Harmony: Content Resolution for Smart Devices using Acoustic Channel

Mostafa Uddin, and Tamer Nadeem
Department of Computer Science, Old Dominion University
Email: {muddin,nadeem}@cs.odu.edu

***Abstract*—In this paper, we utilize a novel communication framework Acoustic-WiFi to develop a smart contention resolution scheme *Harmony* among the contending devices to address the overhead of the traditional Wi-Fi backoff scheme (i.e. contention window countdown, DIFS) and reduce the overall collisions among the devices. *Harmony* uses the acoustic channel for contention resolution in Wi-Fi networks. To the best of our knowledge, *Harmony* is the first to leverage the acoustic interface on commodity smart devices as an addition control channel in parallel with the Wi-Fi interface. We evaluate our scheme using real testbed and simulation. Testbed experiments show 40% throughput gain over traditional Wi-Fi networks, while simulation results show more than 27% gain for dense networks.**

I. Introduction

Because of the low/free cost and high throughput of Wi-Fi networks as well as the data capping restriction enforced by cellular providers, users of smart devices (e.g., smartphones) are turning to Wi-Fi for their ever-heavier data-driven surfing, uploading and downloading. Recent research has found that 74% of smartphones data goes through Wi-Fi [3]. Given the prominent role of the Wi-Fi interface in smart devices, enhancing WiFi network performance and functionalities is very essential to support the widespread use of smart devices. Previous research has tried to address Wi-Fi inefficiencies by utilizing the other co-located wireless interfaces (e.g., Bluetooth, 3G, and ZigBee) for improving the performance of WiFi networks [4]–[6], [14], [22]. Overall, this approach of utilizing such cross-network interface (e.g. Wi-Fi and Bluetooth) can open up new possibilities for developing smarter and more efficient wireless communication.

In this work, we leverage a novel communication framework that utilizes the acoustic/audio channel (i.e., microphone/speaker) as a control channel to develop more efficient Wi-Fi networks. More specifically, in this paper we utilize a cross-interface Acoustic-WiFi framework that integrates the Wi-Fi interface and the acoustic interface of the smart device. Unlike the previous work of integrating two Radio Frequency (RF) interfaces, our framework, for the first time, integrates two wireless interfaces with two different propagation mediums (i.e. electromagnetic and acoustic). Despite the challenges of integrating these two wireless interfaces with different characteristics, Acoustic-WiFi framework leverages the unique features of both interfaces to enhance the overall Wi-Fi network performances. As a proof of concept, we showed in a previous work [26] how Acoustic-WiFi framework could be utilized to improve the efficiency of Wi-Fi power saving scheme for smart devices. In this paper, we utilize Acoustic-WiFi framework to develop a smart contention resolution scheme among the contending devices to address the overhead of the Wi-Fi backoff scheme (i.e. contention window countdown, DIFS) and reduces the overall collisions among the devices.

Wi-Fi backoff scheme provides arbitration among contending nodes to access the share medium in which each node selects a random number and starts counting down. The node that reaches zero first, transmits its data while others freeze their counting until the medium is free again. However, while all nodes are counting down, the channel remains idle which results in under-utilization of the medium. Moreover, in case of a network congestion, the random number range increases exponentially and results in increasing the channel idle time. While authors in [12] show 30% reduction in Wi-Fi (i.e. 802.11g) throughput because of the backoff, authors in [13] show the severity of the backoff when transmitting at high data rates. In this paper, we propose *Harmony* to offload the burden of the backoff scheme to the acoustic channel and reduce the impact of collision over the Wi-Fi channel.

In realizing *Harmony*, we address a number of research challenges: (1) Slow propagation delay of acoustic signals that results in a slow contention resolution over the acoustic interface in comparison to the RF interface. We address this challenge by developing an algorithm to select multiple nodes instead of selecting a single node. In addition, we use a pipelining technique between the contention resolution and the data transmission operations to eliminate the overhead of acoustic signal's propagation delay. (2) Selection of multiple nodes requires a low-overhead scheme to coordinate data transmissions among nodes. We address this challenge by ranking each node and allowing them to transmit data according to their ranking. (3) Limited acoustic bandwidth on smart devices, and the frequency shifting due to reflection and diffraction increase the collision over the acoustic channel. In this paper, we develop a technique to detect and resolve the collision over the acoustic channel.

We evaluate *Harmony* scheme using real testbed and simulation. Experiments using testbed with 10 smart devices show 40% throughput gain over traditional Wi-Fi networks (i.e., running backoff scheme), while simulation results show more than 27% gain for dense networks.

We summarize our contributions in this paper as follows:

- Design and develop *Harmony* that uses the acoustic channel for contention resolution in Wi-Fi networks. To the best of our knowledge, it is the first to leverage the acoustic interface on smart devices as an addition control channel in parallel with the Wi-Fi interface.

- Address different challenges and implement *Harmony* on Commodity-of-the-Shelf smartphones.
- Evaluate *Harmony* using a real testbed and NS3 simulation for different realistic scenarios.

In section III, we start describe our scheme, research challenges and how we address those challenges. In following section IV, we address the challenges of collision over the acoustic channel. In section V and VI, we describe the implementation and evaluation of the *Harmony* scheme respectively. Finally, in section VII we describe the limitation of our scheme.

II. Motivation and Related Work

Application Scope: Slow propagation speed of the audio signal and the incapability of penetrating the walls makes our *Harmony* scheme limited for scenarios with devices in close proximity (e.g., inside a conference/meeting room, multiple devices inside a room or apartment etc.). Given the wide adoption of Wi-Fi in many types of smart devices, Wi-Fi becomes an interesting communication medium for many peer-to-peer applications and services (e.g., file sharing, multiplayer games, media streaming) on smart devices for home and office scenarios. In these usage scenarios, smart devices are relatively in close proximity, which make our *Harmony* scheme applicable in such practical scenarios.

