Wi-FM: Resolving Neighborhood Wireless Network Affairs by Listening to Music

Marcel Flores, Uri Klarman, and Aleksandar Kuzmanovic
Electrical Engineering and Computer Science Department
Northwestern University
{marcel-flores uri.klarman}@u.northwestern.edu akuzma@northwestern.edu

***Abstract*—FM radio, typically broadcast in the 87.5 to 108.0 Mhz range, is widely available in urban areas and beyond. Contrary to GPS, it effectively penetrates buildings; contrary to 3G/4G or TV, FM radio receivers are becoming freely available in mobile devices. Indeed, nearly every smart phone and many other consumer electronics today have a built-in FM chip. In this paper, we demonstrate that this ubiquitous in-the-air and on-device FM radio availability presents a unique opportunity to address some of the fundamental wireless networking problems. In particular, we focus on the problem commonly arising in home networks where devices from neighboring, yet autonomous and non-collaborative, Wi-Fi networks systematically "step on each other's feet", *i.e.*, interfere and degrade each other's performance. We show that the digital signal that accompanies broadcast FM radio has sufficient structure to enable effective scheduling relative to it. It thus provides a common reference for neighboring devices to harmonize their transmissions, yet without requiring any explicit communication among them. To the best of our knowledge, our system is the first to enable such mutually-beneficial, autonomous, and implicit harmonization among Wi-Fi devices *across* administrative network bounds.**

I. INTRODUCTION

Our homes have become complex networking environments with numerous devices utilizing the wireless spectrum in countless ways. In addition to the streaming and gaming applications which have become common-place, today's homes are increasingly becoming filled with a growing number of devices such as cameras, health and motion sensors, thermostats, *etc* [1]. Unless a home is located in the "middle of nowhere," it is more than likely that neighboring home Wi-Fi networks will "step on each other's feet", *i.e.*, associated network flows from different networks will interfere with each other, heavily affecting performance.

Indeed, a recent large-scale measurement study has shown that more than 75% of homes have an overlapping Wi-Fi neighbor [2]. Unfortunately, this number represents only a *lower bound*, since dispersed wireless home-network devices could be exposed to neighboring networks which are immeasurable from a single location. Another in-depth study on home wireless experience was able to attribute substantial periods of poor performance to neighboring Wi-Fi transmitters [3]. In addition to the dense private Wi-Fi deployments, which are key to such problems, the study showed that the majority of networks use a single Wi-Fi channel. This indicates that the majority of home APs use a static Wi-Fi configuration, and are never re-assigned by residents after they are deployed [3].

An enormous research effort has been invested to improve the performance of wireless networks using various methods, *e.g.*, via coordination and scheduling, *e.g.*, [4], [5], [6], [7], [8], [9], resource sharing, *e.g.*, [10], [11], [12], [13], [14], [15], [16], [17], deploying MIMO algorithms, *e.g.*, [18], [19], *etc*. Common to these efforts is that they address the problems in the context of *managed*, often also called *enterprise*, wireless networks. In such networks, all the APs are managed by a single authority, which certainly enables the deployment of collaborative protocols at different APs. While these methods are invaluable in the managed network scenarios that they were designed for, they are inapplicable in many home network environments where each AP is individually managed by its owner.

In this paper, we propose to address the above problems that arise in unmanaged wireless networks via *FM radio*. Nearly every smart phone and various other mobile devices manufactured today contain an FM chip [20]. It is typically implemented together with Wi-Fi and Bluetooth on a single chip [21], [22], making it easier for manufacturers to comprehensively integrate essential functionality [23]. The key driving force behind the wide deployment of FM receivers is not just users' demand for broadcast radio in their devices [24], but rather a significant applicability in times of emergency: without power, land-line communication, mobile phone communication, or cable television, the only functioning source of information is "over the air" broadcasting [23]. This enables users to receive FM radio broadcasts in a user's local area through a built-in FM tuner inside the mobile device *e.g.*, [25], [26]. Beyond smart phones, the list of FM-enabled devices is growing [27], and coordinated efforts are underway to activate the FM functionality by the remaining carriers and in other portable electronics, *e.g.*, [28], [24], [29], [30].

This ubiquitous in-the-air and on-device FM availability provides an unprecedented opportunity to address some of the fundamental wireless home-network issues. Unlike GPS, FM signal successfully penetrates buildings. Unlike cellular 3G/4G or TV, FM radio receiver hardware is becoming freely available on mobile devices. We utilize this broadcast FM signal, in particular a digital signal that accompanies broadcast FM radio, the Radio Data System (RDS), as a medium to effectively harmonize neighboring Wi-Fi devices *relative* to this baseline RDS signal. It is exactly this feature of our system, *i.e.*, *RDS-relative neighborhood harmonization*, which requires *no* explicit communication among participating devices, that enables us to resolve scenarios when nodes, both STAs and APs, from unmanaged networks systematically "step on each other's feet."

We present Wi-FM, a system that utilizes existing ambient FM signal to perform neighborhood harmonization. Wi-FM makes us of the fact that a broadcast radio signal reaches all devices at a location at virtually the same time [31], which helps synchronize all nodes relative to this signal. Unlike many synchronization methods, Wi-FM generates *no* in-band traffic while synchronizing with the ambient signal: neither in-band beacons nor additional node-to-node synchronization messages. Rather, it relies on identifying *time landmarks* in the form of repeated patterns in the structure of the underlying digital RDS signal. Because all participating nodes do this independently, they do not partake in any explicit communication among themselves. This feature enables Wi-FM to effectively harmonize Wi-Fi nodes transmitting in the same channel, even when they belong to independently-managed autonomous wireless networks.

Once synchronized to the RDS digital signal, the nodes can effectively sense the environment and infer the other nodes' scheduling choices over short time-scales. With this information, Wi-FM provides a methodology for arriving senders to determine how best to approach a given network state. It differentiates between light and heavy traffic scenarios, providing mechanisms for the full use of the network in quiet cases, sharing of the network in the case of both light and heavy traffic via *negative scheduling*, and more complex conditions for cases when many heavy-traffic devices must coexist in the network via a *neighborhood fairness* algorithm.

Our experiments show the following: (i) It is possible to effectively detect and select a common FM station in a neighborhood and utilize them for Wi-FM. (ii) The RDS digital signal accompanying FM radio has a sufficient structure to enable RDS-relative harmonization. (iii) This signal is highly resilient to data loss, *i.e.*, it retains device harmonization even in the presence of substantial reduction of matching bits. (iv) Our software-defined radio implementation provides more than sufficient synchronization accuracy to allow for scheduling based on the ambient FM signals. (vi) Our experiments on realistic home-networking setups show scenarios where throughput improves by up to 50%, and by 35% on average. The key to these improvements lies in Wi-FM's ability to systematically reduce contention via network harmonization.

Our key contributions are the following:

- We introduce Wi-FM, the first system to utilize the FM broadcast radio signal, *i.e.*, the associated digital RDS signal, as a vehicle to boost the performance of Wi-Fi networks.
- We present the first system to enable Wi-Fi device harmonization *across* administrative network bounds in unmanaged network scenarios.

We achieved all the above using off-the-shelf network equipment and in software via a Linux based implementation and evaluation. Undoubtedly, hardware-level FM radio signal processing implementations could open the doors to the deployment of advanced wireless networking algorithms and applications, beyond neighborhood harmonization.

