can design efficient non-WiFi device detection using signal measurement samples drawn from commodity WiFi hardware, then it would be easy to embed these functionalities in all WiFi APs and clients. By doing so, each such WiFi AP and client can implement appropriate mitigation mechanisms that can quickly react to presence of significant non-WiFi interference. The following are some examples.

1) A microwave oven, which typically emits high RF energy in 2.45-2.47 GHz frequencies, turns on in the neighborhood of an AP (operating on channel 11) significantly disrupting the throughput to its clients. The AP detects this microwave, infers its disruptive properties, and decides to switch to a different channel (say, 1).

2) An AP-client link experiences short term interference from an analog cordless phone in a narrowband (< 1 MHz). The AP detects the analog phone and its characteristics, and hence decides to use channel width adaptation functions to operate on a narrower, non-overlapping 10 MHz channel instead of the usual 20 MHz channel.

Summarizing, if non-WiFi device detection is implemented using only commodity WiFi hardware, both these examples are possible natively within the AP and the client without requiring any additional spectrum analyzer hardware (either as add-on boards or chipsets) to be installed in them.

Our proposed approach — Airshark

Motivated by the above examples, we propose Airshark, a system that detects non-WiFi RF devices, *using only the functionality provided by commodity WiFi hardware.* Airshark, therefore, is a software-only solution which addresses multiple goals and challenges described next.

1) Multiple, simultaneously active, RF device detection: While Airshark can most accurately detect individual non-WiFi RF devices, it is also designed to effectively discern a small number of *simultaneously operating* non-WiFi devices, while keeping false positives low.

2) Real-time and extensible detection framework: Airshark operates in real-time allowing the WiFi node to take immediate remedial steps to mitigate interference from non-WiFi devices. In addition, its detection framework is extensible—adding hitherto unknown RF device profiles requires a one-time overhead, analogous to commercial systems based on spectrum analyzers [3].

3) Operation under limited view of spectrum: Being implemented using commodity WiFi hardware, Airshark assumes that typically only 20 MHz spectrum snapshots (equal to the width of a single WiFi channel) are available for its use in each channel measurement attempt. This limitation implies that Airshark cannot continuously observe the entire behavior of many non-WiFi frequency hoppers (e.g., Bluetooth). Further, the resolution of these samples are at least 2 orders of magnitude lower than what is available from more sophisticated spectrum analyzers [1] and channel sounders [17]. In addition to this low sampling resolution, we also observed infrequent occurances of missing samples. Finally, various signal characteristics that are available through spectrum analyzer hardware (e.g., phase and modulation properties) are not available from the commodity WiFi hardware. Therefore, Airshark needed to operate purely based on the limited energy samples available from the WiFi cards, and maintain high detection accuracy and low false positives despite these constraints.

Overview of Airshark: Airshark overcomes these challenges using several mechansisms. It uses a *dwell-sample-switch* approach to collect samples across the spectrum (§3.1). It operates using only energy samples from the WiFi card to extract a diverse set of features (§3.3) that capture the spectral and temporal properties of wireless signals. These features are robust to changes in the wireless environment and are used by Airshark's light-weight decision tree classifiers to perform device detection in real-time (§3.4). We systematically evaluate Airshark's performance in a variety of scenarios, and find its performance comparable to a state-of-the-art signal analyzer [3] that employs custom hardware.

Key contributions

In this work, we make the following contributions:

- *Characterizing prevalance of non-WiFi RF devices.* To motivate the need for systems such as Airshark, we first

RF Device Category	Device Models (set up)	Airshark's Accuracy (low RSSI — high RSSI)
High duty, fixed frequency devices — spectral signature, duty, center frequency, bandwidth		
Analog Cordless Phones	Uniden EXP4540 Compact Cordless Phone (phone call)	97.73%—100%
Wireless Video Cameras	Pyrus Electronics Surveillance Camera (video streaming)	92.7%—99.82%
Frequency hoppers — pulse signature, timing signature, pulse spread		
Bluetooth devices (ACL/SCO)	Bluetooth-enabled devices: (i) iPhone, (ii) iPod touch, (iii) Microsoft notebook mouse 5000, (iv) Jabra bluetooth headset (data transfer/audio streaming)	91.63%—99.46%
FHSS Cordless Base/Phones	Panasonic 2.4 KX-TG2343 Cordless Base/Phones (phone call)	96.47%—100%
Wireless Audio Transmitter	GOGroove PurePlay 2.4 GHz Wireless headphones (audio streaming)	91.23%—99.37%
Wireless Game Contollers	(i) Microsoft Xbox, (ii) Nintendo Wii, (iii) Sony Playstation 3 (gaming)	91.75%—99%
Broadband interferers — timing signature, sweep detection		
Microwave Ovens (residential)	(i) Whirlpool MT4110, (ii) Daewoo KOR-630A, (iii) Sunbeam SBM7500W (heating water/food)	93.16%—99.56%
Variable duty, fixed frequency devices — spectral signature, pulse signature		
ZigBee Devices	Jennic JN5121/JN513x based devices (bulk data transfer)	96.23%—99.12%

Table 1: Devices tested with the current implementation of Airshark. Features used to detect the devices include: Pulse signature (duration, bandwidth, center frequency), Spectral signature, Timing signature, Duty cycle, Pulse spread and device specific features (e.g., Sweep detection for Microwave Ovens). Accurac y tests were done in presence of multiple active RF devices and RSSI values range from -80 dBm (low) to -30 dBm (high).

performed a detailed measurement study to characterize the prevalence of non-WiFi RF devices in typical environments — homes, offices, and various public spaces. This study was conducted for more than 600 hours over several weeks across numerous representative locations using signal analyzers [3] that establish the ground truth.

- *Design and implementation of* Airshark *to detect non-WiFi RF devices.* Airshark extracts a unique set of features using the functionality provided by a WiFi card, and accurately detects multiple RF devices (across multiple models listed in Table 1) while maintaining a low false positive rate (§4). Across multiple RF environments, and in the presence of multiple RF devices operating simultaneously, average detection accuracy was 96% at moderate to high signal strengths (≥ -60 dBm). At low signal strengths (-80 dBm), accuracy was 91%. Further, Airshark's performance is comparable to commercial signal analyzers (§4.1.6).

- *Example uses of* Airshark. Through a deployment in two production WLANs, we demonstrate Airshark's potential in monitoring the RF activity, and understanding performance issues that arise due to non-WiFi interference.

To the best of our knowledge, Airshark is the first system that provides a generic, scalable framework to detect non-WiFi RF devices using only commodity WiFi cards and enables non-WiFi interference detection in today's WLANs.

2. CHARACTERIZING PREVALENCE OF NON-WIFI RF DEVICES

In this section, we aim to characterize the prevalence and usage of non-WiFi RF devices in real world networks. First, we describe our measurement equipment, and data sets.

Hardware. We use AirMaestro RF signal analyzer [3] to determine the ground truth about the prevalence of RF devices. This device uses a specialized hardware (BSP2500 RF signal analyzer IC), which generates spectral samples (FFTs) at a very high resolution (every 6 μs, with a resolution bandwidth of 156 kHz) and performs signal processing to detect and classify RF interferers accurately.

— *"Ground truth" validation.* Before using AirMaestro to understand the ground truth about the prevalence of non-WiFi devices, we benchmarked its performance in terms of (i) device detection accuracy and (ii) false positives. We activated different combinations of RF devices (up to 8 devices, listed in Table 1) by placing them at random locations and measuring the accuracy at different signal strengths (up to -100 dBm). Measurements were done during late nights to avoid any external non-WiFi interference. Our results indicate an overall detection accuracy of 98.7% with no false positives. The few cases where AirMaestro failed to detect the devices occurred when the devices were operating at very low signal strengths (≤ -90 dBm).

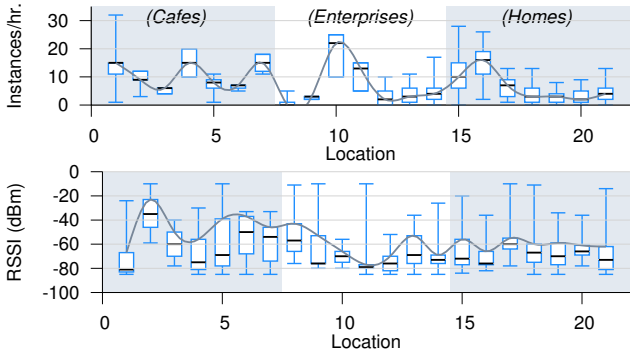

Figure 3: Distribution of (i) non-WiFi device instances/hour at different locations (top), and (ii) RSSI of non-WiFi devices at these locations (bottom). Min, Max, 25th, 50th and 75th percentiles are shown.

Data sets. We collected the RF device usage measurements using the signal analyzer at 21 locations for a total of 640 hours. We broadly categorize these locations into three categories: (i) *cafes (L1-L7)*: these included coffee shops, malls, book-stores (ii) *enterprises (L8-L14)*: offices, university departments, libraries and (iii) *homes (L15-L21)*: these included apartments and independent houses. Measurements were taken over a period of 5 weeks. At some locations, we could collect data for more than 24 hours (e.g., enterprises, homes) but for others we could collect measurements only during the day times (e.g., coffee shops, malls). We now summarize our observations from this data.

Non-WiFi devices are prevalent across locations and often appear with fairly high signal strengths.

Figure 3 (top) shows the distribution of non-WiFi device instances observed per hour in different wireless environments.