Characteristics of acoustic interface: Compared to other common wireless interfaces (e.g., Bluetooth, infrared, ZigBee), the acoustic interface has interesting features that motivate our work in this paper. Unlike Bluetooth interface, the acoustic hardware/software in commodity smart devices are open, flexible and modularized. This enables us to develop low-level (MAC/PHY) protocols and to interact with the WiFi lower networking layers (i.e., MAC layer). In addition, hardware that comes with commodity smart devices is enough to implement the proposed acoustic interface with no extra hardware. Moreover, acoustic interfaces do not require direct line-of-sight for communication as in infrared interface. Finally, acoustic interface works as full duplex, therefore it is able to transmit (i.e., play) and receive (i.e., record) signals simultaneously. These features motivate us to design and develop the Acoustic-WiFi framework for smart devices. Unlike previous work that utilizes the acoustic interface at the application layer only [18], [21], [27], [28], [30], this project aims to integrate the acoustic interface with the WiFi interface at lower layers (e.g., MAC layer) as well as upper layers (e.g., application layer) providing more control and flexibility in developing novel schemes and applications to support WiFi networks.

Acoustic interface as communication medium: In underwater data communication [9], [15], acoustic signals for communication is a common practice. Researchers also have conducted several experiments using acoustic communication in air as a medium [19]. In recent works, the audio interface (microphone & speaker) of smartphones has been used as a new data communication interface. For example, in Dawni [20], authors have developed an acoustic based NFC system that uses the audio interface of the mobile phones to transfer data over short distance. Most of the acoustic based data communication systems are using the higher audio frequency zone that is beyond normal human perception (i.e., 16K-22K). However, the acoustic interface with such low limited bandwidth has very low data rate. Moreover, the acoustic channel becomes more noisy and error prone compared to radio frequency channel for the same range distance. Therefore, Acoustic-WiFi framework only leverages the acoustic interface as a parallel control channel to the Wi-Fi interface.

Contention Resolution: The idea of backoff scheme to resolve the contention in wireless medium is not new [7]. Numerous research has been done both experimentally and analytically to evaluate the backoff scheme. Countless algorithms have been proposed to adapt the backoff scheme based of network traffic and contention to improve the overall performance. Covering such wide area of literature is difficult; therefore we will only focus on some of the recent new ideas that are relevant to us.

Several TDMA-style schemes have been proposed to eliminate the overhead of contention by allocating a fixed channel to each node [17]. These schemes require tight time synchronization and a centralized controller. In order to address TDMA difficulties, ZMAC [23] - a hybrid MAC is proposed to work like TDMA in high contention and CSMA in low contention. However, ZMAC requires heavy coordination between the nodes to understand the topology, which results in high overhead impact on the communication.

There has been some recent efforts to migrate the backoff scheme to frequency domain by using additional antenna [11], [24], [25]. Similar to Sen's work [25] on contention resolution, *Harmony* also uses the frequency domain. However, contrary to the requirements of two Wi-Fi antennas in [25], *Harmony* doesn't require any additional hardware. More specifically, unlike the other schemes that are hardware (or PHY layer) based solutions as in [25], *Harmony* is totally a software based solution that is designed and implemented based only on the available hardware resources of the smart devices (i.e. microphone/speaker/audio driver).

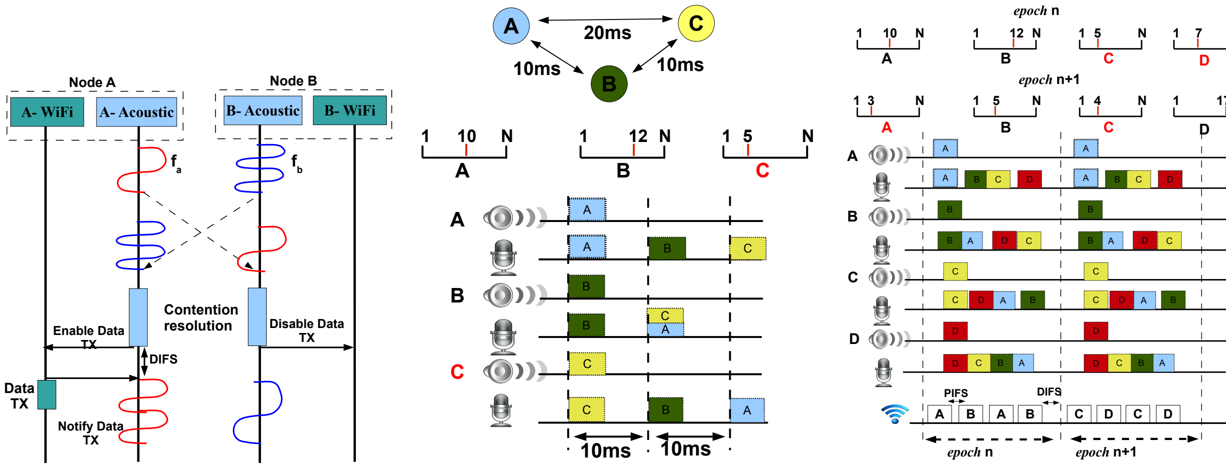
III. *Harmony*

Similar to the backoff scheme, *Harmony* is a distributed protocol with randomization to select the winning nodes from the contending nodes. In this section, we start with an overview of the *Harmony* scheme. Then, we describe the scheme for a simple scenario in which a single node is selected. Following, we generalize the scheme for selecting multiple nodes, and finally, we discuss collisions in acoustic domain.

A. Harmony Overview

In *Harmony*, we have two main operations: (1) Contention resolution over the acoustic channel, and (2) Data transmission over the Wi-Fi channel. Throughout this paper, we describe details on these two operations in incremental fashion.

Figure 1a shows a very simplistic overview of *Harmony* two operations for two contending nodes `A` and `B`. In the contention resolution operation, each acoustic interface of each node generates an *acoustic tone* with random frequency (i.e., f_a and f_b respectively) over the acoustic channel. We use *acoustic tone* to refer to a single frequency sinusoidal acoustic



(a) (b) (c)

Fig. 1: a) **Basic overview of *Harmony*.** b) **Selecting single node for data transmission in *Harmony*.** c) **Selecting multiple nodes for data transmission in *Harmony*.**

signal. After receiving the *acoustic tones* from each other, contention resolution takes place in each node. Since node A generates the lowest frequency tone, $f_a < f_b$, therefore node A wins the contention.

Once node A wins the contention, its acoustic interface triggers the data transmission over the Wi-Fi interface. On the other hand, node B disables its data transmission. In *Harmony*, data transmission operation is initiated after DIFS time period. Therefore, as shown in Figure 1a, node A's Wi-Fi interface waits for DIFS period before transmitting the data. At the same time, the acoustic interface initiates its next round of contention selection process. Thus, *Harmony* imitates the backoff scheme without waiting for the count down of contention window. Instead, *Harmony* saves time by running the node selection process over the acoustic channel in parallel with the data transmission of the previous round.