II. BACKGROUND

Here, we provide the necessary background on the digital signal that accompanies the broadcast FM radio. Then, we explore its repetitive symbol structure and demonstrate how it can be utilized as a common medium to harmonize neighboring Wi-Fi devices.

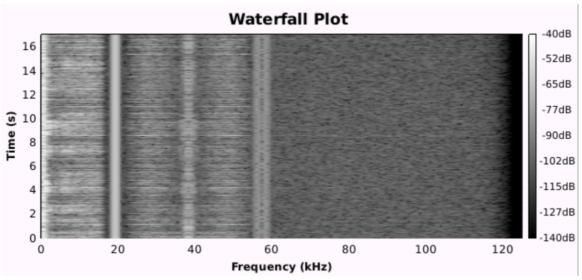

Fig. 1. A waterfall plot of FM radio signal. The RDS digital signal can be seen as two lines at the 57-kHz subcarrier.

A. The Radio Data System

The Radio Data System (RDS), also called the Radio Data Broadcast System (RDBS), is a digital signal that accompanies broadcast FM radio [32]. Intended as a method for improving usability of radios, these digital signals convey station information, current program details, traffic alerts, and other information. Most of this information is then presented to listeners directly through their radio's interface. Indeed, many radio listeners may be familiar with RDS as the mechanism that communicates the data needed to print the song title and artist on their car stereo.

To avoid interfering with legacy radio devices, RDS data is broadcast at the third harmonic of the 19-kHz pilot-tone, as seen in Figure 1. Specifically, the signal can be found at the 57kHz±6Hz subcarrier frequency, which is modulated with a form of two-phase PSK. The clock frequency of the arriving data is given by dividing the transmitted subcarrier frequency by 48, which provides a data-rate of 1187.5 bps. The transmitted bits are further subject to differential coding in order to remove ambiguity in the signal [32].

B. Baseband Coding

The RDS system provides a baseband coding in order to communicate data to the receivers. Without loss of generality, we restrict ourselves to the basic format in order to allow understanding of our overlay scheme. The RDS system has a complex set of messages and communication modes that allow it to communicate a rich set of data in a careful fashion. Each mode provides a set of repeatable patterns that could be exploited for our scheme.

Figure 2 shows the layout of the coding. The first layer of the coding is known as a *group*, which consists of 104 bits. Each group is broken down into four 26 bit *blocks*: Block A (bits 0-25), B (26-51), C (52-77), and D (78-103). Each block consists of 16 bits of data, and 10 bits of checkword to allow for error detection and some correction. The checkword is encoded with an offset, which indicates the current block.

To provide a basis for scheduling, we consider a few additional components of the RDS structure. First, each broadcasting station has a unique code known as a *Program Identifier* (PI) code. This data is included in the first 16 bits of Block

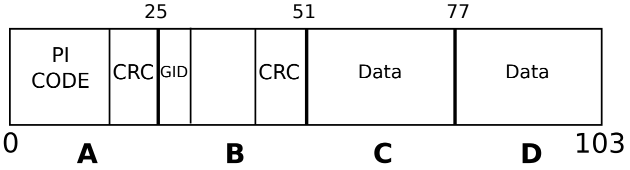

Fig. 2. The base encoding layout of the RDS scheme. The 104 bits repeat and the value of data depends on the current group context.

A. Second, the first 4 bits of Block B provide a *group code*, which indicates what type of data is contained in the remaining blocks in the group. While there are no hard requirements on the distribution of group codes in the stream, the RDS specification provides typical use guidelines, further providing for reliable repetition of group codes over time.

The specification further provides 13 group codes and 2 formats for each code, resulting in 26 possible message formats. The internal structure of each of these groups allows for additional repeated structure. For example, a group 2 message specifies a small amount of display text, up to 64 characters. Specifically, the final 4 data bits of Group B are used to specify a text segment (*i.e.*, an offset), and Groups C and D are used to specify 4 characters of data. The bottom line is that there exist numerous *time landmarks* in the underlying RDS signal. Such landmarks could be extracted and utilized for harmonization. We explain our approach next.

III. Wi-FM ARCHITECTURE

We now present the design of Wi-FM, and the details of how we use ambient FM radio signal to enable Wi-Fi device harmonization across administrative network domains. Specifically we explore how the RDS signal described in the previous section can be used to achieve this purpose.

A. Synchronizing to the Ambient RDS Signal

The first component of our harmonization scheme is our treatment of time. As digital signal, the smallest increment available from RDS is a single bit. Since RDS features a data rate of 1187.5 bps, or approximately $842\mu s$ per bit, we treat this as our basic time unit. This is more than appropriate for our purpose, since we are *not* interested in packet-level scheduling. It is essential to understand that tighter time-scales are certainly feasible, *e.g.*, [31]. In such scenarios the bits from the RDS signal are simply utilized as broadcast beacons. We further consider our time aligned to the 104 bit groups provided by the baseband coding. Hence, any time can be represented as the most recent bit to arrive in the current group, and can therefore be referred to by its *bit index*. Since all machines receive the broadcast radio signal at nearly the same time [31], all machines receive the same bit at virtually the same time, as we demonstrate below.

To begin, the node processes the RDS bits one by one, attempting to match that station's PI code to the stream. When it encounters the code, which it recognizes via the checkword, it enters the synchronized state and resets its current bit index to 0. Once synchronized, the system no longer considers the value of each arriving bit, but instead uses it as a counter to determine the current bit index. When the system reaches 104, it resets the count to 0, and begins checking for the PI code again. If it encounters the PI code where expected (*i.e.*, 104 bits after the last occurrence), the state remains synchronized and the process is repeated.

If, on the other hand, the system fails to detect the PI code, it must return to the un-synchronized state and begin checking for the code at each bit. The next time the PI code is encountered, Wi-FM is able to return to the synchronized state. Losses of synchronization can occur as the result of radio interference scrambling the arriving bits or processing lag, which may cause some bits to be dropped. We explore the frequency of such issues in Section V. Time spent in the un-synchronized state can be reduced by employing a handful of heuristics. First, whenever a PI code is expected but not found, an XOR can determine how many of the bits match. If more than half of the bits are correct, it is likely that a small number of errors have occurred exactly on the PI code, but synchronization can be maintained. Second, the checkword at the end of each block can be used to correct a burst error up to 5 bits [32]. In this way, Wi-FM is able to use the PI code as a repetitive time-landmark in the underlying RDS format.

Station Detection When Wi-FM is activated, it must determine the most suitable FM station available to use for harmonization. To this end, it first performs a simple scan of the FM spectrum, building a list of potentially available stations. This process could be further improved by determining potential nearby stations using the devices coarse-grained location (for example, zip code, *e.g.*, [33]). Wi-FM then tunes to each candidate station, starting at the lowest frequency. It does so for 2 seconds per station, attempts to synchronize as above, and measures the fraction of time it spends in the synchronized state. The search terminates when the list of candidates is exhausted, or Wi-FM detects 3 stations with at least 95% sync rate. It then selects the lowest frequency station from these 3, or the station of the highest sync rate, if no station above 95% sync rate is detected. In Section V, we confirm that this approach works extremely well in reality, *i.e*, that high sync-rate stations are available and that nodes in a single area will converge to a single such station in a short time.

