Using Your Smartphone to Detect and Map Heterogeneous Networks and Devices in the Home

George Nychis
Adaptrum
george@adaptrum.com

Srinivasan Seshan
Carnegie Mellon University
srini@cs.cmu.edu

Peter Steenkiste
Carnegie Mellon University
prs@cs.cmu.edu

ABSTRACT

Heterogeneity in the wireless spectrum is an increasing problem which breaks down coordination and exacerbates wireless interference. In particular, this is a growing problem in the home: heterogeneity is increasing, yet there is a lack of tools and expertise in the home to gather the necessary information about its RF environment to combat this issue.

In this paper, we present a unique monitoring system design which leverages *the smartphone*: now commodity, flexible and familiar to the home user, and equipped with multiple heterogeneous for sensing. Our design overcomes challenges inherent to monitoring with the phone (e.g., it is location agnostic without in-door localization) to derive where signals go and what they interfere with. It does so only requiring simple user-interaction, and as we will show, can bring the information to a level the user can understand.

1. INTRODUCTION

As we find more applications for wireless communication, we are turning to a diverse set of technologies to meet the specific needs and constraints of each application. This has resulted in a significant increase of heterogeneity in the unlicensed bands, and unfortunately, this diversity (in PHY & MAC) often exacerbates interference, reducing network performance and capacity. For example, cordless phones can decrease 802.11's performance by 90% [5, 9], while 802.11 causes the same degradation for ZigBee networks [8].

This is *especially* a concern in the home where heterogeneity and density are increasing. Cordless phones, gaming controllers, baby monitors, wireless speaker systems, and "Smarthome" devices make interference in the home unique and challenging. Proposed solutions to this problem attempt to isolate incompatible technologies (i.e., spectrum management), modify protocols to reduce interference when sharing a channel [5,8], and adapt their spectrum usage (e.g., subcarrier suppression) and transmission power over time. To apply such solutions, it is critical to know where signals go and what they interfere with (i.e., strength at various locations).

Unfortunately, it is nearly impossible to gather this information in the home due to lack of equipment and expertise. Additionally, bringing the information to a level the home user can understand to address issues is a challenge little work has attempted to address. The average home user has no knowledge of dBm, differences in technologies, or even MAC address (i.e., what devices have what address).

Motivated by these concerns, the goal of our work is two-fold: 1) To make it easy to collect (and update) an accurate view of the home's heterogeneous RF environment (i.e., where signals go and what they interfere with), and 2) To bring the level of information collected by the monitor up to a level the user can understand to enable better diagnostics, spectrum management, and coexistence in the home.

Achieving these goals, however, is non-trivial. First, we have relied on dense monitoring infrastructures (JigSaw [3], DAIR [2], WifiNet [10]) to create views of entire environments which, unfortunately, are too costly and complex for the home. Second, current heterogeneous monitors (RFDump [7], Airshark [9], DOF [6]) only able to state that a signal is present at a certain strength (e.g., a Bluetooth Signal @ -52dBm). This is too low-level for the home user, and such systems are unable to derive more complex (but necessary) properties (e.g., two heterogeneous signals coordinate since generated by co-located radios on the same device).

In this paper, we propose exploring the use of the *smartphone*: a single, location agnostic, and power-constrained sensor to create a system that can overcome these challenges. To the best of our knowledge, the design and implementation of a wireless monitoring system based on a single phone (unpredictable in movement and operated without expertise) is an open question with various possible designs that trade-off usability, complexity, power efficiency, and effectiveness.

We present an initial design that does *not* rely on in-door localization, brings information to a level the user can understand by creating *device abstractions* (e.g., "Bob's iPad" that generates specific Bluetooth / Wi-Fi signals), and requires little user information/effort to be *highly usable*. With a short training phase, we obtain baseline information about the environment and then leverage the user's natural movement to map where signals go in the environment; taking measurement when the phone has periodic "close-encounters" near their devices. Although exact measurement locations are unknown, we can still derive what is most important: where signals go and what they interfere with.