B. Single Node Selection

In designing *Harmony*, we start with assuming all nodes can hear each other over the acoustic channel, which means that there is no hidden node. We name this scenario as Single Acoustic Domain (SAD). In addition, we assume the network is saturated and all the nodes have enough data to transmit.

Acoustic channel - contention resolution: We use the frequency band 16 kHz to 21 kHz for the acoustic channel communication. Considering the robustness, we select the lower limit of the frequency band to be 16 kHz since the majority of the background noises, human conversation, music player, FM radio has frequencies up to 12 kHz. Given the limitation of the current smart devices (e.g., smartphones) and their acoustic high sensitivity, we are able to generate and capture sounds in high frequencies up to 21 kHz [20]. Within this frequency band, we select a set of fixed frequencies for generating the *acoustic tones*, referred to as Frequency Set (F). In *Harmony*, we use an index number to label the frequencies in F, which we refer to as s. For example, if we have N different frequencies in F, we index them from 1 to N respectively. Note that *tone numbers* are defined in the incremental order of the frequencies in F.

Figure 1b shows a simple example of *Harmony* where we have three nodes A, B, and C within the range of each other. In this example, only one node is allowed to win the contention. In the figure, nodes A, B, and C transmit *acoustic tones* corresponding to the randomly chosen *tone numbers* 10,12, and 5 respectively. Note that, due to the slow propagation speed of acoustic signals, there is some noticeable delay in receiving the *acoustic tones*. Once the acoustic interface of a node receives an *acoustic tone*, it maps the detected frequency of the *acoustic tone* to the corresponding *tone number*. After listening to all *acoustic tones*, each node will know that 5 is the lowest *tone number* among the nodes. Since node C generated the lowest *tone number*, it wins the wireless medium to transmit its data after a DIFS period. For next contention round, the other two nodes A and B will reduce their previously randomly chosen *tone number* to $(10-5) = 5$ and $(12-5) = 7$ respectively, and node C will select a new random *tone number* to generate over the acoustic channel after a DIFS period.

Algorithm 1 *Harmony:* Single Node Selection Scheme

```
tone ⟵ random(1,maxTone)
generateTone(tone)
listen to other tones tone_1,tone_2,tone_3,...,tone_N
minTone ⟵ min_{i=1..N} tone_i
tone ⟵ (tone − minTone)
if tone ≠ 0 then
    goto step 2
else
    transmit pkt
end if
```

Wi-Fi channel - data transmission and interaction: In *Harmony*, data transmission through Wi-Fi interface is controlled by the acoustic interface thru a shared flag variable. This shared flag variable is enabled and disabled to control the data transmission from the Wi-Fi driver buffer. In Figure 1b, since node C wins the contention for data transmission, its acoustic interface sets the shared flag to trigger its data transmission. Once the Wi-Fi interface of node C transmits its data, it clears the shared flag.

Challenge - propagation delay: Algorithm 1 describes the above basic scheme for selecting a single node. Given the slow propagation speed of the acoustic signals, the contention resolution operation happens over a long duration compared to data transmission using RF signals. This duration is even longer than a typical aggregated transmission of 802.11n [29]. For example, the acoustic tone transmissions takes almost 20ms to propagate over the 7 meters between both nodes A and C as shown in Figure 1b. Therefore, it is inefficient to run the contention resolution operation for every transmission. One way to overcome this inefficiency is to allow the selected node to transmit a batch of packets instead of a single transmission. However, to guarantee fairness between nodes and avoid this inefficiency, we propose the following scheme to select multiple nodes instead of a single node.

C. Multiple-Node Selection

In order to address the inefficiency of content resolution operation, we adopt the following two actions: (1) Multiple-node selection - instead of selecting a single node, we select multiple nodes over acoustic channel without any additional delays. (2) *Pipelining* the two main operations of *Harmony*; contention resolution over the acoustic channel, and data transmission over the Wi-Fi channel.

Acoustic channel - contention resolution: Figure 1c shows a basic example of multiple-node selection scheme with two consecutive *epoch* periods n and $n+1$. *Epoch* period is the overall time required for the contention resolution operation (i.e., selecting the winning node(s) among the contending nodes+DIFS time) over the acoustic channel. In *Harmony*, we select a fixed value for the *epoch* period based on the propagation time, duration of the *acoustic tone* and the computation time for detecting the *acoustic tone's* frequency (section V for details). In this example, instead of selecting one node, we select two nodes that will coordinate their data transmissions over the Wi-Fi channel. As shown in Figure 1c, nodes A and B are transmitting their data in *epoch* period n while they were selected in the previous *epoch* period $n-1$. Similarly, nodes C and D that are selected in *epoch* period n will transmit their data in the next *epoch* period $n+1$. Thus, we pipeline the action of selecting nodes over acoustic channel with the data transmission over the Wi-Fi channel.

As we see in Figure 1c, nodes A, B, C, and D generate *acoustic tones* corresponding to the randomly selected *tone numbers* 10, 12, 5, and 7 respectively during the *epoch* period n. In *Harmony*, every node independently generates its *acoustic tone* at the beginning of the *epoch* period with no time restriction or synchronization requirement to generate these *acoustic tones*. After receiving *acoustic tones* from each other, node C detects that it has the minimum *tone number* (i.e., 5) and, hence, ranked 1st while node D has the second to minimum *tone number* (i.e., 7) and ranked 2nd. Thus, in *epoch* period $n+1$, nodes C and D transmit their data in the order of their ranks. In the next *epoch* period $n+1$, the two unselected nodes A and B update their *tone numbers* from 10 and 12 to 10-7=3 and 12-7=5 respectively, where 7 is the maximum *tone number* among the selected nodes in *epoch* period n. On the other hand, C and D select two new random *tone numbers* (i.e. 4 and 17 respectively) for *epoch* period $n+1$. Then, since nodes A and C have the 1st minimum and the 2nd minimum *tone numbers* respectively, nodes A and C transmit their data in the order of their ranks during *epoch* period $n+2$.