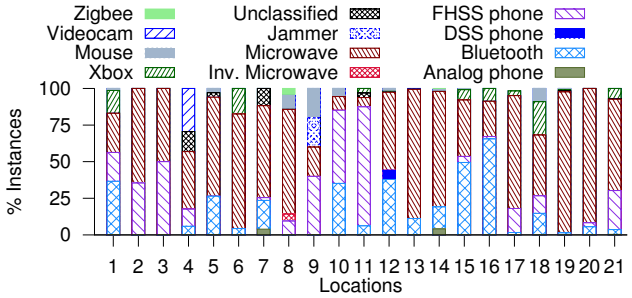

Figure 4: Distribution of non-WiFi device instances at various locations.

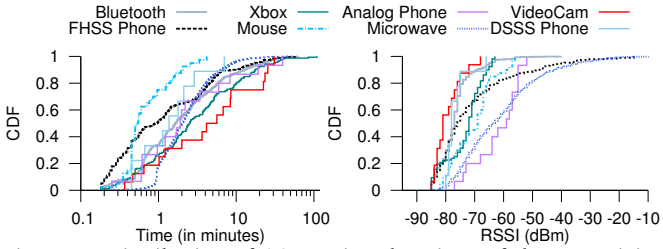

Figure 5: Distribution of (a) Session durations of the non-WiFi device instances (X-axis in log-scale) and (b) RSSIs of the non-WiFi device instances aggregated across all locations.

We observe that device instances/hr. varied across locations, with some locations showing very high device activity (e.g., a median of 22, 16 instances/hr. at locations L10 (an office), L16 (dormitory) respectively). Figure 3 (bottom) shows the distribution of non-WiFi device RSSI at these locations[1]. We observe that the median RSSI varied from -80 to -35 dBm. Further, for around 62% locations, we observe that 75th percentile of RSSI was greater than -60 dBm (shown using a gray line) suggesting strong non-WiFi interference.

Popularity of devices varied with locations, although few devices are popular across many locations.

Figure 4 shows the distribution of non-WiFi instances at different locations. Microwaves, FHSS cordless phones, Bluetooth devices and game controllers were the most popular. However, some other devices appeared frequently at specific locations e.g., video cameras accounted for 29% of instances at location L4 (cafe).

Session durations for non-WiFi devices varied from a few seconds to over 100 minutes.

Figure 5 (left) shows the CDF of the session times for each class of non-WiFi devices. Many devices appear in our traces for a short duration ($\leq$ 2 minutes). These included (i) devices like microwaves that are activated for short durations and (ii) device instances with low signal strengths (≤ -75 dBm) that appeared intermittently at the locations where the signal analyzer was placed. However, for 25% of the cases, the devices were active for more than 5 minutes and in some traces, devices like game controllers (e.g., Xbox) were active for durations of up to 1.8 hours.

[1]Observed RSSI is dependent on the exact location where the measurement node was placed. While we tried to be unbiased, node placement in reality was influenced by few factors like availability of power connection.

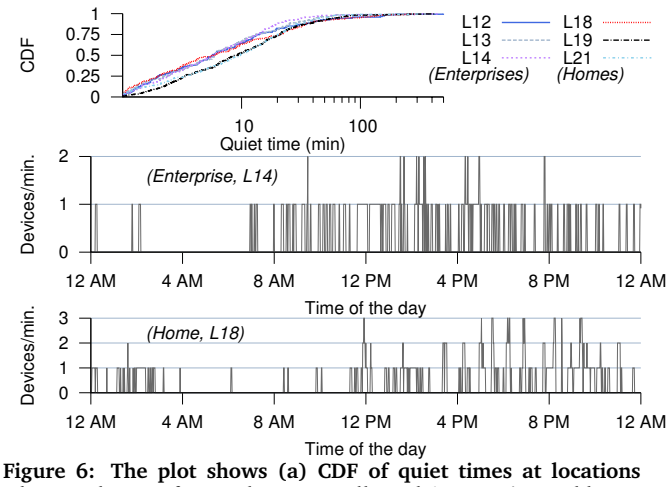

Figure 6: The plot shows (a) CDF of quiet times at locations where 48 hours of trace data was collected (enterprise and home locations). Time-series of non-WiFi device instances per minute at (b) an enterprise (location L14) and (c) and home (location L18) for a 24 hour period shows increased quiet times during late nights.

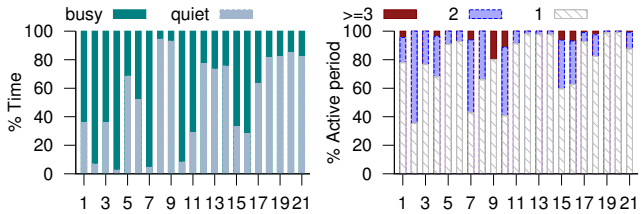

Figure 7: Distribution of (a) quiet and busy periods (b) simultaneously-active devices during the busy periods across different locations

Some of the devices operate at high signal strengths indicating potential interference.

Microwave ovens, video cameras, and analog phones were the most dominant in terms of RSSI (Figure 5 (right)). For e.g., RSSI was ≥ -55 dBm for more than 35% of the observed microwave oven instances, indicating potential interference to nearby WiFi links. Wireless game controllers and Bluetooth devices on the other hand, mostly occurred at low to moderate RSSI of -80 to -65 dBm.

More than 50% of the periods with no non-WiFi device activity were less than 10 minutes.

We define a *quiet period* as a duration in which no non-WiFi devices were active. Figure 6 (top) shows the CDF of the quiet periods for locations (homes, enterprises) at which we collected data for 48 hours. The figure shows that non-WiFi devices appeared quite frequently—more than 50% of the quiet periods were less than 10 minutes, and maximum quiet period in our trace was 8 hours at L12 (office). Figure 6 (middle, bottom) shows a 24 hour time-series of the number of active non-WiFi devices/min. at L14 (enterprise) and L18 (home) respectively. We observe that the longer quiet periods occurred during late nights to early mornings (3 am to 8 am), and most of the non-WiFi device activity was during the daytime, when WiFi utilization is also typically significant.

At most locations, only 1 or 2 non-WiFi devices were active simultaneously for more than 95% of the busy times.

Figure 7 (left) shows that the aggregate quiet times (percentage time with no non-WiFi activity) vary at different locations. In many locations, the aggregate quiet times were around $70-80\%$ *i.e.*, non-WiFi devices were visible for $20-30\%$ of the time. Quiet times were much lesser for cafes (e.g., only 3% at L4) as the traces did not include the measurements during the night times and there was frequent microwave oven and cordless phone activity when we collected the traces (during the day time). Figure 7 (right) shows the distribution of the number of active non-WiFi devices per minute, during the busy times (periods with non-WiFi activity). We find that only a single device was active for 35% (L2, cafe) to 99% (L20, home) of the time, and more than 2 devices were active simultaneously for at most 20% of the time (L9, office).

3. Airshark: DEVICE DETECTION

Prior work has developed device detection mechanisms using commercial signal analyzers [3], channel sounders [17] or software radios [21] which offer fine-grained, very high resolution signal samples, typically collected using a wide-band radio. We focus on designing such a system *using commodity WiFi cards*. WiFi cards are capable of providing similar spectrum data, albeit with limited signal information, and 2 orders of magnitude lesser resolution compared to signal analyzers. Traditionally, WiFi cards have not exposed this functionality, but emerging commodity WiFi cards provide an API through which we can access *spectral samples*—information about the signal power received in each of the sub-carriers of an 802.11 channel, which opens up the possibility of detecting non-WiFi devices. Designing such a detection system using WiFi cards, however, imposes several challenges as discussed below.

Why is it hard to detect devices using WiFi cards?

- *Limited spectrum view.* Unlike sophisticated signal analyzers [1,17] that can sample a wideband of $80-100$ MHz (e.g., the entire 2.4 GHz band), current WiFi cards are designed to operate in a narrowband (e.g., 20 MHz).
- *Limited signal information.* Current WiFi cards provide limited signal information (e.g., the received power per sub-carrier) compared to software radios that provide raw signal samples. Thus traditional device detection approaches like cyclostationary analysis [17], phase analysis or use of protocol specific decoders [21] are not feasible.
- *Reduced sampling resolution.* WiFi cards have a resolution bandwidth of 312.5 kHz (equal to sub-carrier spacing) compared to signal analyzers [1] that offer resolution bandwidths as low as 1 kHz. Further, WiFi cards also have a lower sampling rate—our current implementation uses $\sim$2.5k samples/sec, as opposed to that used by commercial signal analyzers [3] (160k samples/sec) or prior work [21] employing software radios (8 million samples/sec).
- *Other challenges.* Coupled with the above constraints, the presence of regular WiFi packet transmissions in the spectrum further increase the "noise" in the spectral samples, making it more challenging to detect devices.

Overview. We wish to detect the presence of multiple (simultaneously operating) non-WiFi devices in real-time. The above challenges imply that Airshark is constrained to use the *limited signal information* (spectral samples) provided by a WiFi card and must employ *light-weight* detection mechanisms that are *robust to missing samples*. We now present an overview of Airshark's device detection pipeline. Figure 8 illustrates the four steps listed below.

Delay	minimum	25th pc.	median	75th pc	maximum
Inter-sample time	116μs	116μs	122μs	147μs	4.94 ms
Sub-band switching	12.2 ms	14.5 ms	19.7 ms	31 ms	163 ms

Table 2: Time between valid consecutive spectral samples and time taken to switch sub-bands in our current implementation.