To illustrate the power of this information and our system, we build a prototype on Android with a sample application that leverages information collected and *force-directed graphs* to draw an easy-to-understand environmental map of the home by which diagnostic information can be overlaid. Evaluation in a heterogeneous testbed shows initial design to be accurate within 5*dB*.

HotWireless'14, September 11, 2014, Maui, Hawaii, USA.

http://dx.doi.org/10.1145/2643614.2643624 .

2. RELATED WORK

Early spatially-aware monitoring systems such as JigSaw [3] and DAIR [2] enable the monitoring of enterprise environments through a dense deployment of sensors. By doing so, every event (i.e., a transmission) can be observed by at least one sensor (an 802.11 packet-level radio). By co-locating the sensors with APs or placing them in specific office locations, events are localized to the static and known location of the sensor that received it strongest. By localizing, collecting, and intelligently synchronizing the events at a central location, one can obtain a global and spatial view. From this, coverage and interference ranges can be inferred to generate a conflict graph and plan the environment.

To address the rise of heterogeneity, recent work has focused on developing a heterogeneous sensor. RFDump [7] and DOF [6] turned to frequency agile radios with low-level signal access. RFDump analyzes a signal's power, timing, phase, and frequency usage to classify it. DOF improved accuracy through cyclostationary signal analysis. While powerful and feasible, neither pieces of work were immediately implementable on commodity hardware, making them impractical for deployment.

More recently, AirShark [9] has shown that low-level signal information could be extracted from newer 802.11 radios to perform similar signal classification techniques as RFDump. Now, a commodity radio can detect microwaves, cordless phones, gaming controllers, ZigBee networks, and others. From this, the authors built WifiNet [10]: a dense monitoring infrastructure for the enterprise, which is able to detect, localize, and estimate the impact of heterogeneous interference on the Wifi network.

3. TOWARDS A PRACTICAL HOME WIRELESS MONITORING SYSTEM

Requirements of a Home Monitor: We believe there are 4 key requirements to a home monitor. First, it must be heterogeneous in its monitoring capabilities. Second, it must not be WiFi-centric: there are many multi-radio and heterogeneous devices in the home with a lack of priority in terms of performance among technologies (unlike the enterprise). In the home, one must be able to differentiate signals as internal (prioritized & configurable) from external.

Third, the system must be comprehensive in characterizing the environment spatially (where signals go) and temporally (when devices are commonly used). Finally, it must be *user-friendly*: low in cost, simple interactions (e.g., when requesting input or tasks from the user), in addition to information being presented back to the user at a level they can understand. Device abstractions are critical to usability: keeping the level of information at a level the user can understand, rather than signal-level information.

The Potential of a Phone-based Monitor: The smartphone is now commodity, it has a very familiar and flexible interface to make usability possible, and it is equipped with multiple heterogeneous radios for low-power sensing with packet-level information to passively decode, or proactively query, devices for rich and human readable information.

Even though the phone would be considered a single sensor, its ubiquitous movement could likely gain more accurate information about the environment than by collecting information from various fixed points that happen to be equipped with a monitor. As anecdotal evidence in Figure 1, we highlight a path that could be considered typical movement in a home. Walking in to the home, phone in pocket, the phone gains several opportunities to take measurements near devices to learn where signals go (i.e., points 1,3,4,5,9,10).

To further validate this claim, we conducted a (accelerometer based) study involving over 100 users. On average, users' phones

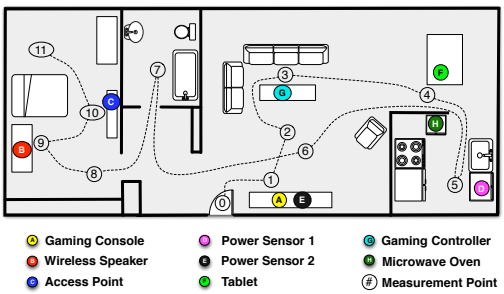

Figure 1: Our heterogeneous home environment and testbed.

were home at least 5 to 6 hours a day (not including time when their owners were asleep). Half of the phones were mobile more than 20% of this time and one fifth for more than 50% of this time.