B. RDS-Relative Neighborhood Harmonization

To harmonize with neighboring signals, each Wi-FM node must independently determine a schedule during which it is best to transmit. In particular, each node must observe the current network state, determine during which RDS defined blocks other nodes are sending, and select a schedule (*i.e.* a set of blocks during which to transmit) which will result in the best performance by minimizing contention in each block. It must make these decisions independently, without the aid of a centralized scheduler.

1) Assessing Outgoing Traffic: First, a wireless node, whether an AP or STA, must determine if there is benefit to applying a schedule, or if fairness and performance are best served by continuing normally with the DCF. If only generating light traffic, it is best to simply send when available, as shown in prior work on centrally-coordinated managed wireless networks [8].

On the other hand, significant benefits can be seen if longer flows are regulated. To make this determination, Wi-FM keeps track of the amount of time in which there are packets in the outgoing buffer of a node. If the buffer is non-empty for more than 2 seconds, Wi-FM employs the below algorithm to determine the best course of action. Alternatively, application level feedback could indicate the presence of longer flows, reducing sensing time, but would require application support.

2) Assessing the Network State: If the outgoing traffic patterns indicate that a schedule is necessary, Wi-FM must measure the current traffic on the network so that it can select an appropriate schedule to harmonize with that traffic. For example, with only light traffic, it makes the most sense for a heavy sender to make space, to allow light flows to proceed more quickly. In the case of other heavy senders, Wi-FM must attempt to reach a state of effective harmonization. To measure the state, Wi-FM samples the current traffic for one group, *i.e.*, 87.6ms, and considers the times packets were sent from each other sender. It then determines during which RDS bit index each packet arrived, declaring a bit to be *active* for a sender if a packet arrived during its time period. If more than half of the bits are active in any slot, that sender is declared a *heavy hitter*, otherwise they are a *light sender*. This process is repeated periodically to reevaluate the current competing traffic.

3) Determining an Appropriate Schedule: Finally, Wi-FM must select a schedule which offers the most potential for performance gains given the observed traffic. We consider the following potential scenarios:

No Traffic. In the case in which Wi-FM detects no other senders, the system takes all available slots as its schedule. In the case when there are no other senders it makes the most sense to use all available resource. However, in neighborhood settings, we expect this scenario to occur rarely.

Light Traffic. When Wi-FM detects only light senders, taking the entire schedule would result in significant performance degradation for the light flows sharing the wireless medium, as we show below. In the case that these light flows are user web browsing, even small increases in delay could result in a degraded user experience. Therefore in such cases, it implements a *negative* schedule. Specifically, it restricts itself to $\frac{3}{4}$ of available slots, allowing light flows to operate unhindered for a time.

Heavy Single-Sourced Traffic. In the presence of a heavy hitter, Wi-FM must select its schedule in a way that carefully shares with existing flows. If the detected heavy hitter, call it Sender 1, uses more than its fair share, *i.e.*, more than half of available slots, Wi-FM chooses a schedule of half of the slots, taking any empty slots if available. When Sender 1 reevaluates the network state, it will see that it is now sharing with an additional heavy hitter and reduce its schedule accordingly, resulting in an even split of network traffic. If, in subsequent iterations, Wi-FM detects that Sender 1 has not adjusted their schedule, for example if Sender 1 is a legacy device, Wi-FM reverts to traditional DCF in order to remain competitive.

Heavy Multi-Sourced Traffic. In the case that Wi-FM detects a state which features multiple heavy-hitters, it must determine a suitable schedule that provides it with a reasonable share of sending time while still attempting to provide a fair use of resources. Wi-FM therefore considers three criteria for the selection of a schedule.

First, let set S be all the observed senders. Next, denote by l_i the number of active slots for sender i. Then, the number of slots allocated is equal to the average of the other observed transmitters. Formally,

$$k = \text{slots to schedule} = \frac{\sum_{i=1}^{S} l_i}{|S|}. \tag{1}$$

This requirement is to ensure that Wi-FM offers its local node a competitive amount of radio time. In a scenario where nodes behave in a greedy manner or simply do not deploy the Wi-FM algorithm, Wi-FM will necessarily behave greedily as well, and hence the system will converge towards the underlying DCF protocol. Later in the paper, we demonstrate that Wi-FM outperforms the underlying DCF protocol, providing clear incentives for users to deploy Wi-FM.

Second, the chosen slots should minimize the number of concurrent senders in each slot. To do so, it attempts to choose those slots which are quietest, reducing radio contention. Denote by B_k the slots in which we schedule our sender, and c_i is the number of concurrent senders observed in the ith slot. Therefore, our goal is to minimize

$$\text{total contention} = \sum_{i \in B_k} c_i. \tag{2}$$

Finally, all else being equal, Wi-FM attempts to select slots which fairly distribute the new load. Denote by B the set of all slots. Let p_i be 1 if the node is active in slot $i \in B$ and 0 otherwise, c_i is the number of concurrent senders in i, and S is the set of observed senders. Formally, we want to minimize the distance of each node from its fair share,

$$\text{fairness error} = \left| \frac{|B|}{|S|} - \sum_{i=1}^{B} \frac{p_i}{c_i} \right|. \tag{3}$$

Requirement (1) can be satisfied with a simple computation. To limit contention as required by (2), Wi-FM first orts the slots by the number of concurrent senders. Suppose the average sending length is currently k slots and there are n slots with a minimal number of concurrent senders. If $n \leq k$, Wi-FM chooses these n slots and moves on to the next smallest number of concurrent senders with the remaining $k - n$ in the same fashion. If $n > k$, (3) requires that the k slots should attempt to fairly distribute its load among the senders with which it is conflicting.

Naively, Wi-FM could consider all possible choices of k slots, compute the number of times that it collides with each sender, and choose the schedule which minimizes the maximum number of collisions. However, such an algorithm comes with significant complexity. Alternatives include greedy algorithms which simply select the first available slots which do not conflict with a sender that conflicted previously. Given the complex and often dynamic nature of network behavior, Wi-FM opts to choose randomly among the n slots, as this provides effective results, given that the number of concurrent senders has already been minimized.

None of the above described scenarios require that any of the senders belong to the same network: the network sampling, harmonization, and scheduling happen locally, based

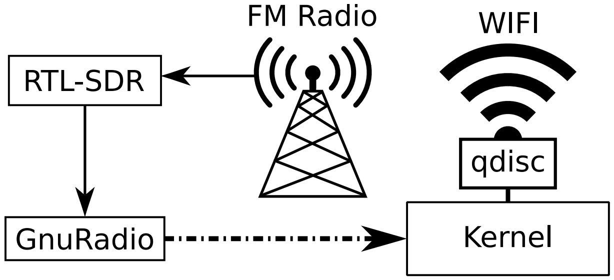

Fig. 3. The radio processing path. The RTL-SDR hears the RDS signal, digital signal processing is done in GnuRadio, and the received bits are sent to the kernel, which handles schedule processing.

on measurements by each node. In home-use settings, this can result in improvement with both a user's local network and amongst neighboring networks in the immediate area.

IV. IMPLEMENTATION

Figure 3 shows the current implementation of Wi-FM. To receive and process FM RDS signal on consumer hardware, we use a NooElec RTL-SDR DVB-T USB stick alongside GNURadio. The DVB-T USB stick uses a Rafael Micro R820T tuner, and a RTL2832U as a de-modulator and USB interface. The tuner provides a full range of about 25MHz to $1,750$MHz, which enables much greater coverage than the 87.5 to 108.0MHz used by traditional FM broadcasters in Europe and the United States.

