Ambient Rendezvous: Energy-Efficient Neighbor Discovery via Acoustic Sensing

Keyu Wang*†, Zheng Yang‡, Zimu Zhou*, Yunhao Liu‡, and Lionel Ni*†

* Department of Computer Science and Engineering
† HKUST Fok Ying Tung Research Institute
Hong Kong University of Science and Technology
‡ School of Software and TNList, Tsinghua University
Email: {kwangad, ni}@cse.ust.hk, {yang, zhouzimu, yunhao}@greenorbs.com

***Abstract*—The continual proliferation of mobile devices has stimulated the development of opportunistic encounter-based networking and has spurred a myriad of proximity-based mobile applications. A primary cornerstone of such applications is to discover neighboring devices effectively and efficiently. Despite extensive protocol optimization, current neighbor discovery modalities mainly rely on radio interfaces, whose energy and wake up delay required to initiate, configure and operate these protocols hamper practical applicability. Unlike conventional schemes that actively emit radio tones, we exploit ubiquitous audio events to discover neighbors passively. The rationale is that spatially adjacent neighbors tend to share similar ambient acoustic environments. We propose AIR, an effective and efficient neighbor discovery protocol via low power acoustic sensing to reduce discovery latency. Especially, AIR substantially increases the discovery probability of the first time they turn the radio on. Compared with the state-of-the-art neighbor discovery protocol, AIR significantly decreases the average discovery latency by around 70%, which is promising for supporting vast proximity-based mobile applications.**

I. Introduction

Rapid development in fabrication and ubiquitous computing technologies has given birth to numerous applications that leverage personal mobile devices, such as smartphones and tablets, to meet people's entertainment and work demands. In particular, Nintendo StreetPass [1] and PlayStation Vita [2] are widely used for mobile gaming, and Nextdoor [3] provides a private mobile social network for users' neighborhood community. Some reports [4] [5] also indicate a bright future for the mobile computing industry.

Meanwhile, more and more mobile applications are providing proximity-based services for geographically co-located people. When people attend special events or are at common locations, they become spatially close and may seek to exploit opportunistic contacts with each other. Zhang et al. indicate that there exists a strong need for direct connection among co-located people in their work [6]. Thus, discovering neighbors effectively and efficiently is crucial for the success of the application and the user experience. Considering privacy issues, the availability of internet-access and the property of proximity-based services, neighbor discovery is better completed by having devices in the vicinity to interact with each other directly instead of matching online. The most effective way of neighbor discovery is certainly to continuously search for neighbors. This however is not affordable as mobile devices run on battery and encounters are opportunistic and unpredictable. A more feasible solution is to operate in low duty cycle mode, which means periodically waking the wireless interface (e.g., WiFi and Bluetooth) up to perform discovery and then keeping it asleep most of the time.

Under such a low duty cycle mode, the success of discovery depends on the existence of an overlap between the wakeup time of neighboring devices. Synchronizing the clock via GPS [7] or NTP servers [8] could easily complete such a task, but they are not power-efficient nor ubiquitously available. As a result, without the time synchronization, it is challenging to ensure the overlap between radio-on time of neighbors by distributed scheduling in low duty cycle mode. Lots of existing asynchronous discovery protocols [9]–[12] address this challenge and improve the discovery latency. However, they still require a dozen periods for discovery while we prefer to discover neighbors in one or two periods.

We are enlightened by the fact that neighbors are spatially close, so they share similar ambient information, such as illumination intensity, temperature, humidity, acoustic sound, and radio signals. We could exploit this information to complete neighbor discovery in only one or two periods. With considerations of availability, accuracy, and energy consumption of sensors, we choose microphone, one of the most important, ubiquitous and energy efficient components on mobile devices, to acquire ambient acoustic information. Many measurement reports show that microphones are energy efficient as it consumes less than one fifth of the energy of WiFi interface [13], and less than half of the latest Bluetooth [14]. Besides, acoustic information is ubiquitous and rich in information as many well-known applications successfully utilize acoustic signals for localization [15] [16], device pairing [14] [17], distance measurements [18] and counting [19]. Furthermore, the correlation of acoustic features between neighbors is high as we show in Section IV. Therefore, we propose AIR (from **A**mb**I**ent **R**endezvous), an effective and efficient neighbor discovery protocol leveraging ambient acoustic information.