Algorithm 2 describes the above scheme for selecting multiple nodes. The algorithm has one input parameter, k, which represents the number of nodes to select for transmission during the *epoch* time. Step 3 in the algorithm shows the list of captured *acoustic tones* in incremental order of their *tone numbers*. Note that n represents the number of contending nodes. Step 4 calculates the $rank$ of a node based on the relative order of its *tone number* with respect to the captured *tone numbers* from other nodes. For example, in Figure 1c, node B generates the 3rd minimum *tone number* during *epoch* period n and therefore its rank is 3. In step 5, each node updates its *tone number*, if it is greater than zero, and wait for the next *epoch* period to generate *acoustic tone* corresponding to its updated *tone number* (step 6-8).

Wi-Fi channel - data transmission and interaction: In Algorithm 2, steps 10-19 describe the overall actions of the Wi-Fi interface as well as the interactions between the WiFi and the acoustic interfaces. Selected nodes in an epoch are alternating their transmissions during the epoch duration in a round-robin fashion. Once the Wi-Fi interface of a selected node senses the medium is free for Wi-Fi `PIFS` time period where `PIFS<DIFS`, it sends a notification to the acoustic interface to decrement the counter i maintained by the acoustic interface (step 13) that is initially initialized to the node's $rank$ (step 10). When the counter reaches zero, the acoustic interface sets the shared flag to trigger the Wi-Fi interface to transmit its data (step 15-16). Once the Wi-Fi interface sends the data frame, it clears the shared flag and notifies the acoustic interface to reinitializes the counter value to k (step 17) and repeats the process (steps 11-19). Thus, the selected nodes maintain their order of data transmission according to their ranks until the current *epoch* period ends. Note that, Wi-Fi interface clears the shared flag for both successful and unsuccessful transmissions. Therefore, if a data transmission failed due to data corruption, Wi-Fi interface gets the chance to retransmit the data frame at the next transmission slot according to its $rank$.

In *Harmony*, every node detects the starting of a new *epoch* period when it senses the Wi-Fi medium is idle for at least `DIFS` period similar to Wi-Fi standards. In other word, when an *epoch* period ends, all the nodes restrain from sending data for at least `DIFS` period of time. Thus, if a new node joins the network, it can easily detect the starting of the *epoch* period along with other existing nodes. When the medium is sensed to be idle for `DIFS` period: a) nodes selected in previous *epoch* start their data transmission over

Wi-Fi interface according to their $ranks$, and b) acoustic interfaces get notified corresponding Wi-Fi interfaces to mark the starting of a new *epoch* period in which each acoustic interface generates an *acoustic tone* corresponding to either an updated *tone number* (in case the node was not selected in the previous *epoch* period), or a new random *tone number* (in case the node was selected in the previous *epoch* period).

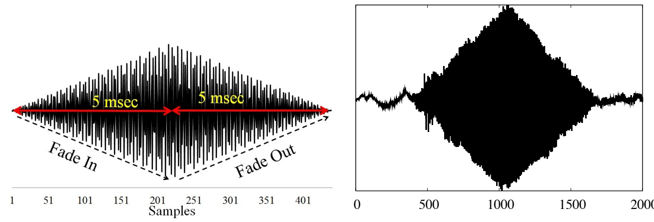
Challenge - collision: Another big challenge in this scheme is collisions over acoustic channel. A collision happens when two nodes generate the same *acoustic tone* with the same *tone number*. The collision of *acoustic tones* over the acoustic channel imposes several challenges in selecting multiple-nodes in our Algorithm 2. For example, in Figure 1c, during the *epoch* period n, a collision happens when node B generates an *acoustic tone* corresponding to *tone number* 7 similar to node D. In that case, there is ambiguity between nodes B and D during the selection of only two nodes for data transmission. In another scenario, when node D generates an *acoustic tone* corresponding to *tone number* 5 similar to node C, there is no ambiguity in selecting the two nodes. However, since nodes C and D have the same rank, their data transmission will collide over the Wi-Fi channel. In the following section, we address these collision challenges over the acoustic channel.

Algorithm 2 *Harmony:* Multiple-Node Selection Scheme

Require: k, number of nodes to select
1: $tone \longleftarrow$ random(1,$maxTone$)
2: generateTone($tone$)
3: listen to other tones $tone_1$<$tone_2$<$tone_3$...<$tone_n$ $\{tone$ is an element of T=$\{tone_1$,$tone_2$...$tone_n\}\}$
4: $rank \leftarrow$ rankcal($tone$,T)
5: $tone \longleftarrow$ max(0,$tone - tone_k$)
6: **if** $tone \neq 0$ **then**
7: wait until next *epoch* time start
8: goto step 2
9: **else**
10: $i \leftarrow rank$
11: **for** until *epoch* time is not done **do**
12: **if** Channel is free for PIFS period **then**
13: $i = i - 1$
14: **end if**
15: **if** $i == 0$ **then**
16: transmit pkt
17: $i \leftarrow k$
18: **end if**
19: **end for**
20: **end if**

IV. Collisions in Acoustic Channel

The design of acoustic hardware enables us to develop a full duplex acoustic channel. Therefore, with careful designing of *acoustic tones*, collision detection is possible over acoustic channel. In this section, we describe a novel collision detection technique to address collisions in *Harmony*. Initially, we start with describing how we design an *acoustic tone*. Then, we describe the Collision Detection (CD) scheme for the acoustic channel by showing how we select the possible number of frequencies (i.e., the frequency set, F) from the chosen frequency band (i.e., 16 kHz - 21 kHz). Finally, we show how to resolve the collision by designing an algorithm that runs double rounds of acoustic contention resolution operation.



(a) (b)

Fig. 2: a) **Shape of a generated *acoustic tone*.** b)**Received *acoustic tone* through microphone.**

A. Acoustic Tone

In this section, we describe the design issues of an *acoustic tone*. In designing the acoustic tone, we consider two issues: (1) The length of the tone, and (2) The shape of the tone.

The length of an *acoustic tone* is an important factor to reliably capture and detect the signal. In the case of a long signal, it is comparatively easy to receive and detect. On the other hand, using long *acoustic tone* could get overlapped with another *acoustic tone* with the same *tone number*. In such case, it is harder for a node to detect the collision. We empirically found that an *acoustic tone* of 10 milliseconds duration, which is equal to 441 samples when sampling rate is 44100 Hz, is enough to easily detect the signal over transmission range of 25 meters.