Our nodes consist of machines running Arch Linux 3.17.4 on 3.3Ghz Intel I5 processors. Each node is equipped with the above described USB radio, as well as a TP-Link TL-WDN3800 2-antenna 802.11n PCIe card. The 802.11n cards run on the Atheros Ath9k driver, which allows the machines to act as APs when needed. In such scenarios, we use Hostapd to provide a software access points. To process raw data from the radio, we use GNURadio for our digital signal processing. We use a modified version of the Plug queuing discipline [34] to control when a node is able to send. We further decrease the available buffer on each card to limit the amount of transmitting the card can do after the qdisc is disabled [15].

The nodes were then placed in various locations around the offices at our institution. The layout can be seen in Figure 4. The nodes are placed both near windows, and internally in a multi-story building, providing a variety of radio conditions. Furthermore, our test-bed is in a relatively crowded wireless area: a large number of wireless networks and devices are constantly visible. This layout presents a reasonable example of a home network environment and the amount of contention that is likely to exist between overlapping senders.

We refrain from disabling DCF, allowing Wi-FM to take share air-time when necessary, *i.e.* non-participating devices, overlap, and failure scenarios. In all of our experiments, we disable Request to Send and Clear to Send (RTS/CTS) features in order to prevent the APs from performing any additional scheduling that might create additional interference. This is a common optimization in many home networks, as it adds latency in many scenarios. Finally, we select the 26-bit (approximately $21.89\,$ms) blocks as our time slot. This provides us with 4 time slots per group. Such a choice enables sufficient flexibility in scheduling, while limiting the potential increase in delay for a single sender. We note that the methods described in Section III-B, however, are generic, and stand to work for any number of available time slots.

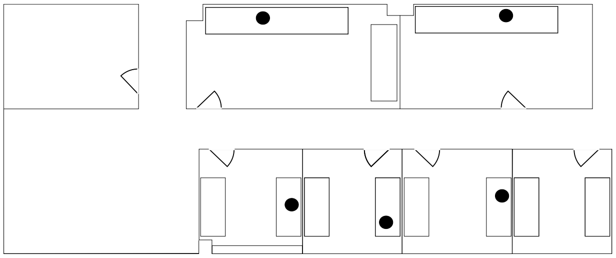

Fig. 4. The floor plan of our testbed. Dots can indicate either AP or STAs, depending on the experiment.

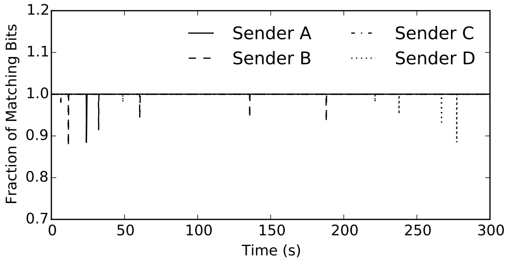

Fig. 5. The fraction of bits that matched the reference for each group.

V. FM RADIO PROPERTIES

Here, we evaluate FM radio properties, *i.e.*, FM station availability and neighborhood spatial consistency, as well as RDS signal quality. Then, we analyze how well our system implementation processes and utilizes this signal.

A. FM Station Consistency

In a sample of available radio stations in the bottom half of the FM radio range, *i.e.*, from 87.5 to 97.75 MHz, Wi-FM was able to read RDS signal from 5 different FM broadcasters. Only the bottom half of the range was used, as a sufficient number of stations were detected early in the scan. Short 2 second samples of each of these stations revealed 2 stations which were able to achieve greater than 95% sync rates, while one of the remaining achieves a 73% sync rate, and the final two obtain below 10%. A further 2 minute sample of these stations was in agreement, with the two strongest stations remained synchronized for nearly 100% of the experiment duration, while the other performed approximately as before, well below the thresholds described in Section III-A.

The similarity between the short samples and the longer 2 minute samples suggests that the synchronization can be accurately assessed on short time scales. All the nodes in our testbed obtained the same results and converged to the same RDS signal. Specifically, they all selected the lower frequency of the two qualifying stations. We use this station as our leading harmonization signal in the remainder of the paper.

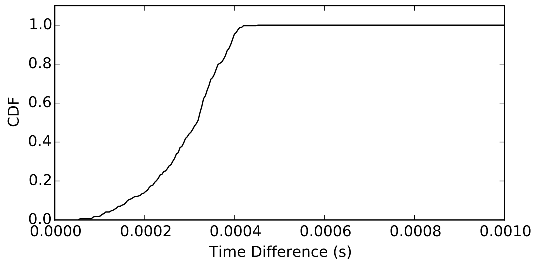

Fig. 6. Timing error between two nodes over a 1 minute period.

B. RDS Signal Consistency

Next, we explore if the RDS signal received at all of our network devices matches. Indeed, if nodes are receiving significantly different signal, they will not be able to schedule transmissions with any reasonable accuracy. To test this, we collected radio signal at all 5 nodes for 5 minutes. We then designated a single random node as the reference signal. Necessarily, the reference signal itself may have errors. Hence, simultaneous errors at all other nodes would likely indicate an error with our reference. Then for each group of 104 bits, we measure the fraction of bits that match the reference.

Figure 5 shows the result of this experiment. While each node did encounter differences, and therefore some bit errors, the received signals matched extremely closely. Furthermore, while the fraction of matched bits reaches as low as .89 at discrete points in time, all nodes maintained full synchronization for the entirety of the experiment. This stability arises from the fact that the errors never fell on the synchronizing PI codes, and as such caused no change in performance in our system. Moreover, as discussed above, an XOR scheme could retain synchronization even when PI codes are partially corrupted.

C. Software Delay

Necessarily, our implementation of Wi-FM includes delay due to the time spent on radio processing and internal switching, resulting in some timing error. To develop an understanding of the magnitude of such error, we conduct the following experiment. Two nodes, A and B, simultaneously listen to the RDS signal for 1 minute. When they encounter the station PI code, node A records its current local time, and node B sends a single UDP packet to node A over the wired network. Node A records its local time at the arrival time of the packet from node B. Both timings are therefore recorded at node A using its local time stamp counter. Node A then notes the difference between these two times. Notably, this difference includes potential difference in processing delay, as well as the network delay between the two nodes.

To establish an understanding of the expected network delay between the two nodes, we performed ping measurements between the nodes every second for 5 minutes, and took the delay to be half the average round trip time. This gave an average network delay of 371μs, with a variance of about 100μs. Figure 6 plots the CDF of the absolute difference of the recorded times less the expected value of the network delay. While there is some variation, our overall error has a median value of around 300μs, and a variance of about 100μs, similarly to the wired network variance reported above.

From Wi-FM's perspective, these results indicate that time variations are small enough to avoid damaging the synchronization state between nodes. Furthermore, it will result in very little wasted time when switching between blocks, *i.e.*, switching between senders. While this number could be further improved via software and hardware optimization, as we discuss in Section VII, the achieved time-scales serve the core Wi-FM's purpose. Moreover, Wi-FM nodes simultaneously synchronize to an out-of-band signal without *any* communication among themselves. Such an ability is most relevant in neighborhood scenarios, when devices belong to different, non-managed networks, hence cannot explicitly communicate.