Challenges in a Phone-based Monitor: A first-order concern is whether the phone can provide accurate signal measurements, i.e., given its orientation and levels of obstruction (e.g., being attenuated in a pocket). In addition, the location of the phone at any point in time within the home is generally unknown (*assuming no in-door localization*). Temporal dynamics (e.g., how frequently devices are used), typically learned through continuous monitoring, must instead be learned through periodic monitoring on top of a monitor whose availability in the environment (i.e., home) is unpredictable. Smartphone power usage is a first-order concern.

4. SYSTEM DESIGN

Designing a smartphone-based home monitor to achieve the requirements we have identified (§3) is an open question that, to the best of our knowledge, our work is the first to address. In fact, there are various possible designs, many of which we have considered, that typically trade-off usability, complexity, power efficiency, and effectiveness. The majority of designs can be decomposed to the level of user involvement (tasks/input), and the system's degree of proactivity.

User Involvement: One could imagine requiring the user to walk around the home, trigger measurements, and manually label the physical location of each measurement. However, minimizing information and tasks required from the user make it more usable. The trade-off is that each piece of information needed to properly monitor and manage the environment that is not provided increases system complexity by requiring our power-constrained monitor to derive it (e.g., regarding *spatial diversity:* has a device moved?).

Degree of Proactivity: One could design an entirely reactive system that would only monitor when notified by the user of a problem which would be power efficient, but likely to require more time and user involvement to resolve issues (requiring the user to take the phone to various locations). A highly proactive monitor would continuously monitor and learn the environment to prevent / diagnose issues faster, but requires careful design to be power efficient.

4.1 Our Proposed Design & Vision

Any design would be hard to argue as ideal when addressing the requirements of the system. However, we present a 3 phase approach that we believe is a balanced and practical.

Phase 1 – Training: First, the user trains the system of their home location for power efficiency reasons. Our system uses coarse location information and disables itself when not in the home. Next, the user is asked to turn on all devices in their home. Using the phone's heterogeneous radios and techniques to derive device abstractions, we scan for devices in the area and ask the user to select

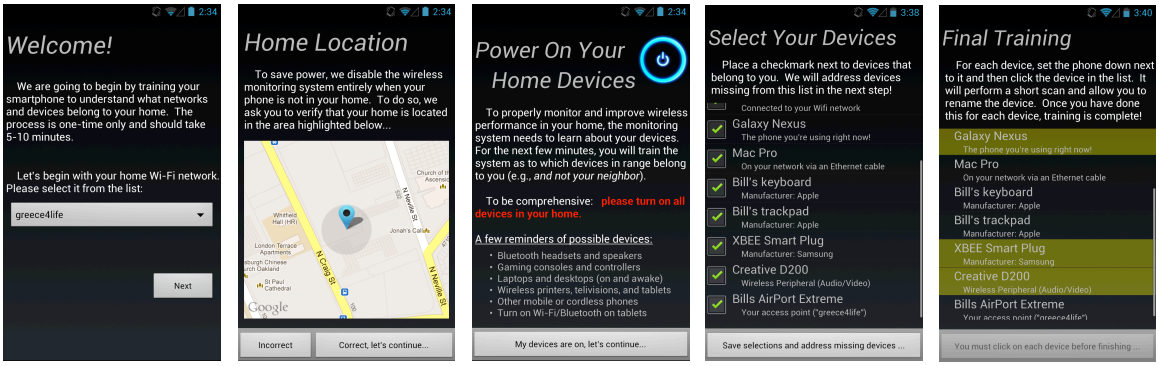

Figure 2: Screenshots of our system's interface, highlighting its simplicity, as well as usability through user recognizable identifiers.