Generating an *acoustic tone* in such high frequency over smartphone speaker creates clicking noise, which is perceptible to human. Since speakers are mechanical system, sudden generation of high frequency have such artifacts [16]. In order to alleviate this problem, we use the amplitude fade-in and fad-out approach. In doing this, we select the duration of both fade-in and fade-out to be 5 milliseconds. Figure 2a shows, the overall shape of an *acoustic tone* and Figure 2b shows an example of the captured *acoustic tone*.

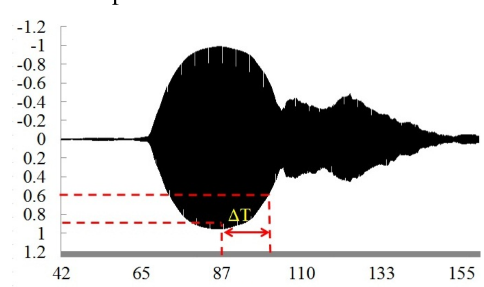


Fig. 3: **The cross-correlation of the captured own *acoustic tone* with no collision.**

B. Collision Detection

Giving that the acoustic interface is full duplex, a node in *Harmony* can hear *acoustic tone*s that are generated either by itself and by nearby nodes without being able to identify the original source of the acoustic tone. Since a collision in *Harmony* happens when a node hears an *acoustic tone* with an identical *tone number* to its own *tone number*, there are two possible scenarios: (1) the two acoustic tones are not overlapped over time, or (2) the acoustic tones are overlapped over time.

In the first scenario (scenario #1), it is easy for a node to detect the collision because the two *acoustic tones* are received separately. However, in the second scenario (scenario #2), it is challenging for a node to detect these two acoustic tones with the same *tone numbers*. Figure 3 shows a scenario when a node applies the cross-correlation to the captured signal of its own *acoustic tone* with no collision. In the figure, ΔT represents the time required for the absolute correlation value to drop from its maximum to lower than 0.6. On the other hand, Figure 4 shows the correlation value of the captured *acoustic tone* with scenario #2 collision. As we can see, due to reverberation and reflection, the ΔT value is larger compared to the non-collision scenario in figure 3. A key observation from Figure 4 is that while an *acoustic tone* is generated from a distance, the signal get elongated due to reverberation. Therefore, in scenario #2 collision, the ΔT value is relatively large from non-collision scenario. Note that, we assume that distance between any two nodes is longer than 3 feet, which is a reasonable distance between two people in a gathering (i.e meeting, conference).

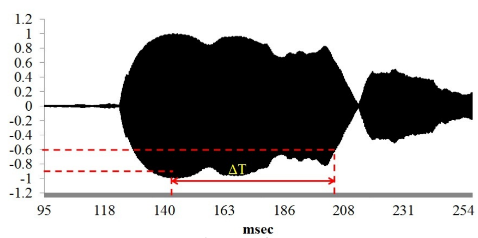


Fig. 4: **The cross-correlation of the captured *acoustic tone* with collision.**

Figure 5a shows the scatter plot of the ΔT values for both the collision (scenario #2) and the non-collision scenarios. From the scatter plot, we observe two different clusters of ΔT values that clearly represents the two scenarios. Therefore, we use ΔT value as a measurement to detect collisions corresponding to scenario #2.

C. Frequency Set F

The larger the number of frequencies in frequency set F, the less the probability of the contending nodes to select the same *tone number*. Therefore, our objective is to select as many frequencies for the frequency set F as possible from the chosen frequency band in order to reduce the collision probability. However, while an *acoustic tone* travels from one node to another, it gets reflected, diffracted over the air medium [16] that makes it hard to detect the original frequency of the *acoustic tone* due to frequency shifting and the decay in signal strength. Therefore, adjacent frequencies in F should be selected such that they have enough gaps in between in order to reduce the impact of the erroneous detection of the frequency in an *acoustic tone*.

Figure 5b shows the cross-correlation (in dB) between a captured *acoustic tone* transmitted at specific frequency and the captured signal of the same *acoustic tone* but when it is transmitted with a shifted frequency. We experimented with different shift values (i.e., 0 Hz, 100 Hz, 200 Hz, 300 Hz, 400 Hz, and 500 Hz) as well as different transmission distances. From the figure, it is obvious that it is not easily to distinguish the original frequency of an *acoustic tone* when the shift is 100 Hz or less. On the other hand, the original frequency is easily distinguishable when we use gap of 200 Hz or more. Therefore, in *Harmony*, we choose the frequencies in F such that they are 200 Hz apart from each other. Giving the frequency range is 16 kHz to 21kHz, we have a total of 26 different frequency in F.

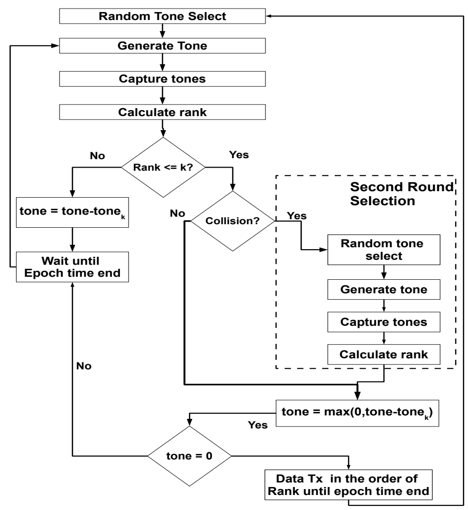
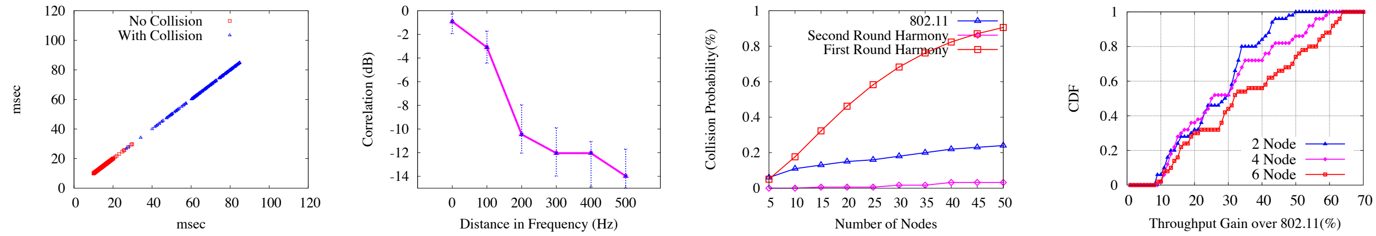


Fig. 6: **Complete flow chart of the *Harmony* scheme including the second round selection.**

D. Double Rounds of Multiple-Node Selection

With only 26 available frequencies to choose from, large dense networks will suffer from high collisions of acoustic tones. In order to resolve collisions over the acoustic channel, our basic idea is to run a second round of random tones generation for only the set of nodes that are selected in the first round if a collision is detected. Figure 5c shows the overall collision probability as the number of contending nodes increases. We observe that with only 26 unique frequency tones, the probability of collision increases rapidly with the number of contending node. Therefore, we run a second round of contention resolution for only the set of *possible winning* nodes selected in the first round. A node is considered a *possible winning* node if its $rank$ in the first round is less than or equal to the intended number of nodes to select (i.e., k). Figure 5c shows how drastically the collision probability reduces in the second round. Even with the increase in the number of contending nodes, it remains below 3%.