VI. RDS-RELATIVE NEIGHBORHOOD HARMONIZATION

We now consider the performance of networks running Wi-FM in a number of scenarios. We show how adding relatively straightforward timing to the dynamics of the wireless network, specifically reducing contention transmission time, can have notable positive effects on the systems performance.

A. Light Traffic Scenario

First, we consider the applications of Wi-FM to a network which experiences light ambient traffic (sourced either in this or in neighboring networks), yet concurrently experiences a large amount of up-link traffic. In particular, the case when a STA uploads data to the AP, and likely out to the wider Internet. These scenarios are likely to grow in frequency, as users increasingly generate more data, both from applications responsible for data syncing between devices (iCloud, Dropbox, *etc*), and sensors in the home, *e.g.*, [1]. In such scenarios, these bulk transfers may dominate radio time, potentially greatly decreasing the performance for other small flows, for example other users performing light web browsing. Wi-FM determines a schedule is appropriate for the STA, and a sample of the network allows it to detect the presence of light flows. Wi-FM therefore employs a *negative* schedule that uses $\frac{3}{4}$ of the available slots, as described in Section III-B.

To test the performance of Wi-FM in this case, we consider the following experimental setup. A set of light flows are sent from two STAs, denoted by Sender 2 and Sender 3, to the AP. These flows consist of short requests, followed by variable length responses from the AP. Request times are modeled as Poisson arrivals with a mean arrival time of 1s, with a Pareto file size distribution for responses with a mean of 125KB and shape 1.5, as in [18]. To model the up-link traffic, a single STA (denoted by Sender 1) performs a bulk transfer to the AP for the 5 minute duration of the experiment.

Figure 7 shows the observed network packets after Wi-FM has implemented its schedule. First, we note that Sender 1 generates the most traffic, and is generally dominating the network airtime. However, we see that after bit 77 (*i.e.*, the end of block C), Sender 1 stops sending, and allows the light flows to operate unhindered during block D. Another version of this plot (not shown) indicates that while Sender 3 sends a large number of packets during slots A, B, and C, it sends a significantly fewer *bytes* relative to Sender 1.

We consider three conditions: the first is the light flows in isolation, when the light flows sent by Senders 2 and 3 do not have to share air time with the heavy flow generated by Sender

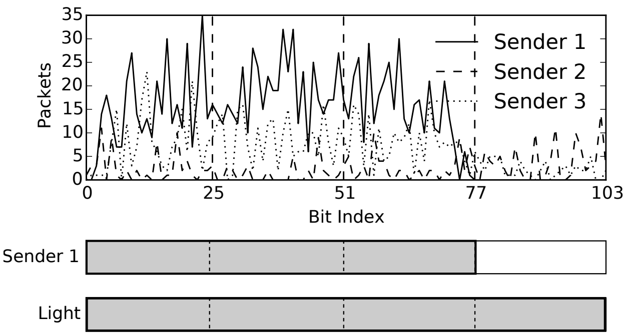

Fig. 7. The network traffic observed with Wi-FM in place.

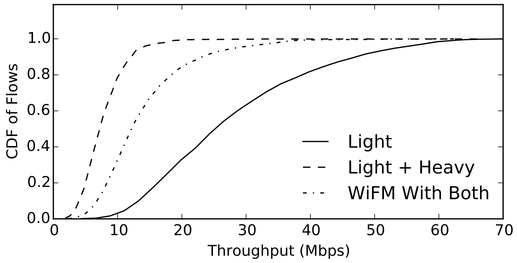

Fig. 8. The throughput of the light flows when a) the light flows are alone, b) the light flows must compete with a heavy upload, c) Wi-FM is managing the heavy flows.

1, denoted by “Light”. Next, we consider the light flows alongside the heavy flows, with no scheduling and just traditional DCF, denoted by “Light + Heavy”. Finally, we consider the case where Wi-FM is used to schedule the heavy flows for blocks A, B, and C, *i.e.* $\frac{3}{4}$ of the time. Light flows are left unscheduled. We denote this scenario by “WiFM With Both”.

Figure 8 shows a CDF of the throughput of the light flows for the three conditions. As expected, the difference at the median between the light-alone case and the light + heavy case is significant: nearly 20 Mbps. Once the heavy-hitter flow joins the network, it dramatically affects performance of the light flows. By running Wi-FM just at the heavy hitter STA, the median case sees an increase in throughput of 8.2 Mpbs, making a more acceptable environment for the light flows. We show below that the impact on the latency is even more significant.

Necessarily, the throughput of the heavy flow (not shown in the figure) is smaller in the “WiFM With Both” scenario than in the “Light + Heavy” scenario. The difference is almost exactly in proportion to its schedule, *i.e.*, 48 Mbps relative to 68 Mbps without Wi-FM. Such necessary performance trade offs should not be understood as pure altruism demonstrated by Sender 1. It is rather a collaborative effort that will equally help Sender 1 in a reversed scenario, *e.g.*, when some other source is a heavy hitter. Later, we demonstrate concurrent win-win scenarios, in which substantial throughput improvements are achievable for all senders.

Figure 9 shows similar performance for response latency of the light flows. The latency of these flows is significantly degraded when the heavy-hitter joins the network, *i.e.*, the median latency increases by more than 40 ms. Wi-FM enables a much better environment for light flows such that the median latency improves by nearly 20 ms. Moreover, the latency improvement is particularly significant at the tail of the distribution. For example, more than 40 ms for the 80-percentile latency. This makes intuitive sense: with the negative schedule in place, the light flows experience less competition in the air, hence improve their performance.

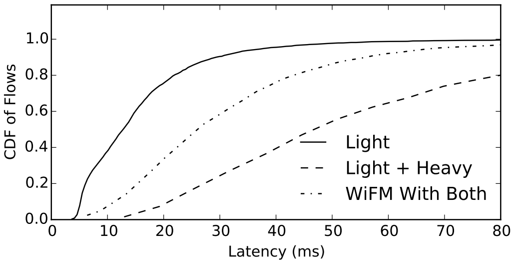

Fig. 9. The time to the first byte of the reply for the light flows in the same three secenarios.

1) Down-link Scenario: Next we consider a scenario in which there is significant down-link traffic. This is a common case in home networks, given the increasing popularity of streaming video services (*i.e*, Netflix, Hulu, Amazon). This case is further complicated by the fact that the AP becomes the heavy hitter. However, if Wi-FM enforces *node-level* schedules on the AP, performance to other STAs will also be impacted. To resolve this inter-dependency, Wi-FM allows APs to determine heavy hitters on a *per-destination* basis, restricting traffic only to those destinations pulling significant amounts of data. In this case, Wi-FM on the AP detects that there is a heavy hitter alongside light flows, which may be occurring in the same or different networks, and therefore implements the $\frac{3}{4}$ negative schedule for only the heavy hitter destination.

In the down-link case, we model the light flows as before, but now we generate a large transfer from the AP to a single STA for the 5 minutes of the experiment. We again consider three relevant scenarios, *i.e.*, “Light,” “Light + Heavy,” and “Wi-FM With Both”. We omit showing the results since they are similar to the up-link scenario. An exception is that the latency performance in the down-link scenario for “Wi-FM With Both” becomes much closer to the “Light” scenario, *i.e.*, the median distance is less than 5 ms.