1. Spectral samples from the WiFi card are generated using a scanning procedure (§3.1). This procedure divides the entire spectrum into a number of *sub-bands* and generates spectral samples for each sub-band. Samples that comprise only WiFi transmissions are "purged" and remaining samples are passed to the next stage.
2. Next, spectral samples are processed to detect signal *pulses*—time-frequency blocks that contain potential signals of interest, and collect some aggregate statistics based on received power values (§3.2).
3. In this stage, we extract a set of *light-weight* and unique features from the pulses and statistics (§3.3)—derived using only received power values—that capture the spectral and temporal properties of signals. Example features include: *spectral signatures* that characterize the shape of the signal's power distribution across its frequency band, *inter-pulse timing signatures* that measure the time difference between the pulses, and device specific features like *sweep detection* (used to detect microwave ovens).
4. In the final stage of the pipeline, the above features are used by different device analyzers that employ decision tree models (§3.4) trained to detect their target device.

We now explain the above detection procedure in detail.

3.1 Spectral Sampling

We start by explaining the details of sampling procedure employed in our current implementation.

Spectral samples. We implement Airshark using an Atheros AR9280 AGN wireless card. We use the card in 802.11n 20 MHz HT mode, where a 20 MHz channel is divided into 64 sub-carriers, spaced 312.5 KHz apart and the signal data is transmitted on 56 of these sub-carriers. Each spectral sample (FFT) generated by the wireless card comprises the power received in 56 sub-carriers (FFT bins) and corresponds to a 17.5 MHz (56×0.3125 MHz) chunk of spectrum, which we refer to as a *sub-band*. Additionally, the wireless card also provides the timestamp t (in μs) at which the sample was taken, and the noise floor at that instant.

Purging WiFi spectral samples. We efficiently filter the spectral samples that comprise only WiFi transmissions as follows: all the samples for which Airshark's radio is able to successfully decode a WiFi packet are marked as potential candidates for purging. Airshark then reports a spectral sample for further processing only if it detects non Wi-Fi energy in that sample. To be more precise, if the radio is receiving a packet, Airshark will not report the sample unless the interference signal is stronger than the 802.11 signal being received. One downside to this approach is that Airshark will also report spectral samples corresponding to weak 802.11 signals that fail carrier detection. However, as we show in §3.3, this is not a problem as Airshark can filter out the

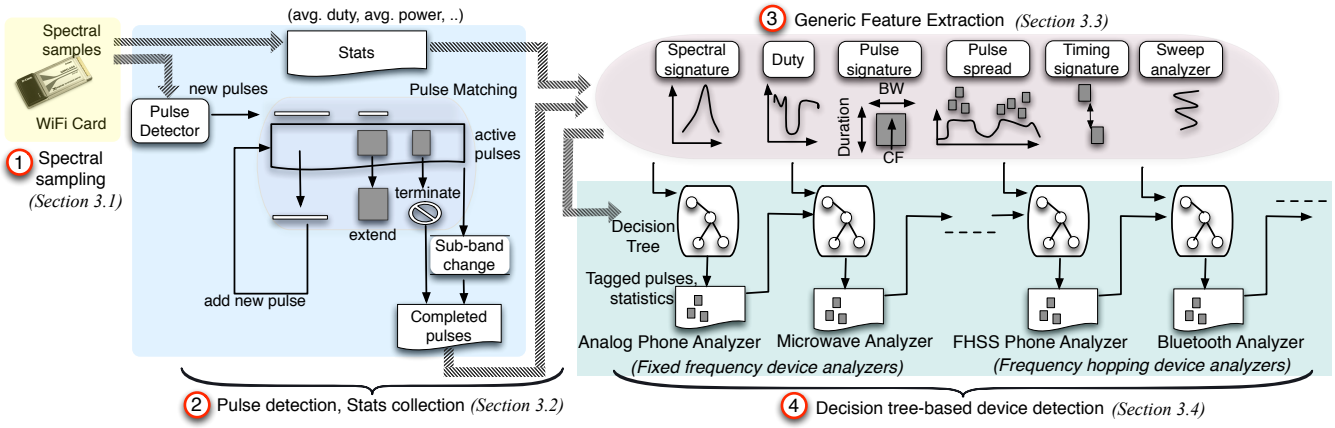

Figure 8: (a) Illustration of Airshark's detection pipeline. Spectral samples from the WiFi card are generated using a scanning procedure (§3.1). These samples are processed to detect signal *pulses*, and collect some aggregate statistics based on the received power values (§3.2). In the next stage, various features capturing the spectral and temporal properties of the signals are extracted (§3.3), and are used by different device analyzers that employ decision tree models (§3.4) trained to detect their target RF devices.

samples relevant to non-WiFi transmissions by employing device detection mechanisms. We term the samples reported by Airshark after this purging step as *valid* spectral samples.

Scanning procedure. Airshark divides the entire spectrum (e.g., 80 MHz) into several (possibly overlapping) sub-bands, and samples one sub-band at a time. Our current implementation uses 7 sub-bands with center frequencies corresponding to the WiFi channels 1, 3, 6, 9, 11, 13 and 14. Table 2 shows (i) inter-sample time: the time between two consecutive valid spectral samples (within a sub-band) and (ii) time taken to switch the sub-bands. Increased gap in the inter-sample time for a few samples ($\geq 150\mu s$) is due to the nature of the wireless environment—in the absence of strong non-WiFi devices transmissions, intermittent interference from WiFi transmissions causes gaps due to purged spectral samples. Sampling gaps are also caused when switching sub-bands ($\sim$ 20 ms on an average, and 163 ms in the worst case).

To amoritize the cost of switching sub-bands, Airshark employs a *dwell-sample-switch* approach to sampling: Airshark dwells for 100 ms in each sub-band, captures the spectral samples and then switches to the next sub-band. As we show later, in spite of the increased gap for few samples, we find the sampling resolution of current WiFi cards to be adequate in detecting devices (across different wireless environments) with a reasonable accuracy (§4). In §4, we demonstrate the adversarial case where strong WiFi interference coupled with weak non-WiFi signal transmissions can affect Airshark's detection capabilities.

3.2 Extracting signal data

We now explain the next stage in the detection pipeline that operates on the spectral samples to generate signal *pulses*, along with some aggregate statistics.

"Pulse" Detection. Each spectral sample is processed to identify the signal "peaks". Several complex mechanisms have been proposed for peak detection [11, 16]. To keep our implementation efficient, we use a simple and a fairly standard algorithm [12,14]—peaks are identified by searching for "local maximas" that are above a minimum energy threshold γ_s. For each peak, the pulse detector generates a *pulse* as a set

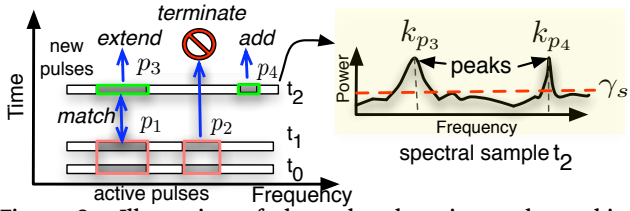

Figure 9: Illustration of the pulse detection and matching procedure. Pulse detector processes the spectral sample at time t_2 to output two new pulses p_3 and p_4. New pulse p_3 matches with the active pulse p_1, and results in extending p_1. Active pulse p_2 is terminated as there is no matching new pulse, and new pulse p_4 is added to the active pulse list.

of contiguous FFT bins that surround this peak. A pulse corresponds to a signal of interest, and its start and end frequencies are computed as explained below.

— *frequency and bandwidth estimation:* Let k_p denote the peak bin and $p(k_p)$ denote the power received in this bin. We first find the set of contiguous FFT bins $[k'_s, k'_e]$ such that $k'_s \leq k_p \leq k'_e$ and power received in each bin is (i) above the energy threshold, γ_s and (ii) within δ_B of the peak power $p(k_p)$ *i.e.*, $p(k) \geq \gamma_s \bigwedge p(k_p) - p(k) < \delta_B \forall k \in [k'_s, k'_e]$. The center frequency (CF) and the bandwidth (BW) of a pulse corresponding to this peak bin can be characterized by considering its mean localization and dispersion in the frequency domain:

$$k_c = \frac{1}{\sum_k p(k)} \sum_k k \cdot p(k), \;\; k'_s \leq k \leq k'_e$$

$$B = 2 \sqrt{\frac{1}{\sum_k p(k)} \sum_k (k - k_c)^2 \cdot p(k)}, \;\; k'_s \leq k \leq k'_e$$

The center frequency bin k_c is computed as the center point of the power distribution, and the frequency spread around this center point is defined as the bandwidth B of the pulse. For each peak, we restrict the bandwidth of interest to comprise bins whose power values are more than γ_s and are within δ_B of the peak power $p(k_p)$. We use this mechanism as it is simple to compute and it provides reasonable estimates as we show in §4. Based on the computed bandwidth, the start bin (k_s) and the end bin (k_e) are determined. The pulse detector