To accommodate the delay imposed by the second round,the *epoch* time will be extended to almost the double of its original time (i.e., when only single round of selection is used). Fortunately, in the typical scenarios of *Harmony*, it is quite unlikely to have many contending nodes (e.g., more than 10 nodes). Therefore, the probability to use a second



(a) (b) (c) (d)

Fig. 5: a)**The values of the ΔT for collision and no-collision scenarios.** b) **Effect of adjacent frequencies in calculating the cross-correlation.** c) **The comparison of collision probability between Wi-Fi backoff and, the First and Second round of the *Harmony*.** d) **Throughput gain with *Harmony* over Wi-Fi backoff for different number of winning nodes (k).**

selection round is very low. Figure 5c shows that the collision probability using only single selection round is similar to traditional Wi-Fi collisions for networks of 10 nodes or less.

Figure 6 shows the complete flow chart of *Harmony* including the optional second round selection. We see that from the first round, if $rank \leq k$ and a collision is detected by a node, it selects a new random tone for the second round. Otherwise, if no node detects any collision, the second round of selections is avoided. Each selected node from the first round generates a new random tone in the second round. Then, each selected node calculates its new $rank$ and updates its $tonenumber$ according to algorithm 2. Finally, the algorithm selects the final nodes and determines their corresponding ranks. Then, similarly to algorithm 2, each selected final node uses their rank to interact with the Wi-Fi interface for the data transmission. Note that, nodes with $rank > k$ in the first round, don't participate in the second selection round and wait for the next epoch to participate.

V. Implementation

In this section, we describe the implementation of the Acoustic-WiFi framework for the smart devices. Acoustic-WiFi framework consists of the acoustic interface, the WiFi interface, and the interactions between the different layers of the two interfaces. The acoustic interface in this framework has two layers: 1) A-PHY layer and 2) A-MAC layer. In addition, the framework defines the interaction between the Wi-Fi MAC and upper (e.g., TCP/IP and applications) with the acoustic interface. Figure 7 shows the detail architecture of the implemented Acoustic-WiFi framework for our *Harmony* scheme in smart devices.

A. Acoustic Interface

Figure 7 shows the standard architecture of the acoustic interface(left), the WiFi interface(right) and the additional modules of the Acoustic-WiFi framework(shaded boxes). We found that the acoustic interface in commodity smart devices has three main components i) Codec Driver, ii) Platform Driver, and iii) Machine Driver. In order to implement any of these drivers, there are standard set of struct operation or function pointer API that needs to be implemented [1], [10].

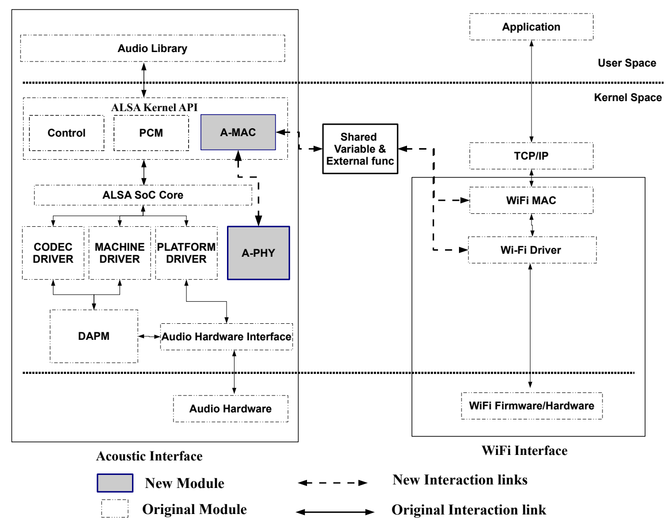


Fig. 7: **Acoustic-WiFi architecture implementation in Smart devices**

1) A-PHY Layer: The A-PHY module in figure 7, is a platform driver that is responsible for the DMA data transfer between the audio driver and the audio hardware. In addition to that A-PHY also implement the function `generateTone(tone)` to generate the *acoustic tone*. We pre-allocate the 26 defined tones in memory using flag `GFP_ATOMIC`. Note that, the Acoustic-WiFi framework generate the *acoustic tone* from the kernel space, therefore it significantly reduces the delay of actually emitting the tone from the speaker after the node sends the command to generate the signal. A-PHY applies low order Finite Impulse response (FIR) high pass filter on the captured raw acoustic signal(i.e. PCM) to diminish low frequency noise. A-PHY layer also uses a parallel cross-correlation technique with the allocated tones to detect the frequency of the captured *acoustic tone*.

2) A-MAC Layer: A-MAC is a more sophisticated module that is implemented as a part of the audio library/API in kernel space. A-MAC directly interacts with the A-PHY module to generate an *acoustic tone* of certain *tone number* or to receive the *tone number* from the captured *acoustic tone*. A-MAC runs a continuous loop of our *Harmony* scheme similar to the flow chart 6. A-MAC layer also interacts with Wi-Fi driver to get the notification of the `PIFS` to update the $rank$ of the node that is selected for the data transmission. While the $rank$ turns to zero A-MAC enables a flag in Wi-Fi driver that allows the Wi-Fi interface to transmit data frame from the driver buffer. After successful or unsuccessful transmission Wi-Fi driver disables the flag and notifies it to the A-MAC. In unsuccessful data transmission, Wi-Fi driver keeps the data frame in the buffer; therefore the data frame gets the chance to be retransmitted in the next transmission slot according to its order. In addition, the Wi-Fi interface also notifies `DIFS` to the A-MAC to indicate the beginning of the new *epoch* period.