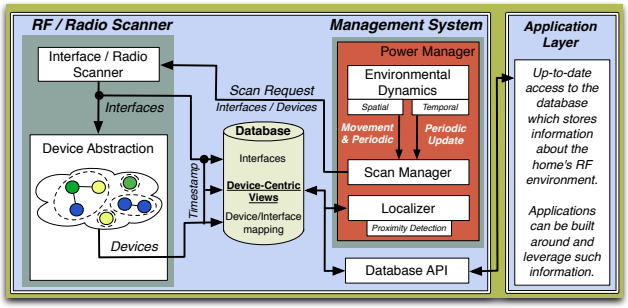

Figure 3: Overview of our system design and its components.

which devices are theirs. This performs signal differentiation (key to prioritizing/configuring internal networks) *at the device level*.

Finally, we ask the user to take the phone and set it down next to each device. By doing so, we are able to: 1) Derive initial spatial dynamics and a conflict graph, and 2) Derive signal strength thresholds indicative of being 1-2ft from the device, used by our system to opportunistically update measurements over time.

Phase 2 – Monitoring: At this point, our system performs background monitoring tasks whose details and operations are largely transparent to the home user. In this phase, the system (*not user*) tasks are: 1) Ensuring an up-to-date and complete list of internal networks by periodically scanning for new devices we believe may be the users, 2) Opportunistically updating where signals go as the user walks near devices, and 3) Learning of spatial and temporal dynamics (e.g., when devices move / how active devices are).

Phase 3 – Diagnostics & Management: The information collected in the first two phases is useful and important to various applications. This information can be used to perform diagnostics, implement coexistence techniques, or even draw an easy-to-understand environmental map with overlaid connectivity information.

5. SYSTEM DESIGN & COMPONENTS

An overview of our proposed system is shown in Figure 3. The high level design includes a *RF / radio scanner* and a *management system*. The management system deals with the strategy of when to take measurements in the environment to comprehensively map the environment, yet to be power efficient. The RF / radio scanner is responsible for low-level scans and using heuristics to create device abstractions. These two components access and update the database.

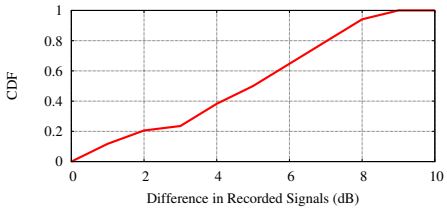

Figure 4: The phone can be calibrated to report signals with an accuracy of ± 5 dB.

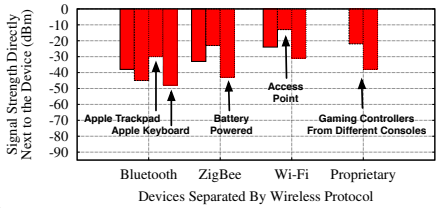

Figure 5: No single threshold will work to localize measurements to a device's relative location. It must be learned to detect proximity.

5.1 System Components

Device-Centric Views

As discussed, our goal is to take opportunistic measurements when near a device. Although we do not know true physical location, our goal is to detect proximity, measure, and *anchor* the measurement to the relative location of the internal device. A history of measurements of these measurements are then stored in the database as being at the device's relative location.

There are two challenges in this, however. The first is illustrated in Figure 5. We placed the phone directly next to several devices in the home and record the signal strength of the device observed. As shown, signal strengths when near a device are truly device dependent, even within a manufacturer. Therefore, signal strengths indicative of being near a device needs to be learned per device. As part of our system design, when the user trains the system (i.e., phase 1), the user is asked to set the phone next to each device to empirically learn proximity thresholds for each device. To detect proximity, trigger a measurement, and anchor it we monitor the accelerometer on the phone to detect movement and then continuously monitor signal strengths of internal devices to be within 2dB of the strength learned in training.

The second challenge is that the physical location of a device can change, leading to device-centric views being anchored at two different relative locations. To overcome this, our system employs two techniques. First, during the training phase when the user sets their phone down next to each device we ask them if it is mobile: "Is this device always in this location?" If not, we only keep a

measurement history of depth 1 for the device: its most current measurement is the only one we can trust. On the other hand, fixed devices can still move (e.g., the Xbox changing rooms). In §5.1, we show how to detect the movement of fixed devices and invalidate device-centric views in the database once moved.