B. Heavy Single-Sourced Traffic Scenario

Here, we generate a heavy-hitter (call it Sender 1) in Network I. In addition, we generate light background traffic in the neighboring Network II. Then, a new heavy-hitter (call it Sender 2), joins Network II. Below we explain how Wi-FM behaves in such a scenario. We further note that the senders could be APs or STAs: the procedure is generic.

Figure 10(a) shows the network state before Sender 2 joins. Initially, Sender 1 has converged to the three-slot scenario due to light background traffic from network II (*not* shown in the figure). Based on the neighborhood harmonization algorithm, Sender 2 should broadcast in half of possible time slots, 2 RDS blocks in this case. Sender 2 is able to observe that block D is currently available, hence it selects it. Of the remaining blocks,

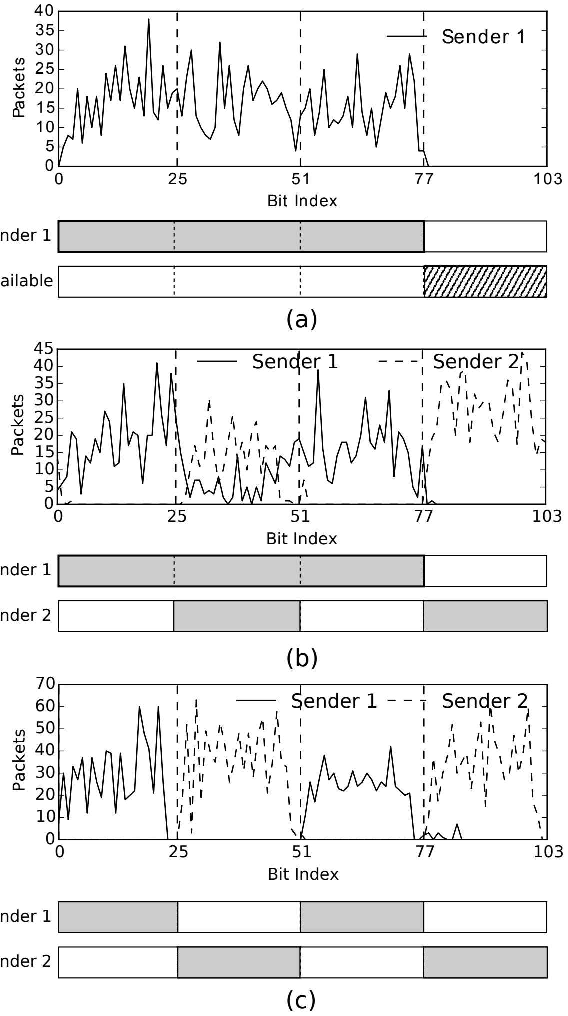

Fig. 10. (a) Packets observed from a single sender folded onto the bit indices. (b) Packets observed after Sender 2 has begin sending. (c) Packets observed after Sender 1 has adjusted.

it randomly selects a block with equal contention, *i.e.*, any of the remaining blocks, in this case block B.

Figure 10(b) presents a sample of the network traffic after Sender 2 has begun transmitting. We see that now Senders 1 and 2 must compete for block B. Blocks A and C are used only by Sender 1, while block D is utilized exclusively by Sender 2. Thus, after Sender 1 scans the network again, it discovers the presence of Sender 2, and recognizes that it is sharing the network with a single additional heavy hitter. Sender 2 therefore reduces to 2 blocks and takes those with no contention. Figure 10(c) demonstrates the state of the network after this change. Both nodes can be seen taking their fair share of the network, avoiding contention on any block.

Figure 11 shows the cumulative distribution of aggregate throughput gains sampled every second over 10 minutes seen by the Wi-FM case over the 10-minute average of the pure non-Wi-FM case. In particular we see that at least 80% of the time, the throughput increases. In at least 50% of the time, the throughput increases by 20% or more. Finally, in at least 20% of the time, these gains are greater than 40%.

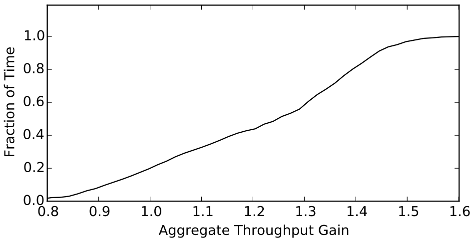

Fig. 11. Aggregate throughput gain over the pure unscheduled network.

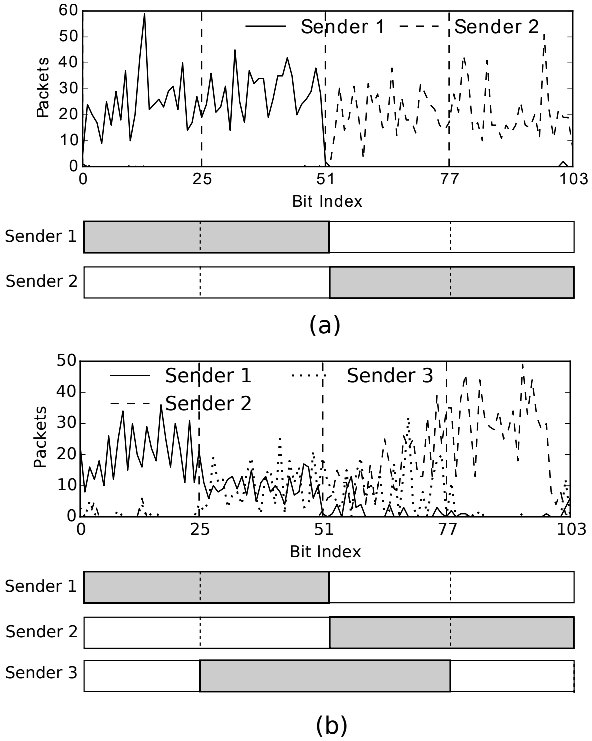

Fig. 12. (a) Packets observed from a two senders folded onto the bit indices. (b) Packets observed after Sender 3 joins.

C. Heavy Multi-Sourced Traffic Scenario

Next we consider a more complex case with multiple heavy hitters. In particular, consider the case when a new heavy hitter arrives, samples the network traffic, and finds all time slots to be fully allocated.

Figure 12(a) presents this case. Specifically, Sender 1 from network I is broadcasting in both blocks A and B, and Sender 2 from network II is broadcasting in blocks C and D. The arriving Sender 3 observes that Sender 1 and Sender 2 are both broadcasting for 2 blocks worth of time. Therefore, Sender 3 should also choose 2 blocks worth of time, for its fair share. Next, Sender 3 selects the times which minimize the number of total senders in each block. However, since all blocks are occupied, and any choice of 2 blocks results in a maximum of 2 concurrent senders, Sender 3 is able to choose any blocks to maintain levels of radio sharing. Finally, Sender 3 selects randomly, and in this particular case selects blocks B and C.

Figure 12(b) shows the network state after Sender 3 joins the network. While there is contention for the medium, by

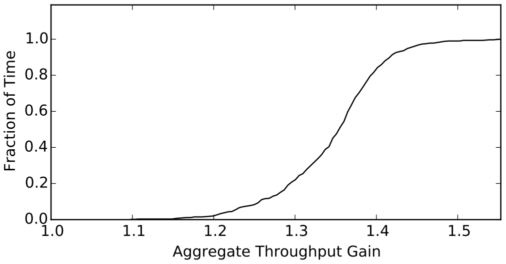

Fig. 13. Aggregate throughput gain over the pure unharmonzied scenario.

leaving DCF intact, Wi-FM is still able to function without having to communicate a schedule to the existing senders. Figure 12(b) shows that there is some spill over between time slots, which is an artifact of our implementation: since the schedules are enforced before the driver, any network level delays to packet transmission will be handled by the driver. Therefore lower level re-transmits may still occur outside of the time slot. Such issues could be solved with tight integration between devices drivers and Wi-FM.