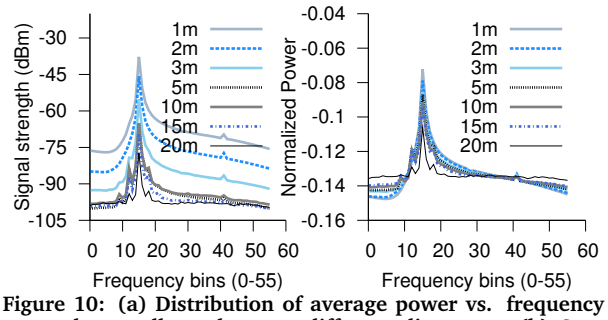

Figure 10: (a) Distribution of average power vs. frequency for an analog cordless phone at different distances. (b) Spectral signatures for the analog cordless phone are not affected for RSSI values of ≥ -80 dBm.

can potentially output multiple pulses for a spectral sample. Each pulse in a spectral sample observed at time t can be represented using the tuple $[t, k_s, k_c, k_e, [p(k_s) \ldots p(k_e)]]$

Pulse Matching. Airshark maintains a list of *active pulses* for the current sub-band. This active pulse list is empty at the start of the sub-band, and the first set of pulses (obtained after processing a spectral sample in the sub-band) are added to this list as active pulses. For the rest of the samples in the sub-band, a pulse matching procedure is employed: the pulse detector outputs a set of *new pulses* after processing the sample. These new pulses are compared against the list of active pulses to determine a match. In our current implementation, we use a strict criteria to determine a match between a new pulse and an active pulse: the CFs and BWs of the new pulse and the active pulse must be equal, and their peak power values must be within 3 dB (to accommodate signal strength variations). Once a match is determined, the new pulse is merged with the active pulse to *extend* it *i.e.*, the duration of the active pulse is increased to accommodate this new pulse, and the power values of the active pulse are updated by taking a weighted average of power values of the new and the active pulse.

After the pulse matching procedure, any left over new pulses in the current spectral sample are added to the active pulse list. The active pulses that did not find a matching new pulse in the current sample are terminated. Active pulses are also terminated if Airshark encounters more than one missing spectral sample (*i.e.*, inter-sample time $\geq$ 150 μs). Once an active pulse is terminated, it is moved to the current sub-band's list of *completed pulses*. It is possible that some of the active pulses are prematurely terminated due to the strict match and termination criteria. However, doing so helps Airshark maintain a low false positive rate as it only operates on well-formed pulses that satisfy this strict criteria (§4). Figure 9 illustrates this pulse detection procedure.

Stats Module. The stats module operates independently of the above pulse logic. It processes all the spectral samples of a sub-band to generate the following statistics: (i) *average power*: this is the average power in each FFT bin for the duration of the sub-band, (ii) *average duty*: this is the average duty cycle for each bin in the sub-band. The duty cycle of an FFT bin k is computed as 1 if $p(k) \geq \gamma_s$, otherwise it is 0. (iii) *high duty zones*: After processing a sub-band, a mechanism similar to peak detection, followed by CF and BW estimation procedure is applied on the "average power" statistic to identify the high duty zones in the sub-band. These are used to quickly detect the presence of high duty devices.

Before switching to the next sub-band, all the active pulses for the current sub-band are terminated and pushed to the list of the sub-band's completed pulses. The list of completed pulses along with the aggregate statistics are then passed on to the next stage of the pipeline to perform feature extraction.

3.3 Feature Extraction

Using the completed pulses list and statistics, we extract a set of *generic features* that capture the spectral and temporal properties of different non-WiFi device transmissions. These features—frequency, bandwidth, spectral signature, duty cycle, pulse signature, inter-pulse timing signature, pulse spread and device specific features like sweep detection—form the building blocks of Airshark's decision tree-based device detection mechanisms. We now explain these features.

(F1) Frequency and Bandwidth. Most RF devices operate using pre-defined center frequencies, and their waveforms occupy a specific bandwidth. For e.g, a ZigBee device operates on one of the pre-defined 16 channels [22], and occupies a bandwidth of 2 MHz. The center frequency and bandwidth of the pulses (and sub-band's high duty zones) are used as features in Airshark's decision tree models.

(F2) Spectral signatures. Many RF devices also exhibit certain power versus frequency characteristics. We capture this using a *spectral signature*: given a set of frequency bins $[k_s \ldots k_e]$ and corresponding power values $[p(k_s) \ldots p(k_e)]$, if we treat the frequency bins as a set of orthogonal axes, we can construct a vector $\overrightarrow{s} = p(k_s)\hat{k_s} + \ldots + p(k_e)\hat{k_e}$ that represents the power received in each of the bins. We then normalize this vector to derive a unit vector representing the spectral signature: $\hat{s} = \frac{\overrightarrow{s}}{|s|}$. Given a reference spectral signature $\hat{s}_r$ and a measured spectral signature $\hat{s}_m$, we compute the similarity between the spectral signatures as the *angular difference* (θ): $\cos^{-1}(\hat{s}_r \cdot \hat{s}_m)$. The angular difference captures the degree of alignment between the vectors, and is close to 0° when the relative composition of the vectors is similar.

Spectral signatures can be computed on the average power values of the pulses (e.g., ZigBee pulse) or on the high duty zones (e.g., for high duty devices like analog phones) to aid in device detection. Figure 10 shows the power distribution of an analog cordless phone at different distances, and the corresponding spectral signatures computed at each distance. The figure shows that normalization aids in making the signatures robust to the changes in the signal strengths of the RF devices. However, at very low signal strengths (≤ -90 dBm), the spectral signatures tend to deviate and result in an increased theta, leading to false negatives (§4).

(F3) Duty cycle. The duty cycle $\mathcal{D}$ of a device is the fraction of time the device spends in "active" state. This can be used to identify high duty devices, e.g., analog phones and wireless video cameras have $\mathcal{D}$=1, or identify devices with characteristic duty cycles e.g., microwave ovens have $\mathcal{D}$=0.5. In reality, due to the presence of multiple devices, it is possible for the duty cycle of the bandwidth (FFT bins) used by a device to be more than its expected duty cycle. We therefore use the notion of *minimum* duty cycle $\mathcal{D}_{min}$ for devices ($\mathcal{D}_{min}$=0.5 for a microwave oven) as one of the features.

Protocol/Device	Bandwidth	Duration	Frequency usage
WDCT Cordless Phone	0.892 KHz	700 μs	FHSS, 90 channels
Bluetooth	1 MHz	366 μs - 3 ms	FHSS, 79 channels
ZigBee	2 MHz	$<$ 5 ms	Static, 16 channels
Game controller	500 KHz	235 μs	FHSS, 40 channels

Table 3: Pulse signatures for different RF devices.

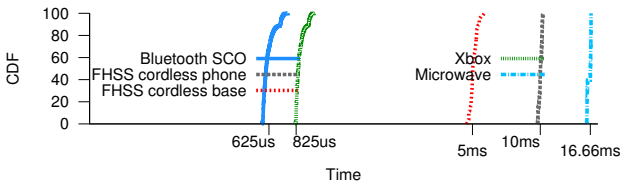

Figure 11: Inter-pulse timing signature for different devices.

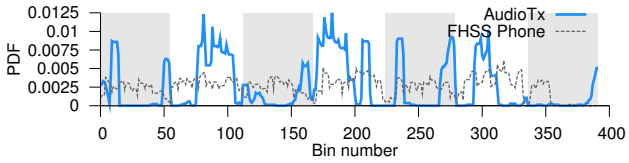

Figure 12: Pulse distribution of FHSS cordless phone and an audio transmitter as captured by Airshark.

(F4) Pulse signatures. Along with CF and BW, the transmission durations of many devices conform to their protocol standards. For e.g., in Bluetooth, the duration of a transmission slot is 625 μs, out of which 366 μs is spent in active transmission. Similarly, WDCT cordless phones (FHSS phones) have a pulse duration of 700 μs. Table 3 shows these properties (frequency, bandwidth, and duration of the pulses) for different devices. Airshark combines these three properties together to define *pulse signatures* for devices that communicate using pulses (e.g., ZigBee, Bluetooth) and uses them as features in the detection process.

(F5) Inter-pulse timing signatures. Timing between the transmissions of many devices also exhibit certain properties. In Bluetooth SCO, for example, examining the spectrum will reveal sets of two consecutive pulses that satisfy the Bluetooth pulse signature (Table 3) and are separated by a time difference of 625 μs. WDCT cordless phones and game controllers (e.g., Wii) exhibit similar properties with time difference between consecutive pulses (occuring at the same center frequency) in a set being 5 ms and 825 μs respectively. Similarly, microwaves exhibit an ON-OFF cycle with a period of 16.6 ms. Figure 11 illustrates these timing properties.

Since Airshark can only sample a particular sub-band at a time, it cannot capture all the pulses of a device. This is especially true for frequency hopping devices. Due to the nature of sampling, we cannot expect every captured pulse to exhibit the above timing property. Airshark's device analyzers therefore use a relaxed constraint—number of pulse sets that satisfy a particular timing property is used as one of the features in the decision tree models (§3.4).

(F6) Pulse spread. Airshark accumulates the pulses for a number of sub-bands, and extracts features from pulses belonging to a particular pulse signature to detect the presence of frequency hopping devices. Together, these features represent the *pulse spread* across different sub-bands.

1. Pulses-per-band (mean and variance). We use the average number of pulses per sub-band, and the corresponding variance as one of the measures to characterize the pulse spread. For frequency hoppers, we can expect the average number of pulses in each sub-band to be higher (and the variance lower) compared to fixed frequency devices.