Creating Device Abstractions

We leverage the phone's packet capabilities to extract user recognizable identifiers. *Passive packet inspection* is used to extract network names, e.g., "The Smith's Wifi" included in broadcast traffic such as beacons. *Service discovery protocols* (SDPs) such as Bonjour, UPnP, and SSDP provide user specified names (e.g., "Jack's PC"), hardware specifics (e.g., "4th Gen MacBook Air"), and OS specifics (e.g., "Ubuntu"). Bluetooth devices respond with a "Major Service Class" (e.g., audio), as well as a "Major Device Class" (e.g., phone, speaker, toy). ZigBee devices also respond with available services.

Additionally, the smartphone queries *IEEE's OUI* database [1] which maps the first 24-bits of a MAC address (Wifi, ZigBee, Bluetooth, and others) to the organization who assigned the address. This returns names such as: Dell Computer, Microsoft Corp., and Logitech, helping a user identify a device.

Finally, we leverage hints across the PHY, MAC, and network layers that radios/interfaces belong to the same device. Sample heuristics look at relationship between MAC addresses, e.g., it is now common that manufacturers assign adjacent MAC addresses to interfaces that belong to the same device. Additionally, a heuristic looks at user-recognizable identifier names. Bluetooth and Wi-Fi interfaces respond with the same name ("Bill's MacBook Air"). Using these heuristics that we are developing, we can group radios/signals to the same device. Then, we can present such devices to users to ask which are theirs (i.e., differentiation).

Environmental Dynamics

Detecting spatial changes are critical to: 1) Notify higher layer diagnostics which may need to account for the change, and 2) Invalidate device-centric views that were "anchored" to an old relative location.

We consider differentiating fixed and mobile devices as important, since the movement of fixed (as opposed to mobile) devices is a more critical change. The reason for this is that spectrum management algorithms can prioritize spectrum configuration based on fixed devices that are frequently used, whose interference is stable. In contrast, mobile device interference is harder to account/configure for. We believe the average consumer is less tolerable of their fixed devices performing improperly (e.g., their AP or Xbox).

To detect fixed device movement, *we track the variations of signals between fixed devices in the environment, from other fixed devices*. Systematically, we use the history of (valid) device-centric views at each fixed device and compute a *variation score* of their signal strengths across their history. If the variation exceeds a threshold, we can determine that the device has moved.

Temporal dynamics are largely spatially independent, meaning that as long as the phone is in the home, it can observe temporal network and device usage. For this reason, we can periodically monitor for the usage of devices (once every 30 minutes in our design), and measure their airtime. This usage is kept as a history within our system's database.

Energy Efficiency Manager

We make several key design decisions to improve the energy efficiency of our monitor. Importantly, we *only* enable monitoring functionality when the phone is in the home. During the training phase (§4.1), we record the phone's coordinates and use them to enable/disable monitoring. Next, we also only enable monitoring

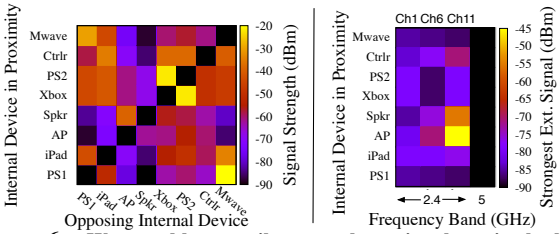

Figure 6: We are able to easily map where signals go in the home, providing insight in to signal strengths between devices.

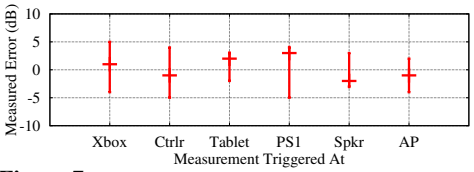

Figure 7: Device-centric views are within ±5dB error.

when the phone is not in use. This is to avoid draining the battery when the phone is in use, and disrupting connectivity.