Figure 13 shows the cumulative distribution of aggregate throughput gains seen by the Wi-FM case over the non-Wi-FM case. Here, Wi-FM throughput was *always* above the non-Wi-FM case. The gain is distributed between 20% and 50% over the non-Wi-FM case, with the median at about 35%. We emphasize that all three senders are connected to different APs, and therefore are not able to communicate directly. However, by using Wi-FM, they avoid creating worst-case contention scenarios and improve the aggregate throughput of all networks involved.

VII. DISCUSSION

A hardware implementation could likely reduce the complexity of the system. In particular, moving the relatively straightforward radio processing to a small dedicated circuit could eliminate much of the overhead in our current implementation and improve the accuracy. Moreover, such improvements would open the door for integration with more complex systems, *e.g.*, advanced distributed MIMO. Tight integration with the wireless card could further improve the situation by controlling the sending directly, rather than via the kernel queuing disciplines. While such adjustments stand to improve the response time to the radio, our experiments in Section V demonstrated that the current system provides sufficient performance in these regards.

VIII. RELATED WORK

Ambient backscatter uses ambient RF, *i.e.*, TV, cellular [35], [36] or Wi-Fi [37] signals, to enable powerless communication. Devices communicate by backscattering ambient RF signals thus enabling power-efficient radio communication. Our work is similar to ambient backscatter in the sense that we also use ubiquitous ambient signal, *i.e.*, FM radio signal, to achieve more effective Wi-Fi communication, particularly in home environments. We have demonstrated that the pervasive nature of FM radio signals coupled with growing penetration of FM radio hardware in consumer electronics, makes it possible to achieve device synchronization relative to the digital RDS component of FM signal.

While our system neither needs nor utilizes absolute time synchronization, FM broadcast signal could be used for such a purpose. There are two types of broadcast synchronization approaches in wireless networks. The first is in-band time synchronization. A representative example is the one proposed in [31], which uses in-band reference broadcast on the wireless channel to achieve time synchronization. Such an algorithm could be effectively utilized to achieve absolute time synchronization, yet using the out-of-band FM radio signal, hence without any in-band broadcasts. Moreover, contrary to the in-band broadcast which is discrete in nature, the FM radio signal is continuously present at receivers. The second is out-of-band time synchronization. In [38] the authors explored the use of RDS to synchronize clocks in sensor networks. While similar in its use of RDS, Wi-FM enables effective node harmonization among unmanaged Wi-Fi networks and devices.

There has been an extensive body of work that deals with coordination in *managed* 802.11 wireless networks. Such networks typically build knowledge about conflicting nodes and then use central coordination to either schedule AP-to-endpoint data transmissions [4], [5], [8], update transmission rates [7], avoid collisions via relative scheduling [9], utilize MIMO-type optimization [18], [19], *etc*. Beyond 802.11, other methods for distributed coordination in wireless have been considered, *e.g.* [39], [40]. Our FM radio approach enables the deployment of many of the above central coordination schemes via out-of-band broadcast synchronization. In addition to bringing value to such central coordination systems, our system enables individual endpoints to *autonomously* harmonize their transmissions in environments where they are not traditionally applicable. This is possible to achieve even when only a subset of nodes is FM-enabled.

In the context of control and coordination, our work further relates to Flashback [6], an in-band scheme that enables a control plane to 802.11 networks without altering data transmissions. The common thread between Flashback and Wi-FM lies in the desire to avoid expensive control transmissions that collide with the data plane. Contrary to Flashback, our system achieves this goal by using an existing out-of band signal and hence works across individual 802.11 network boundaries in a distributed fashion.

Finally, our work relates to the body of work on enabling TDMA within, or on top of, Wi-Fi networks, *e.g.*,[10], [11], [12], [13], [14], [15], [16], [17]. Indeed, while DCF works fine during light traffic epochs, it has been shown that TDMA provides much better performance during busy network periods. We have demonstrated that TDMA-like harmonization on top of DCF is effective in scenarios when a node is exposed to the interfering traffic from a neighboring network. Further, contrary to the above related work, which utilizes centralized network control, our system enables the creation of harmonized TDMA-like network islands in a distributed fashion among FM-enabled devices across autonomous network bounds.

IX. CONCLUSIONS

FM radio signal, broadcast by numerous local and national stations worldwide, is widely available in urban areas and beyond (24 hours a day, indoors and outdoors). In this paper, we demonstrated that this ubiquitous in-the-air FM radio availability, accompanied by the thriving on-device FM receiver hardware proliferation, provides an exciting opportunity to utilize this omnipresent signal beyond its common purpose. In particular, we focused on the problem that commonly arises in home networks when devices from neighboring Wi-Fi networks interfere and degrade each other's performance. We demonstrated that FM radio, *i.e.*, the RDS signal associated with it, provides a common medium that enables FM-enabled devices to effectively harmonize their transmissions. To the best of our knowledge, our system is the first to enable such harmonization *beyond* administrative network bounds.

Our key insights are the following: (i) The RDS digital signal accompanying FM radio has a sufficient structure to enable RDS-relative harmonization. (ii) This signal is highly resilient to data loss, *i.e.*, it retains device harmonization even in the presence of substantial reductions in matching bits. (iii) RDS could be utilized to realize device harmonization in a completely autonomous fashion. (iv) The key to such distributed harmonization lies in RDS-relative synchronization, which enables nodes to effectively infer others' scheduling choices. (v) Our approach requires no explicit communication among devices, enabling harmonization in unmanaged network scenarios, beyond administrative network bounds. (vi) Hence, it opens the door to novel cross-domain network policies in such unmanaged scenarios. (vii) This is achievable in a completely incrementally-deployable fashion. $(viii)$ Our system accomplished the above via off-the-shelf network equipment and in software. Undoubtedly, hardware-level FM radio signal processing implementations could open the doors to the deployment of advanced wireless networking algorithms and applications.

REFERENCES

[1] T. Gupta, R. Singh, A. Phanishayee, J. Jung, and R. Mahajan, "Bolt: Data Management for Connected Homes," in *Proceedings of NSDI '14*, Seattle, WA, Apr. 2014.

[2] L. Dicioccio, R. Teixeira, and C. Rosenberg, "Measuring Home Networks with HomeNet Profiler," in *Proceedings of Passive and Active Measurement Conference*, Hong Kong, China, Apr. 2013.

[3] A. Patro, S. Govindan, and S. Banarjee, "Observing Home Wireless Experience through WiFi APs," in *Proceedings of ACM MOBICOM '13*, Miami, FL, Sep. 2013.

[4] N. Ahmed, S. Keshav, and K. Papagiannaki, "OmniVoice: A Mobile Voice Solution for Small-Scale Enterprises," in *Proceedings of ACM MOBIHOC '11*, Paris, France, May 2011.