2. Pulse distribution. Pulses of many frequency hopping devices tend to conform to a particular distribution. For example, FHSS cordless phone pulses are spread uniformly across the entire 80 MHz band, whereas, the pulse distribution for other frequency hoppers like audio transmitters may tend to be concentrated on certain frequencies of sub-bands, as shown in Figure 12. The X-axis shows the bin number b for each of the seven sub-bands ($b_{max} = 56 \times 7 = 392$), and Y-axis shows the fraction of the pulses that fall into each bin.[2]

Airshark checks whether the distribution of pulses across the sub-bands conforms to an expected pulse distribution using Normalized Kullback-Leibler Divergence (NKLD) [20], a well known metric in information theory. NKLD is simple to compute and can be used to quantify the 'distance' or the relative entropy between two probability distributions. NKLD is zero when the two distributions are identical, and a higher value of NKLD implies increased distance between the two distributions. The definition of NKLD is assymetric, therefore we use a symmetric version of NKLD [20] to compare two distributions. Let $r(b)$ be the reference pulse distribution over all the bins ($b \in \mathcal{B} = [0, b_{max}]$), computed over a large period of time. Let $m(b)$ be the measured pulse distribution over a smaller time period t_m. The symmetric NKLD for two distributions $r(b)$ and $m(b)$ can be defined as:

$$\text{NKLD}(m(b), r(b)) = \frac{1}{2}\Big(\frac{D(m(b)\|r(b))}{H(m(b))} + \frac{D(r(b)\|m(b))}{H(r(b))}\Big)$$

where, $D(m(b)\|r(b))$ quantifies the divergence and is computed as $\sum_{b\in\mathcal{B}} m(b)\left|\log\frac{m(b)}{r(b)}\right|$, and $H(m(b))$ is the entropy of the random variable b with distribution $m(b)$ *i.e.*, $H(m(b)) = -\sum_{b\in\mathcal{B}} m(b)\log_2 m(b)$.

While Airshark can measure the pulse distribution $m(b)$ over a large time scale, and check if it conforms to $r(b)$, this will increase the time to detect the device. This leads to the question, "what is the minimum time scale t_m at which the pulse distribution can be measured?" We chose this time scale by empirically measuring how the NKLD values converge with the increase in the number of samples under different conditions. For the devices that we tested, we observed around 15000 samples (around 6 scans of the entire 80 MHz band, amounting to 6−7 seconds) was sufficient. We show how the number of samples affect the NKLD values in §4.2.3. We note that not all devices conform to a particular pulse distribution e.g., in our experiments, we found variable pulse distributions for Bluetooth as it employs adaptive hopping mechanisms.

(F7) Device specific features. Detection accuracy can be improved by using features unique to the target device. We illustrate this using a feature specific to microwave ovens.

— *Sweep detector.* The heating source in a residential

[2]Instead of measuring the actual pulse distribution over the 80 MHz band, we stitch the sub-bands together (ignoring the overlaps) and measuring the pulse distribution over the stitched sub-bands.

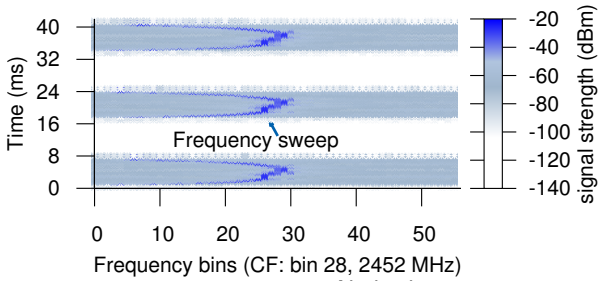

Figure 13: Spectral samples from Airshark capturing the activity of a residential microwave. The plot shows (i) the ON-OFF cycle for is around 16.6 ms and (ii) "frequency sweeps" during the ON periods.

microwave oven is based on a single magnetron tube that generates high power electromagnetic waves whenever the input voltage is above some threshold. This results in an ON-OFF pattern, typically periodic with a frequency of 60 Hz (frequency of the AC supply line). Although there might be differences between the emissions from ovens of different manufacturers, the peak operational power is mostly around 2.45-2.47 GHz and during the ON periods, the radiated signal exhibits a *frequency sweep* of around $4-6$ MHz [19, 27].

Figure 13 shows the resulting 16.66 ms periodic ON-OFF pattern and the frequency sweeps during the ON periods of a microwave oven as captured by Airshark. In the current prototype of Airshark, the microwave oven analyzer includes sweep detection, along with timing signature analysis. We tested 6 microwaves (from different manufacturers), and Airshark was able to detect all of them using these features.

3.4 Device Detection

Airshark uses decision tree [24] based classifiers in order to detect the presence of RF devices. A decision tree is a mapping from observations about an item (feature set) to conclusions about its target value (class). It employs a supervised learning model where a small set of labeled data, referred to as training data, is first used to build the tree and is later used to classify unlabeled data. In Airshark, we use the popular C4.5 algorithm [24] to construct the decision trees. For further details about mechanisms to build decision trees and the classification process, we refer the readers to [24].

Airshark employs a separate analyzer for each class of devices. These device analyzers operate on a subset of features described previously, and make use of decision tree classifiers trained to detect their corresponding RF devices. The advantages of using per-device classifiers are three-fold: (i) each classifier can use a separate feature subset, (ii) classifiers can operate at different time granularities e.g., fixed frequency device analyzers (e.g., analog phone) can carry out the classification when Airshark finishes processing a sub-band, whereas for frequency hopping device analyzers like (e.g., Bluetooth, game controllers) the classification decision can only take place after enough samples have been processed (§3.3), (iii) classification process is more efficient when multiple devices are simultaneously active—each classifier outputs either label 1 (indicating the presence of the device), or label 0 (indicating the absence of the device). The alternative approach of using a single classifier is cumbersome as it requires training the classifier for all possible device combinations (each with a separate label).

Training. Before Airshark can identify a new RF device, its features have to be recorded for training. To do this, features relevant to this device are identified, and then extracted from spectral samples for the cases when the device is *active in isolation* (label 1), and when the device is inactive (label 0). For example, when adding the analog phone analyzer, we collected the spectral samples when the phone was activated in isolation and when the phone was inactive. We then instantiated analog phone's device analyzer to extract these features: bandwidth, spectral signature and duty cycle (measured from the recorded spectral samples) and the list of possible CFs the phone can operate on. It is worth pointing out that identifying the relevant feature set for a device and training the corresponding device analyzer is a *one time overhead before adding a new RF device* to Airshark. Table 1 lists the feature set employed by device analyzers in our current implementation.

Classification. We now summarize Airshark's detection pipeline. Each sample is processed by the first stage of the pipeline, and results in updating the completed pulse list and aggregate statistics. Device analyzers are invoked when Airshark finishes processing a sub-band:

1. Each device analyzer operates on the completed pulses and aggregate statistics, to derive its features. The features may include: CF, BW, angular difference (corresponding to its spectral signature), duty cycle, number and the spread of the pulses satisfying its pulse signature and timing signature.
2. The device analyzer's decision tree is invoked to output either label 1 or 0.
3. In case the decision tree outputs label 1, Airshark invokes a module that tags the selected pulses (satisfying the pulse signature and timing signature) as "owned" by this RF device.

An additional check is performed for frequency hopping device analyzers: if there are not enough accumulated samples to perform the classification, the classification decision is deferred to the next sub-band.

Dealing with multiple RF devices and overlapping signals. When multiple RF devices are simultaneously active, the spectrum may be occupied by a large number of transmissions (signal pulses). If the transmissions from multiple devices do not overlap in time or in frequency (either because of the diversity in the device transmission times, or because the devices operate in a non-overlapping spectrum bands), Airshark's device analyzers can proceed as is. Further, for certain combinations of devices, transmissions may overlap in both time and frequency, but not always. For example, this is the case when frequency hopping devices and fixed-frequency, low duty devices are present. In our benchmarks for these combinations, Airshark could always find enough pulses that do not overlap, and therefore was able to correctly detect the devices.

Transmissions from multiple devices that *always* overlap in time and frequency, however, can decrease the detection accuracy if the above techniques are used as is. For example, if the transmissions from a fixed-frequency, always-on device (e.g., analog phone) overlap in frequency with another fixed-frequency device (e.g., ZigBee device), features like spectral signatures will not perform well. This is because overlapping signals change the "shape" of the power distribution and increase the angular difference as shown in Figure 14(a). One

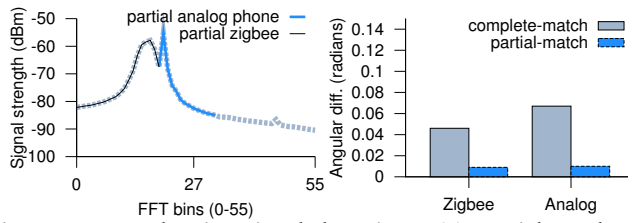

Figure 14: Overlapping signal detection. (a) partial overlap between ZigBee and analog signals (b) using partial matches between spectral signatures reduces the angular difference in overlapping cases.

approach to resolve such overlaps, is to use cyclostationary analysis [17] on raw signal samples. Such rich signal information (very high resolution, raw signal samples), however, is not available through WiFi cards. We extend the basic approach used in Airshark to handle these cases as follows: device analyzers first identify the potential peaks that match their CFs. For each peak, instead of a complete match on the signal's bandwidth BW, a *partial match* of the spectral signatures is performed. The bandwidth BW_{par} for the partial match is decided by the bandwidth detection algorithm, and is required to be above a minimum bandwidth BW_{min} in order to control the false positives (*i.e.*, $\text{BW}_{min} \leq \text{BW}_{par} \leq \text{BW}$). In our benchmarks, setting BW_{min} to $0.6\times$BW improved the accuracy without increasing the false positives (§4). Figure 14(b) shows the reduction in angular difference when using a partial match.