In calculating the *epoch* period, we consider the propagation time, the length of the *acoustic tone* and the computation time for detecting *acoustic tone*. We use 100 msec as our minimum *epoch* period requirement. In indoor environment, the maximum distance, a node can hear the acoustic tone from a distance of 25 meters. Which is also the typical range for Wi-Fi in indoor environment [8]. The acoustic signal requires 75 msec to propagate the distance of 25 meters, which is less than 100 msec.

In Wi-Fi MAC, we configure both the CW_{min} and the CW_{max} to zero to disable the backoff scheme.

VI. Evaluation

Real TestBed: We evaluated *Harmony* using real testbed, where we implemented the scheme on Android devices. In the testbed, we use 10 LG Nexus 4 smartphones and a laptop as an Access Point (AP) in which the wireless interfaces of both phones and the laptop are running 802.11n standards. We modified the Android kernel of the smartphones to implement our Acoustic-WiFi and Harmony on the smartphones. We setup our testbed in a large conference room of size 16 m x16 m with many chairs and tables. Note that, there were background noises of running computer machines, air conditioners, footsteps, and conversations. We placed the phones on top of a large circular conference table arranged in a circle with a maximum distance between phones of 10 meters, which is less then the maximum distance of 20 meter that *Harmony* scheme can support. We run `iperf` client on each Android device and `iperf` server on the laptop. Note that, all Android phones are associated with the AP. We placed both the AP (i.e., the laptop) at the middle of the circular conference table.

Figure 5d shows the distribution of the overall throughput gain of *Harmony* over Wi-Fi for different number of selected k winning nodes. For each setting, we run the experiment for more than 50 times in which we calculate the average throughput gain for each run time. In Figure 5d, we observe that increasing the number of winning nodes increases the overall throughput gain. For example, with a selection of 6 winning nodes, we get more than 40% throughput gain for almost 40% of the time (i.e., runs) while this 40% gain drops to less than 10% of the time when we select 2 winning nodes instead. Note that with Wi-Fi high data rates (e.g., 802.11n), increasing the number of winning nodes will increase the throughput gain. However, selecting many winning nodes will not always lead to an increase in the overall throughput. A higher number of winning nodes will eventually increase the chance of collisions, thus it will negatively impact the overall throughput. Giving the small scale of the testbed, we use a simulation in the next subsection to evaluate the scheme performance under high dense networks.

Figure 8a evaluates the impact of the *epoch* time on the throughput gain. In this evaluation, we select 6 winning nodes for the three different *epoch* times. We found that *epoch* time has almost no impact on the throughput gain. We evaluate the *epoch* time up to 200 msec, which is enough time for the contending node to select the winning nodes over the acoustic channel.

Dense Networks Simulation: To investigate the scalability of our scheme with more contending nodes, we run our scheme through ns-3 simulation environment [2] running 802.11g standard. In simulation, we evaluated the Single Acoustic Domain scenario, where nodes were placed randomly in an area of 15 m x 15 m. Figure 8b shows the throughput gain over the Wi-Fi for different number of contending nodes. Note that we set the winning nodes selection parameter k to 6, and the *epoch* time to 200 msec in the simulation. We run the simulation over 250 times for each setting of contending nodes. Figure 8b shows the mean and the variance of the overall throughput gain. As shown, we observe slight degradation in throughput performance gain with the increase in number of contending nodes due to the increase in collision probability. More specifically, while we get a throughput gain up to 37% for light networks, this gain drops to about 27% for dense networks.

Figure 8c shows the overall throughput gain over Wi-Fi backoff for different number of winning nodes. Note that we use 30 contending nodes in the simulation and 200 msec *epoch* time. Throughput gain increases with the increase in the number of the winning nodes and that number reaches 8 in which we start to see a slight degradation in the throughput gain. This supports our hypothesis that selecting many winning nodes will not always increase the overall throughput because of the increase in collision probability.

Fairness: We have seen clear throughput gain of the *Harmony* scheme compared to Wi-Fi standards. One of the important characteristics of the Wi-Fi backoff scheme is guaranteeing fairness through randomness. In *Harmony*, we also use similar randomness in selecting the multiple winning nodes. In order to evaluate the fairness of our scheme, we use Jain's fairness index based on the throughput obtained by each contending node in *Harmony*. Figure 8d compares the fairness index for both *Harmony* and Wi-Fi under different numbers of contending nodes. The plot shows that *Harmony* has a stable fairness index as in traditional Wi-Fi networks.

VII. Conclusion and Future Work

In this paper, we designed, implemented and evaluated a new contention resolution scheme *Harmony*, which addresses the overhead of the current 802.11 backoff scheme. In, *Harmony*, we leveraged the novel cross-interface framework Acoustic-WiFi, where we used the acoustic channel as an additional control channel in parallel with the Wi-Fi interface. As a proof of concept, we evaluated *Harmony* on a small testbed of smartphones as well as built a simulation to confirm the feasibility and the performance improvement of our scheme.

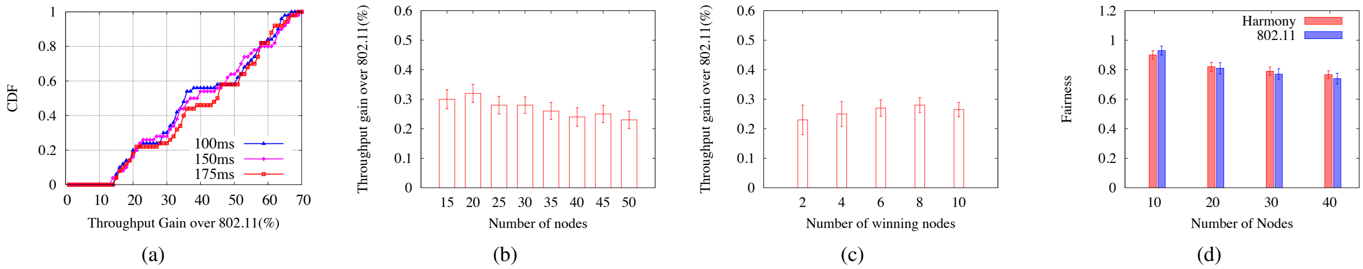


Fig. 8: a) **Throughput gain with *Harmony* scheme over Wi-Fi for different duration of *epoch* time.** b) **Throughput gain with *Harmony* scheme over Wi-Fi for different number of contending nodes in simulation.** c) **Throughput gain with *Harmony* scheme over Wi-Fi backoff scheme for different number of winning nodes (k). The total number contending node is 30 in the simulation.** d) ***Harmony* scheme shows almost similar fairness compare Wi-Fi backoff scheme for different number of contending nodes in simulation.**