[5] Y. Bejerano and R. Bhatia, "MiFi: a Framework for Fairness and QoS Assurance for Current IEEE 802.11 Networks with Multiple Access Points," *IEEE/ACM Transactions on Networking*, vol. 14, no. 4, pp. 849–862, Aug. 2006.

[6] A. Cidon, K. Nagaraj, S. Katti, and P. Viswanath, "Flashback: Decoupled Lightweight Wireless Control," in *Proceedings of ACM SIGCOMM '12*, Helsinki, Finland, Aug. 2012.

[7] E. Magistretti, O. Gurewitz, and E. Knightly, "Inferring and Mitigating A Link's Hindering Transmissions in Managed 802.11 Wireless Networks," in *Proceedings of ACM MOBICOM '10*, Chicago, IL, Sep. 2010.

[8] V. Srivastava, N. Ahmed, S. Rayanchu, S. Banarjee, S. Keshav, K. Papagiannaki, and A. Mishra, "CENTAUR: Realizing the Full Potential of Centralized WLANs through a Hybrid Data Path," in *Proceedings of ACM MOBICOM '09*, Beijing, China, Sep. 2009.

[9] W. Zhou, D. Li, K. Srinivisan, and P. Sinha, "DOMINO: Relative Scheduling in Enterprise Wireless LANs," in *Proceedings of ACM CONEXT '13*, Santa Barbara, CA, Dec. 2013.

[10] P. Djukic and P. Mohapatra, "Soft-TDMA: A Software TDMA-based MAC over Commodity 802.11 Hardware," in *Proceedings of INFOCOM '09*, Rio De Janeiro, Brazil, May 2009.

[11] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. Sicker, and D. Grunwald, "MultiMAC - an Adaptive MAC Framework for Dynamic Radio Networking," in *Proceedings of IEEE DySPAN '05*, Dublin, Ireland, Apr. 2005.

[12] M. Neufeld, J. Fifield, C. Doerr, A. Sheth, and D. Grunwald, "SoftMAC - flexible wireless research platform," in *Proceedings of HotNets '05*, College Park, MD, Nov. 2005.

[13] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer, "WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks," in *Proceedings of NSDI '07*, Cambridge, MA, Apr. 2007.

[14] B. Raman and K. Chebrolu, "Design and Evaluation of a New MAC Protocol for Long-Distance 802.11 Mesh Networks," in *Proceedings of ACM MOBICOM '05*, Cologne, Germany, Aug. 2005.

[15] A. Rao and I. Stoica, " An Overlay MAC Layer for 802.11 Networks," in *Proceedings of MobiSys '05*, Seattle, WA, Jun. 2005.

[16] A. Sharma and E. Belding, "FreeMAC: Framework for Multi-Channel MAC Development on 802.11 Hadware," in *Proceedings of ACM SIGCOMM PRESTO '08*, Seattle, WA, Aug. 2008.

[17] A. Sharma, M. Tiwari, and H. Zheng, " MadMAC: Building a reconfigurable radio testbed using commodity 802.11 hardware ," in *Proceedings of IEEE SECON SDR '06*, Reston, VA, Sep. 2006.

[18] S. Kumar, D. Cifuentes, S. Gollakota, and D. Katabi, "Bringing Cross-Layer MIMO to Today's Wireless LANs," in *Proceedings of ACM SIGCOMM '13*, Hong Kong, China, Aug. 2013.

[19] V. Yenamandra and K. Srinivasan, "Vidyut: Exploiting Power Line Infrastructure for Enterprise Wireless Networks," in *Proceedings of ACM SIGCOMM '14*, Chicago, IL, Aug. 2014.

[20] B. Kinney, "Unlock The FM Radio Hiding In Your Smart Phone," http://digital.vpr.net/post/unlock-fm-radio-hiding-your-smart-phone.

[21] "Broadcom 802.11 Wireless LAN Solutions," http://www.broadcom.com/products/Wireless-LAN/802.11-Wireless-LAN-Solutions.

[22] "Broadcom's New Combo Chip Integrates 802.11n Wi-Fi, Bluetooth 4.0 + HS and FM to Bring New Multimedia Applications to Smartphones, Tablets and Other Mobile Devices," http://www.broadcom.com/press/release.php?id=s549642.

[23] "FM Radio in Mobile Phones," http://radioheardhere.com/fm-mobile.htm.

[24] BBC, "Radio Industry Moves to Meet Demand for Hybrid Radio," http://www.bbc.co.uk/mediacentre/latestnews/2014/hybrid-radio.

[25] "NextRadio: FM Radio on Your Smartphone," http://www.sprint.com/landings/nextradio/.

[26] "Sprint customers to enjoy local FM radio on smartphones via FM radio chip," http://newsroom.sprint.com/news-releases/sprint-customers-to-enjoy-local-fm-radio-on-smartphones-via-fm-radio-chip.htm.

[27] "FM Radio Phones," http://www.phonearena.com/phones/Multimedia/FM+Radio.

[28] "Free Radio on My Phone," http://freeradioonmyphone.org/.

[29] E. Oswald, "RIAA, broadcasters want mandatory FM radio in cell phones, electronics," http://betanews.com/2010/08/17/riaa-broadcasters-want-mandatory-fm-radio-in-cell-phones-electronics/.

[30] L. Stimson, "Worldwide FM Smartphone Effort Underway," http://www.radioworld.com/article/worldwide-fm-smartphone-effort-underway/272856.

[31] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of OSDI '02*, Boston, MA, Dec. 2002.

[32] National Radio Systems Committee, "United states rdbs standard," ftp://ftp.rds.org.uk/pub/acrobat/rbds1998.pdf.

[33] "Radio Locator," http://radio-locator.com/cgi-bin/locate.

[34] B. Cully, G. Lefebvre, D. Meyer, M. Feeley, N. Hutchinson, and A. Warfield, “Remus: High availability via asynchronous virtual machine replication,” in *Proceedings of NSDI ’08*, San Francisco, CA, 2008.

[35] V. Liu, A. Parks, V. Talla, S. Gollakota, D. Wetherall, and J. Smith, “Ambient Backskatter: Wireless Communication Out of Thin Air,” in *Proceedings of ACM SIGCOMM ’13*, Hong Kong, China, Aug. 2013.

[36] A. Parks, A. Liu, S. Gollakota, and J. Smith, “Turbocharging Ambient Backscatter Communication,” in *Proceedings of ACM SIGCOMM ’14*, Chicago, IL, Aug. 2014.

[37] B. Kellogg, A. Parks, S. Gollakota, J. Smith, and D. Wetherall, “Wi-Fi Backscatter: Internet Connectivity for RF-Powered Devices,” in *Proceedings of ACM SIGCOMM ’14*, Chicago, IL, Aug. 2014.

[38] L. Li, G. Xing, L. Sun, W. Huangfu, R. Zhou, and H. Zhu, “Exploiting fm radio data system for adaptive clock calibration in sensor networks,” in *Proceedings of MobiSyS ’11*, New York, NY, USA, 2011.

[39] P. Chaporkar and A. Proutiere, “Optimal distributed scheduling in wireless networks under sinr interference model,” in *Proceedings of Allerton Conference*, Oct 2013.

[40] D. Qian, D. Zheng, J. Zhang, N. Shroff, and C. Joo, “Distributed CSMA Algorithms for Link Scheduling in Multihop MIMO Networks Under SINR Model,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 746–759, Jun. 2013.