Alternative classifiers. We also built another classifier based on support vector machines (SVM) [5]. In our experiments, we found that in most cases, Airshark's SVM-based and decision tree based classifiers had similar detection accuracies. In some scenarios involving multiple devices, SVM-based classifier performed slightly better (§4.1.5). However, we elected to use a decision tree based classifier, as it has very low memory and processing requirements, thus making it feasible to embed non-WiFi device detection functionality in commodity wireless APs and clients.

4. EXPERIMENTAL RESULTS

In this section, we evaluate Airshark's performance under a variety of scenarios, and present real-world applications of Airshark through a small scale deployment. We start by presenting the details of our implementation and testbed set up, followed by the metrics used for evaluation.

Implementation. Our implementation of Airshark consists of few hundred lines of C code that sets up the FFT sampling from the Atheros AR9280 AGN based wireless card, and about 4500 lines of Python scripts that implement the detection pipeline. We used an off-the-shelf implementation of the C4.5 decision tree algorithm [6] for training and classification. For the alternative SVM-based classifier, we used an SVM implementation [5] employing a radial basis function kernel with default parameter setting. We focus on detecting devices in 2.4 GHz spectrum, and our current prototype has been tested with 8 classes of devices (across multiple device models) mentioned in Table 1.

Evaluation set up. We performed all our experiments in a university building (except those in §4.1.4). Our training data was taken during the late evenings and night times to minimize the impact of external non-WiFi interference. Our

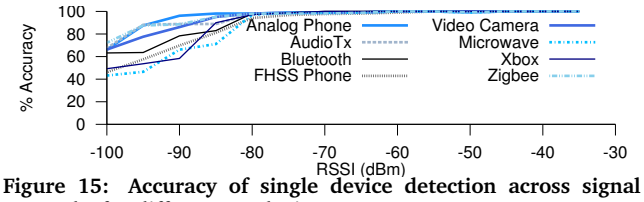

Figure 15: Accuracy of single device detection across signal strengths for different RF devices.

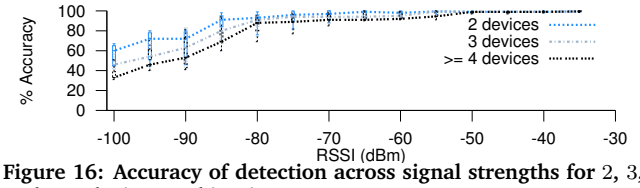

Figure 16: Accuracy of detection across signal strengths for 2, 3, and ≥ 4 device combinations.

evaluation experiments, however, were performed over a period of one week that included both busy hours and night times. We also used the AirMaestro signal analyzer [3] in order to determine the "ground truth" about the presence of any external non-WiFi RF devices during our experiments.

Evaluation Metrics. We use the following metrics to evaluate the performance of Airshark:
1. Detection accuracy: This is the fraction of correctly identified RF device instances. This estimates the probability that Airshark accurately detects the presence of an RF device.
2. False positive rate (FPR): This is defined as the fraction of false positives. This estimates the probability that Airshark incorrectly determines the presence of an RF device.

We will first evaluate Airshark's performance in various scenarios, and then comment on the parameters we chose. We set the energy threshold γ_s to -95 dBm, δ_B to 10 dB, and for computing NKLD we use 15000 samples. The RF devices tested and the features used are listed in Table 1.

4.1 Performance evaluation

We start by evaluating Airshark using controlled experiments with different RF devices.

4.1.1 Single device detection accuracy

Method. We measured the accuracy of device detection when only one of the RF devices mentioned in Table 1 was activated. The methodology used to activate the devices is also listed in Table 1. We placed the devices at random locations to generate the samples at different RSSI values, and then computed the average detection accuracy at each RSSI.

Results. Figure 15 shows the detection accuracy as a function of RSSI for different RF devices. We observe that Airshark achieves an accuracy of 98% for RSSI values as low as -80 dBm. For RSSI values ≤ -80 dBm, the accuracy drops down due to the reduced number of pulses detected at such low signal strengths. Further, the drop is sharper for frequency hopping devices, compared to fixed frequency, high duty devices like analog phones and video cameras.

4.1.2 Multiple device detection accuracy

Method. For each run, we chose $2 \leq n \leq 8$ random devices from our device set, placed them at random locations and activated them simultaneously to generate samples at different RSSI values. We then computed the average detection

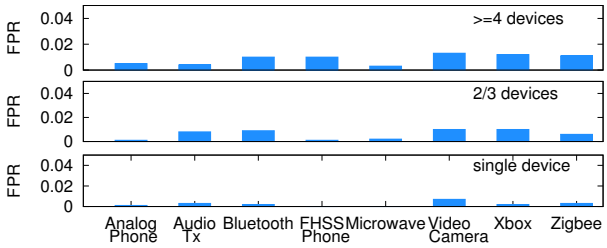

Figure 17: False positive rate for different devices.

Location/environment	Accuracy	False +ves
Indoor offices (floor-to-ceiling walls)	98.47%	0.029%
Lab environment (cubicle-style offices)	94.3%	0.067%
Apartments (dormitory-style)	96.21%	0.043%

Table 4: Airshark's performance in different environments.

accuracy at each RSSI. We repeat the experiments for different combinations of devices and locations. We note that our experiments include the "overlapping signal" cases (§3.4).

Results. Figure 16 shows the detection accuracy for 2, 3, and ≥ 4 device combinations. We observe that even when ≥ 4 devices are activated simultaneously, the average detection accuracy is more than 91% for RSSI values as low as -80 dBm. For higher RSSI values (≥ -60 dBm), the detection accuracy was 96%. For lower RSSI values, in the presence of multiple RF devices, we observed that features like spectral signatures, duty cycles do not perform well, and hence result in reduced accuracy (§4.2.1). Overall, we find that Airshark is reasonably accurate, and as we show in §4.1.6, its performance is close to that of signal analyzers [3] using custom hardware.

4.1.3 *False positives*

Method. When performing the above experiments for single and multiple device detection, we also recorded the false positives. Figure 17 shows the distribution of false positive rate across different RF devices.

Results. We observe that Airshark has a particularly low false positive rate — even when using ≥ 4 RF devices, operating under a wide range of signal strengths, the average FPR was 0.39% (maximum observed FPR was 1.3%). Further, for RSSI values ≥ -80 dBm, the average FPR was $\leq 0.068\%$.

Overall performance summary. For a total of 8 classes of RF devices used in our evaluation, across multiple runs and in presence of simultaneous activity from multiple RF devices at different signal strengths, Airshark exhibits detection accuracy of $\geq 91\%$ even for very low signal strengths of -80 dBm. The average false positive rate was 0.39%. At higher signal strengths (≥ -60 dBm) the accuracy was $\geq 96\%$.

In a typical enterprise deployment with multiple APs running Airshark, performance at low RSSI might not be a concern as we can expect at least one AP to capture the non-WiFi device signals with RSSI ≥ -80 dBm. Below, we benchmark the performance under the cases with RSSI ≥ -80 dBm. We revisit the performance at lower RSSI in §4.2.1.

4.1.4 *Location insensitivity*

Method. To understand whether the peformance of our decision tree models was affected by the location and the nature of the wireless environment, we repeated the controlled

RF device	Airshark-SVM (%) Accuracy/FPR	Airshark-DTree (%)Accuracy/FPR
Analog cordless phone	98.31% / 0.037%	97.73% / 0.012%
Bluetooth (ACL/SCO)	92.03% / 0.094%	91.63% / 0.076%
FHSS cordless phone	98.44% / 0.052%	96.47% / 0.037%
Microwave oven	94.02% / 0.012%	93.16% / 0.06%
ZigBee device	97.49% / 0.048%	96.23% / 0.036%
Video camera	94.24% / 0.08%	92.70% / 0.072%
Audio tx/headphones	92.27% / 0.016%	91.23% / 0.014%
Game controller (Xbox/Wii)	90.32% / 0.064%	91.75% / 0.046%

Table 5: Comparison of SVM and decision tree based approaches. Table shows per-device accuracy in the presence of multiple RF devices. The RSSI of the devices were ≥ -80 dBm.

Detection device	Online tests	Accuracy	False +ves
AirMaestro [3]	1827	1803 (98.7%)	NA
Airshark	1827	1761 (96.3%)	12 (0.07%)

Table 6: Comparison of Airshark and a detection device that uses a specialized hardware (AirMaestro RF signal analyzer).

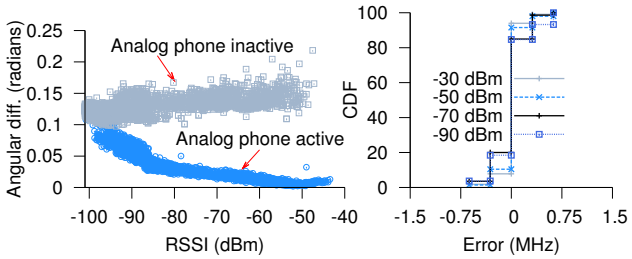

Figure 18: (a) RSSI vs. angular difference with respect to analog phone's spectral signature when the device is switched on and off (b) CDF of bandwidth estimation error at different signal strengths.

experiments in three different environments. In each case, we activated the RF devices at different signal strengths and measured Airshark's performance.