However, there are several research challenges that are still remaining open and need to be addressed in our future work. For example, giving Acoustic-WiFi framework uses two separate mediums with different channel characteristics, there might be scenarios when two or more contending nodes are in the Wi-Fi range of each other but not in the acoustic range. In these scenarios, there is a chance to have **hidden nodes** with respect to the acoustic channel. For example, in figure 9, we have two Single Acoustic Domain(SAD)s, one consists of A & B nodes and the other consists of B & C nodes, where nodes A and C are not in the acoustic range of each other. We refer to such arrangement of nodes with multiple SADs as a Multi-Acoustic Domain (MAD).

In MAD, if two nodes generate the same *acoustic tone* and they are hidden to each other, it is challenging for nodes to detect the collision with respect to acoustic channel. As a result, collision might happen during the data transmission between the two nodes. For example,nodes A & C in Figure 9 could select the same *tone number* during an epoch time. Since the nodes are hidden to each other, both nodes will not be able to hear the acoustic tone of the other node and, consequent, a collision incident will be overlooked by both nodes. Our algorithm needs to be improved to address this MAD scenario challenge.

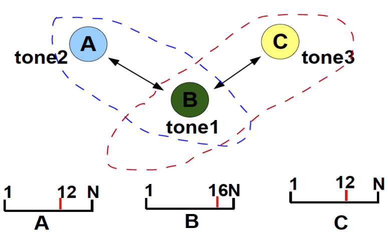


Fig. 9: **Scenario of Multi-Acoustic Domain**

In addition, *Harmony* is designed and evaluated without considering its impact on the legacy devices, which uses conventional 802.11 backoff. Since legacy devices don't use the acoustic channel, they could be considered as a hidden node for the *Harmony* enabled nodes.

For both MAD and Legacy device scenarios, *Harmony* increases collision over Wi-Fi channel. One simple scheme to address these issue is whenever the Wi-Fi collision frequency exceeds a certain threshold, a *Harmony* node could assume MAD and Legacy devices hidden node scenarios and automatically disable the *Harmony* scheme. Based on the plot in figure 5c, we set this threshold to 10%. In future work, we would like to consider other schemes to accommodate these hidden node scenarios.

References

[1] ALSA. http://www.alsa-project.org.
[2] ns-3. http://www.nsnam.org/.
[3] Android smartphone users download just 870 Mb over cellular per month., 2012. http://tinyurl.com/p7doszj.
[4] Y. Agarwal et al. Switchr: Reducing system power consumption in a multi-client, multi-radio environment. In *IEEE ISWC*, 2008.
[5] G. Ananthanarayanan and I. Stoica. Blue-fi: Enhancing wi-fi performance using bluetooth signals. ACM MobiSys, 2009.
[6] A. Balasubramanian et al. Augmenting mobile 3g using wifi. ACM MobiSys, 2010.
[7] V. Bharghavan et al. Macaw: A media access protocol for wireless lan's. ACM SIGCOMM, 1994.
[8] Y. Chen et al. Fm-based indoor localization. ACM MobiSys, 2012.
[9] M. Chitre et al. Underwater acoustic communications and networking: Recent advances and future challenges. *Marine Technology Soc. J*, 2008.
[10] S. Dimitrov and S. Serafin. Minivosc-a minimal virtual oscillator driver for alsa (advanced linux sound architecture). *Not published*, 2011.
[11] J. Fang et al. Fine-grained channel access in wireless lan. *IEEE/ACM Trans. Netw.*, 2013.
[12] Y. He et al. Semi-random backoff: Towards resource reservation for channel access in wireless lans. *IEEE/ACM Trans. Netw.*, February 2013.
[13] A. P. Jardosh et al. Understanding congestion in ieee 802.11b wireless networks. ACM IMC, 2005.
[14] T. Jin et al. Wizi-cloud: Application-transparent dual zigbee-wifi radios for low power internet access. IEEE INFOCOM, 2011.
[15] R. Jurdak, C. Lopes, P. M. Q. Aguiar, and P. Baldi. A comparative analysis and experimental study on wireless aerial and underwater acoustic communications. IEEE ICDT, 2006.
[16] P. Lazik and A. Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. ACM SenSys, 2012.
[17] C. Lim and C.-H. Choi. Tdm-based coordination function (tcf) in wlan for high throughput. In *IEEE GLOBECOM*, 2004.
[18] H. Liu et al. Push the limit of wifi based localization for smartphones. ACM MobiCom, 2012.
[19] C. V. Lopes and P. M. Q. Aguiar. Acoustic modems for ubiquitous computing. *IEEE Pervasive Computing*, 2003.
[20] R. Nandakumar et al. Dhwani: Secure peer-to-peer acoustic nfc. ACM SIGCOMM, 2013.
[21] C. Peng et al. Beepbeep: A high accuracy acoustic ranging system using cots mobile devices. ACM SenSys, 2007.
[22] T. Pering et al. Coolspots: Reducing the power consumption of wireless mobile devices with multiple radio interfaces. ACM MobiSys, 2006.
[23] I. Rhee et al. Z-MAC: a hybrid MAC for wireless sensor networks. ACM SenSys, 2005.
[24] S. Sen et al. Listen before you talk, but on the frequency domain. *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, 2010.
[25] S. Sen et al. No time to countdown: Migrating backoff to the frequency domain. ACM MobiCom, 2011.
[26] M. Uddin and T. Nadeem. A2psm: Audio assisted wi-fi power saving mechanism for smart devices. HotMobile '13. ACM.
[27] M. Uddin and T. Nadeem. Rf-beep: A light ranging scheme for smart devices. 2013.
[28] J. Yang et al. Detecting driver phone use leveraging car speakers. ACM MobiCom, 2011.
[29] A. Yarali and B. Ahsant. 802.11n: The new wave in wlan technology. ACM Mobility, 2007.
[30] Z. Zhang et al. Swordfight: Enabling a new class of phone-to-phone action games on commodity phones. ACM MobiSys, 2012.