For energy efficiency in capturing spatial dynamics, we make the system cognizant of the fact that when the phone is not moving, spatial dynamics are likely the same. Therefore, we only enable proximity detection and device-centric views when the phone begins moving. Additionally, we make the system aware of the fact that some radios are coupled to the same hardware and behavior e.g., observing the Xbox's diurnal usage is sufficient to learn of its controller's usage.

6. PROTOTYPE & EVALUTION

We build a full prototype on Android OS (Galaxy Nexus hardware) and evaluate its accuracy in a controlled testbed. Our prototype integrates a ZigBee radio, and Airshark-like functionality using a WiSpy device. Airshark was not publicly available at our submission.

Accuracy of the Phone-based Monitor

First, we evaluate our system's ability to collect meaningful and useful information in the training phase. In Figure 6, we show a heatmap of the signals at each internal device, from every other device which is collected, stored in our database. It is meaningful and useful: e.g., *Power Sensor 1 (PS1)* is hidden to the *AP*, but not visaversa; *PS1* is also in strong interference range of the microwave.

After training our system, we walk through the environment along the path highlighted in Figure 1 with the prototype in our pocket. Through this simple movement with the phone, a set of measurements should be triggered which "update" where signals go and what they interfere with, which (since we did not physically move anything) should match the signals observed in the training phase. Through this movement (which we repeat several times) and the proximity thresholds we derived in the training phase (§5.1), we find that our system consistently triggers measurements when nearby each device to update the device-centric views.

We verify the accuracy of these device-centric views to match our expectation given the phone's orientation, body attenuation, and movement when taking these close-encounter based measurements. For each internal device, we calculate the max, average, and minimum observed error between the device-centric view captured in training, compared to the device-centric view updated when walking. We present the results in Figure 7, shown to be bounded within $\pm 5dB$.

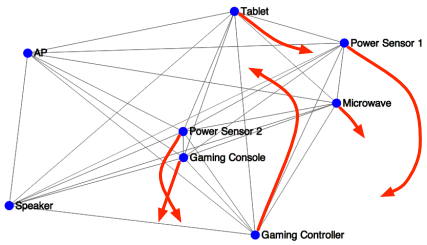

Figure 8: Many devices end up out of place when using basic and agnostic spring forces.

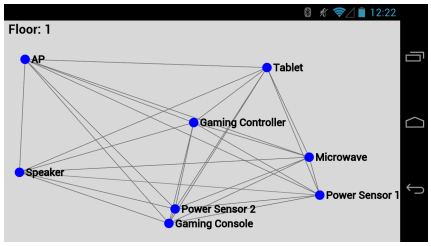

Figure 9: Our force-directed model and spring forces reflect true layout.

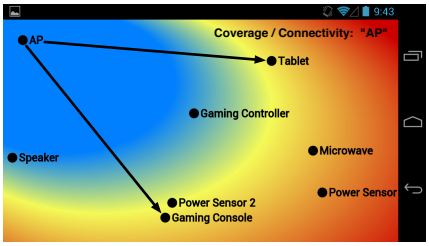

Figure 10: Overlays such as this communication/coverage overlay can be easily understood.

Creating a User-friendly Map

While the information is the heatmap is useful to the average wireless network administrator, the average home does not understand it. Therefore, we use the information in the heatmap with our device abstractions and *force-directed graphing* to explore a more user-friendly map. Orthogonal to a large amount of work which has attempted to "map" layouts based on relative data (e.g., mapping the Internet using RTT times [4]), our goal is to generate a layout of devices in the home using their relative signal strengths to each other.

The key to leveraging the force-directed graph is determining appropriate forces. Using the raw data from the heatmap is insufficient since devices transmit at different powers. This must be taken in to consideration, in addition to the frequency of operation. If this is not taken in to consideration, error in where devices are placed on the map is high. We show this in Figure 8 for our heterogeneous testbed. Devices are placed in ways that do not reflect reality; the red arrows pointing to where they should be.