Results. Table 4 shows that Airshark performs reasonably well under all the three environments with an average detection accuracy of 94.3%–98.4% and an average FPR of 0.029%–0.067%. This shows that our decision tree models are general, and are applicable in different environments.

4.1.5 *Performance of SVM-based classifier*

Method. We compared the performance of SVM-based implementation of Airshark with the decision tree based version. Both SVM and decision tree implementations were trained using the same data. Similar to the previous experiments, we placed the RF devices at random locations to evaluate the performance at different signal strengths.

Results. Table 5 shows that the performance of SVM and decision tree for different RF devices. We observe that while SVM based implementation performs slightly better in terms of the detection accuracy (an improvement of up to 4%), the number false positives also increase. We elected to use the decision tree approach as it was much faster and has comparable performance.

4.1.6 *Comparison with specialized signal analyzers*

Method. We compared the accuracy of Airshark with the AirMaestro RF signal analyzer [3] by employing following methodology: we performed experiments by activating a

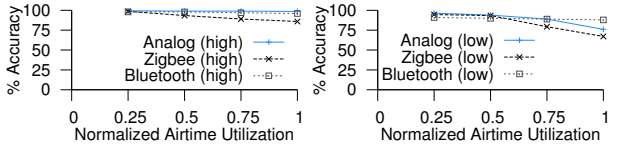

Figure 19: Stress testing Airshark with extreme WiFi interference. Detection accuracy is reduced for pulsed transmission devices (e.g., ZigBee), whereas accuracy for frequency hoppers is minimally affected.

Choice of γ_s	-105 dBm	-95 dBm	-85 dBm
Accuracy (FPR)	97.3% (4.7%)	92.13% (0.041%)	89.24% (0.023%)

Table 7: Effect of different thresholds on Airshark's performance.

combination of RF devices at different signal strengths and collected traces from both Airshark and the AirMaestro device simultaneously. Table 6 shows the results.

Results. We observe that out of 1827 device instances, AirMaestro was correctly able to detect 1803 (98.7%), whereas Airshark detected 1761 (96.3%) instances. Further, out of 66 instances where Airshark failed to identify the device, 48 instances had RSSI values of less than -80 dBm and the rest involved frequency hopping devices with multiple other RF devices operating simultaneously. We observed a total of 12 (0.07%) false positives and these instances occured when operating multiple RF devices at low signal strengths.

4.2 Microbenchmarks

Airshark's detection accuracy is affected by low signal strengths and increased WiFi interference. Below we investigate these scenarios.

4.2.1 Performance under low signal strengths

We now highlight some of the reasons for reduced accuracy at low signal strengths by examining two of the features.

— *Spectral signatures*. Consider a particular center frequency and associated bandwidth where we can expect an analog phone to operate. We wish to compute the spectral signature on this band (based on the received power in the FFT bins) and then measure the angular difference w.r.t. analog phone's spectral signature for (i) when the analog phone is active at this center frequency, (ii) when the phone is inactive. For Airshark to clearly distinguish between these two cases, there must be a clear separation between the angular differences *i.e.*, angular difference must be low when the phone is active, and higher when it is inactive. To understand the worst case performance, we also activate multiple other RF devices by placing them at random locations. Figure 18(a) shows that even in the presence of multiple devices, the angular difference is very low when the phone is operating at higher RSSI. However, when the phone is operating at lower signal strengths, the angular difference increases, thereby reducing Airshark's detection accuracy.

—*Bandwidth estimation.* In each run we activate a random RF device at a random location and let Airshark compute the bandwidth of the signal. Figure 18(b) shows the error in computed bandwidth at different RSSI values. We observe that Airshark performs very well at high RSSI values, but the bandwidth estimation error increases at low signal strengths, thereby affecting the detection accuracy.

4.2.2 Performance under extreme interference

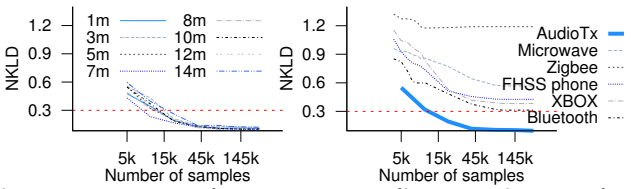

Figure 20: NKLD values wrt. to audio transmitter's pulse distribution for (a) audio transmitter at different distances (b) different RF devices

Deployment	**Proportion of RF device instances**				
	Microwave	Bluetooth	FHSS Phone	Videocam	Xbox
WLAN1	37.16%	52.34%	9.87%	–	0.6%
WLAN2	81.65%	17.43%	–	0.917%	–

Table 8: Proportion of non-WiFi RF device instances in 2 production WLANs. We collected data using Airshark for a duration of 48 hours.

We performed stress tests on Airshark by introducing additional WiFi interference traffic. We placed a WiFi transmitter close to the Airshark node (distance of 1m) and let it broadcast packets on the same channel as the fixed frequency RF devices. We changed the WiFi traffic load resulting in different airtime utilizations. We tested the detection accuracy of RF devices at HIGH and LOW signal strengths (-50 dBm and -80 dBm respectively). Figure 19 shows the effect on Airshark's detection accuracy w.r.t. normalized air time utilization (air time utilization is maximum, when the transmitter broadcasts packets at full throughput). Accuracy of high duty devices (analog phone) is affected only in the LOW case, when normalized airtime utilization is close to 1. For devices like ZigBee (fixed frequency, pulsed transmissions), the effect is more severe in the LOW case under increased airtime utilization. Frequency hopping devices like Bluetooth, however, are not affected because Airshark is able to collect enough pulses from other sub-bands. It is worth pointing out that all the previous experiments were performed in the presence of regular WiFi traffic and in different wireless environments. We therefore believe that Airshark performs reasonably well under realistic WiFi workloads.

4.2.3 Parameter tuning

We now discuss the empirically established parameters of our system. Table 7 shows the effect of using different energy thresholds. The set up for the experiments was similar to that in §4.1.2. We observe that while it is possible to improve Airshark's accuracy at lower RSSI values by lowering the threshold, this comes at the cost of increased false positives. Increasing the threshold reduces the number of peaks (and hence pulses) detected and reduces the detection accuracy.

We now show the effect of number of samples on the NKLD values. Figure 20 (left) shows how the NKLD values converge for an audio transmitter device (placed at different distances) with the total number of samples processed by Airshark. We find that around 15000 samples, the NKLD values converge to 0.3. Figure 20 (right) compares the NKLD of different RF devices when using the pulse distribution of the audio transmitter device as reference. We find that 15000 samples are sufficient, as NKLD values of ≤ 0.3 can be used to indicate the pulse distribution of the audio transmitter.

4.3 Example uses of Airshark

We now demonstrate Airshark's potential through example applications. We monitored the RF activity on a single floor of

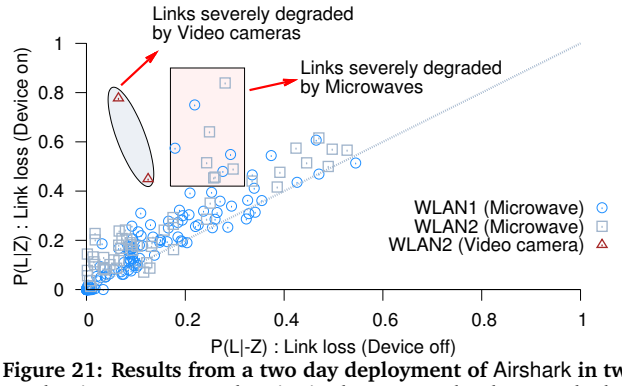

Figure 21: Results from a two day deployment of Airshark in two production WLANs. Each point in the scatter plot denotes the loss rate for a link in the absence ($p(L|\neg Z)$) and the loss rate in the presence ($p(L|Z)$) of (i) microwave ovens, (ii) video cameras, for a total of 224 links (168 links in WLAN1 and 56 links in WLAN2).

Airshark Trace Duration	Microwave	Loss rate Link L1	Airshark Trace Duration	Microwave	Loss rate Link L2
14:35:19–14:55:18	OFF	10.52%	16:50:36–16:53:19	OFF	16.66%
14:55:18–14:55:46	ON	86.20%	16:53:19–16:53:52	ON	79.61%
14:55:46–15:00:22	OFF	10.91%	16:53:52–17:13:22	OFF	15.78%
15:00:22–15:02:40	ON	45.88%	17:13:22–17:15:19	ON	48.78%
15:02:40–15:04:45	OFF	7.63%	17:15:19–17:33:20	OFF	15.00%

Table 9: Airshark traces for the time periods relevant to links L1 and L2 (WLAN1) showing increased losses due to microwave oven activity.

two production WLANs using regular wireless laptops that ran Airshark and also captured packets using *tcpdump*. WLAN1 used 7 APs and WLAN2 employed 3 APs on the monitored floor. We collected the data for 48 hours at each location. To provide confidence in our statistical estimates, we restrict our analysis to a total of 224 links (168 links in WLAN1, 56 links in WLAN2) that exchange at least 150 packets in our packet traces. Airshark can help WLAN administrators answer the following questions about RF device activity in their networks:

Question. "Which non-WiFi RF devices were visible in the WLANs? How long were the devices active, and which devices appeared more frequently?"

Analysis. Using traces from Airshark, we found that non-WiFi devices were active for 22.6% (10.48 hrs) and 13.92% (6.68 hrs) of the trace duration in WLAN1 and WLAN2 respectively. Table 8 shows the proportion of non-WiFi RF device instances in the two WLANs—microwave ovens (37.16% and 81.65%) and Bluetooth devices (52.34% and 17.43%) occurred most frequently in WLAN1 and WLAN2. FHSS cordless phones accounted for 9.87% of the instances in WLAN1. Game controllers and video cameras were also visible, albeit for very short durations.

Question. "Did any of the links in the WLAN suffer from interference due to non-WiFi devices? Which non-WiFi devices caused the most interference?"

Analysis. Airshark can help identify the interference-prone links as follows: Let L denote the event of a packet loss on a wireless link. We note that L might include losses due to non-WiFi interference as well as those due to "background losses" (e.g., due to weak signal)[3]. Let Z be the event that a non-WiFi device z is active. We compute the probability of a packet loss given the device is active, $p(L|Z)$, and the probability of a packet loss given the device is inactive, $p(L|\neg Z)$ as follows:

1. Using Airshark, we identify the periods when the device z was active (t_{on}), and when the device z was inactive (t_{off}).
2. For each link, we compute the total number of packets transmitted on the link during t_{on}, and the corresponding number of packets lost, to measure $p(L|Z)$. Similarly, we compute $p(L|\neg Z)$ by measuring the loss rate during t_{off}.
3. For the links severely interfered by device z, we can expect $p(L|Z) \gg p(L|\neg Z)$.

We make the following observations:

— *Microwave Ovens:* Figure 21 shows the impact of microwave oven activity using a scatter plot of $p(L|Z)$ and $p(L|\neg Z)$. We observe increased losses for a few links (70−80%). We found that around 20% links in WLAN1 and 10% links in WLAN2 had more than 20% increase in loss rates. Further, for around 5% of the links, the loss rates increased by more than 40%. Table 9 shows snapshots of Airshark traces and loss rates for two links L1 and L2 that experienced interference from microwave ovens.

— *Video camera:* The camera was active only for around 3 minutes in the WLAN2 trace, but it had a severe impact on two of the links as shown in Figure 21. Losses for the two links increased from 6.47% to 77.67%, and 12.47% to 44.85% during the period the camera was on.

— *Bluetooth/Xbox/FHSS phones:* In both the traces, we did not find any impact of these devices on the link loss rates.

5. RELATED WORK

There is a large body of literature on signal classification that includes work on cyclostationary signal analysis [28], blind signal detection [23], and other spectrum sensing techniques [8]. In our work, we only focus on signal detection methods that can be implemented on top of the functionality exposed by commercial WiFi cards. Present day solutions that detect RF devices include entry-level products like AirMedic [2], Wi-Spy [7] that use extra hardware to display spectrum occupancy, but cannot detect RF devices automatically. More expensive solutions like Cisco Spectrum Expert/CleanAir [4], AirMagnet Spectrum XT [2], and Bandspeed AirMaestro [3] use specialized hardware (signal analyzer ICs) to perform high resolution spectral sampling and detect RF devices. Airshark offers a similar performance, is more cost effective as it operates using commodity WiFi cards, and requires only a software upgrade to be readily integrated in existing WLAN deployments.

Recent research work [17, 21] also leverages specialized hardware to detect non-WiFi devices. Hong et. al [17] use a modified channel sounder to sample a wideband (100 MHz), and present novel cyclostationary signal analysis to accurately detect non-WiFi devices. RFDump uses GNURadio and employs timing/phase analysis along with protocol specific demodulators to detect devices. Airshark builds such functionality under the constraints of using commodity WiFi hardware.

Using controlled measurements, prior work [9,13,15,22,25] has studied the impact of non-WiFi devices on WiFi links. Many of them have focused on targeted interference scenarios, e.g.,

[3]We note that intermittent losses may also occur due to potential hidden terminals in the network. We used PIE [26], a system that can detect hidden conflicts, to discard such cases from the traces.

between Bluetooth-WiFi [13] or ZigBee-WiFi [22, 25], and proposed mechanisms for co-existence [10, 18]. Similar to [9, 15], we consider the general problem of non-WiFi interference, but specifically, we focus on the problem of making existing WiFi links better aware of non-WiFi RF devices, thereby paving the way for corrective actions that can be implemented in today's networks. Further, our mechanism is complementary to the above solutions, and can be used in conjunction to more effectively tackle non-WiFi interference.

6. CONCLUSION

In this work, we first motivated the need to detect non-WiFi RF devices by characterizing their prevalence in typical environments. We then presented Airshark, a system that can detect the non-WiFi devices using only the functionality provided by commodity WiFi cards. Airshark extracts unique features using energy samples from a WiFi card and presents a generic, and extensible framework to accurately detect multiple non-WiFi devices, while maintaining a low false positive rate. We also found its performance to be comparable to a commercial signal analyzer. Through a deployment in two production WLANs, we demonstrated Airshark's potential in understanding non-WiFi interference issues.

We envision embedding Airshark in commodity wireless APs that opens up numerous other possibilities. WLAN administrators can monitor RF device activity across the network without deploying additional hardware. APs can also employ real-time, non-WiFi interference mitigation mechanisms based on the input from Airshark (e.g., device RSSI/channel). Another application is to physically locate the RF interferer. Localization can be done by correlating the device transmissions across samples from multiple Airshark nodes and then using triangulation methods.

Acknowledgements

We thank Vishnu Katreddy for help with some of the measurements carried out in this paper. We would also like to thank the anonymous reviewers, whose comments helped bring the paper into its final form. Shravan Rayanchu, Ashish Patro, and Suman Banerjee have been supported in part by the US National Science Foundation through awards CNS-1059306, CNS-0855201, CNS-0747177, CNS-0916955, CNS-1040648, and CNS-1064944.

7. REFERENCES

[1] Agilent spectrum analyzers (signal analyzers). http://www.agilent.com/.

[2] AirMagnet AirMedic and Spectrum XT. www.airmagnet.net/products.

[3] Bandspeed AirMaestro spectrum analysis solution. http://www.bandspeed.com/.

[4] Cisco Spectrum Expert. http://www.cisco.com/en/US/products/ps9393/index.html.

[5] LIBSVM: A Library for Support Vector Machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm/.

[6] The WEKA data mining software: an update. ACM SIGKDD Explorations Newsletter.

[7] Wi-Spy spectrum analyzer. www.metageek.net.

[8] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. Next generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Comput. Netw.*, 2006.

[9] A. Baid, S. Mathur, I. Seskar, T. Singh, S. Jain, D. Raychaudhuri, S. Paul, and A. Das. Spectrum MRI: Towards diagnosis of multi-radio interference in the unlicensed band. In *IEEE WCNC 2011*.

[10] M. C.-H. Chek and Y.-K. Kwok. Design and evaluation of practical coexistence management schemes for Bluetooth and IEEE 802.11b systems. In *Computer Networks,'07*.

[11] P. Du, W. A. Kibbe, and S. M. Lin. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. In *Bioinformatics'06*.

[12] G. Palshikar. Simple algorithms for peak detection in time-series, trddc technical report'09. In *Technical Report, TRDDC*.

[13] N. Golmie, N. Chevrollier, and O. Rebala. Bluetooth and WLAN coexistence: Challenges and solutions. *IEEE Wireless Communications Magazine'03*.

[14] R. C. Gonzalez and R. E. Woods. Digital image processing, 3rd ed. 2006.

[15] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan. Understanding and mitigating the impact of RF interference on 802.11 networks. In *SIGCOMM '07*.

[16] K. Harmer, G. Howells, W. Sheng, M. Fairhurst, and F. Deravi. A peak-trough detection algorithm based on momentum. In *CISP'08*.

[17] S. Hong and S. Katti. Wispy: Knowing your RF neighborhood. In *ACM SIGCOMM 2011*.

[18] X. Jing, S. Anandaraman, M. Ergin, I. Seskar, and D. Raychaudhuri. Distributed coordination schemes for multi-radio co-existence in dense spectrum environments. In *IEEE DySpan '08*.

[19] A. Kamerman and N. Erkogevi. Microwave oven interference on wireless Lans operating in the 2.4 GHz ISM band. In *PIMRC'97*.

[20] Kullback, S. Information theory and statistics, 1959.

[21] K. Lakshminarayanan, S. Sapra, S. Seshan, and P. Steenkiste. RFDump: an architecture for monitoring the wireless ether. In *CoNext'09*.

[22] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving Wi-Fi interference in low power zigbee networks. In *ACM SenSys'10*.

[23] O. Zakaria. Blind signal detection and identification over the 2.4 GHz ISM band for cognitive radio. In *MS Thesis USF'09*.

[24] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., 1993.

[25] S. Y. Shin, J. S. Kang, and H. S. Park. Packet error rate analysis of zigbee under wlan and bluetooth interferences. In *IEEE Trans. Wireless Comm.'06*.

[26] V. Shrivastava, S. Rayanchu, S. Banerjee, and K. Papagiannaki. PIE in the Sky: online passive interference estimation for enterprise WLANs. In *NSDI'11*.

[27] T. M. Taher, A. Z. Al-Banna, D. R. Ucci, and J. L. LoCicero. Characterization of an unintentional Wi-Fi interference device-the residential microwave oven. In *Milcom'06*.

[28] W. Gardner. Exploitation of spectral redundancy in cyclostationary signals. In *IEEE Signal Processing Magazine'91*.