Fortunately, when the user places their phone near each device during the training phase we have a baseline of how strong the transmitter is to account for differences in transmission powers. When taking these factors in to consideration, we have the possibility of generating a physical map that more accurately reflects true placement (Figure 9).

With this map, we can potentially generate interfaces that the average home user can interact with. For example, the one we illustrate in Figure 10 that places all of the devices on the map and the user can select on a transmitter and see its coverage range. They can then easily manage their home network and see how placement affects their network performance. This overlays the heatmap information on to true layout with user-recognizable identifiers (e.g., "Gaming Controller").

7. LIMITATIONS AND DISCUSSION

Multiple-phone Design: In our work, we presented the design of a system that was based on a single smartphone collecting information about the home's heterogeneous RF environment. The information collected is therefore based on the interactions of this single phone in the environment (i.e., where that phone goes, and particularly what it comes in close contact with). There are, however, *many* potential smartphones in a single home environment and, in particular, those phones go to different locations of the home and may interact more closely with different devices. This opens up the possibility of a multi-phone design that introduces new challenges, and potentially more rich information. Now, multiple monitors must be able to coordinate and share information.

User Involvement vs. Complexity: Although we presented a design that requires a rather significant amount of user involvement in the training phase, many will argue that designs are possible that require absolutely no user involvement. This would be equivalent to a service running on the phone that collects information about the environment over time without any user involvement. While seemingly challenging, such a design is likely possible and considered future work. For example, how does the monitoring system learn which wireless devices belong to the home user? It may be possible that the training phase instead takes days instead of minutes, monitoring which devices it comes in close contact with multiple times, and assumes that these devices belong to the user. There may also be ways to diagnose and reconfigure the environment without involving the user, also.

Other Spectrum Bands: Given our use of the smartphone as the base of our monitoring system, we are limited to only being able to monitor certain spectrum bands. While this may change over time, there are still a few spectrum bands that wireless devices use in the home that may not be supported by smartphones like the 900 MHz, 60 GHz, and white space spectrum bands.

8. CONCLUSIONS

We presented the design of a practical and usable home monitor, based on the smartphone. It is able derive where signals go in the home (i.e., their strength at various locations) with little user involvement, and avoids the cost and complexity of multi-sensor deployments. A key to our work is trying to bring the information up to a level the typical home user can understand. Additionally, the system collects information that can be used to implement various applications.

Project & Code Download: The system we have developed has been made public for download and additional research on Github: https://github.com/gnychis/android-wmon. Our prototype can be used to study home environments, as well as explore other smartphone-based designs.

9. REFERENCES

[1] IEEE Registration Authority OUI Public Listing, http://standards.ieee.org/develop/regauth/oui/public.html.
[2] P. Bahl et al. Enhancing the Security of Corporate Wi-Fi Networks Using DAIR. MobiSys 2006.
[3] Y. chung Cheng et al. Jigsaw: Solving the puzzle of enterprise 802.11 analysis. In *SIGCOMM*, 2006.
[4] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *Proceedings of the ACM SIGCOMM '04 Conference*, Portland, Oregon, August 2004.
[5] S. Gollakota et al. Clearing the RF smog: making 802.11n robust to cross-technology interference. In *SIGCOMM 2011*.
[6] S. S. Hong and S. R. Katti. Dof: a local wireless information plane. In *SIGCOMM 2011*.
[7] K. Lakshminarayanan et al. RFDump: An Architecture for Monitoring the Wireless Ether. In *CoNEXT '09*.
[8] C.-J. M. Liang, N. B. Priyantha, J. Liu, and A. Terzis. Surviving wi-fi interference in low power zigbee networks. In *SenSys '10*.
[9] S. Rayanchu et al. Airshark: detecting non-WiFi RF devices using commodity WiFi hardware. IMC 2011.
[10] S. Rayanchu et al. Catching whales and minnows using wifinet: Deconstructing non-wifi interference using wifi hardware. NSDI, 2012.