AIR is a fully passive design that only records ambient

978-1-4799-8381-0/15/$31.00 ©2015 IEEE

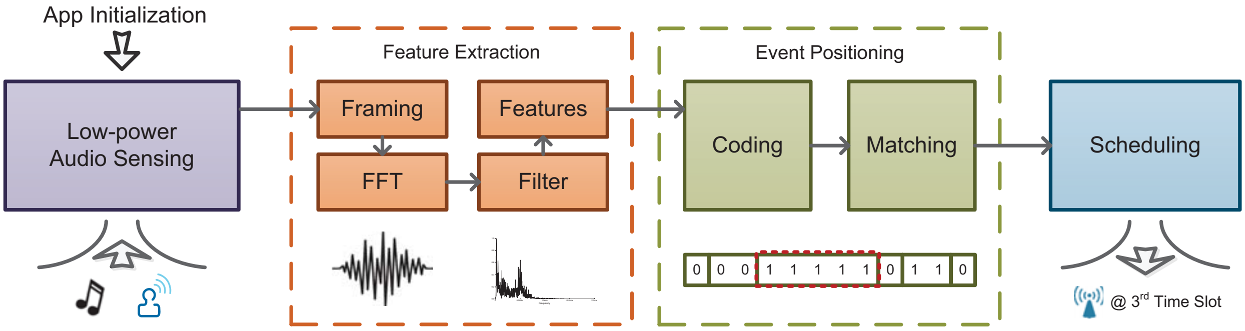


Fig. 1. System Architecture.

acoustic information at a low rate. Every sample record is short in length without emitting any tones since a passive design never interferes with users and the environment. Therefore, it is widely applicable and preferable. To precisely detect ubiquitous and various audio events, such as voice, music and noise, AIR extracts spectral entropy and flux after low power audio sensing. Then, among a series of event points, we design a novel coding and matching algorithm to ensure neighbors select the same audio event without negotiation. At last, AIR decides its radio-on schedule according to the selected event position. The performance of AIR is evaluated through extensive experiments. AIR achieves around 90% discovery rate the first three times the radio is turned on, and compared with the state-of-the-art neighbor discovery protocol Searchlight [12], AIR significantly decreases the discovery latency by around 70% on average.

Major contributions are as follows:

- **The first neighbor discovery protocol utilizing ambient acoustic information.** AIR is neither probabilistic nor deterministic, while it adopts a novel coding and matching algorithm based on acoustic information to discover neighboring devices.
- **Fully passive design without emitting audio tones or extra hardware.** It is unnecessary for AIR to emit any tones such that it is more compatible with abundant future applications.
- **Extensive experiments under various environments** AIR is feasible under various environments, such as different background sounds, different rooms, indoors and outdoors, and so on.
- **Good performance with over 90% discovery rate in the first three periods.** The evaluation results show that AIR performs well in practical usage. A high success rate also verifies its feasibility.

We discuss the related work in Section II. Then, we introduce the system overview and design challenges in Section III, and detail our design in Section IV. Section V presents the performance evaluation. Finally, we conclude the paper and indicate our future work in Section VI.

II. Related Work

The design of AIR is closely related to the following two categories of research.

Neighbor Discovery Protocols: Most neighbor discovery protocols are time-slotted where time is partitioned into equal-length timeslots and each timeslot is assigned as an active or idle state as demanded. In general, existing protocols are from two main classes - probabilistic protocols and deterministic protocols.

Probabilistic schemes assign the nodes active/idle with a given probability, for example, "birthday protocol" [9]. When the density of devices is low, such a scheme demands a long discovery latency. On the contrary, deterministic protocols are suitable for low density situations even for only two devices. Quorum-based protocol [20] [21] bounds the worst-case latency by scheduling devices periodically with a global parameter. It promises the mutual discovery between neighbors as there exists at least two intersections. Disco [10] selects a pair of different prime numbers and successfully guarantees the worst-case bound as the product. Later, Kandhalu et al. further improve the metric of energy-latency product through another prime-based protocol called U-Connect [11]. Recently, Searchlight [12] proposes a new method that defines two types of active intervals: *anchor* (a fixed timeslot in each period) and *probe* (scanning timeslots of the period). This ensures that active intervals overlap between neighbors during the moving of *probe* intervals. Up to now, Searchlight achieves the shortest discovery latency. On the other hand, Acc [22] proposes an on-demand generic discovery accelerating middleware supporting both direct and indirect neighbor discoveries.

AIR is neither probabilistic nor deterministic, while it is a novel method leveraging environmental information for neighbor discovery problems.

Sound Applications: Since acoustic information is ubiquitous and the energy efficient acoustic sensors, such as the microphone, are widely equipped on devices, acoustic information is exploited in numerous applications, for example, sound classification [23], room-level localization [15] [16], paring [14] [17], distance measurement [18] and counting [19]. Part of these works proactively emit a high tone for detection or distance measurement. Mostly, they utilize the Doppler effect [24], TDoA [16] or EToA [17] to achieve their goals. In order to accurately detect the audio tone, they adopt some signal processing techniques [25] [26] for analysis. The other part of these works passively record acoustic information, and then analyze acoustic information in the time domain like sound level, or in the frequency domain like entropy, bandwidth, and so on to achieve individual goals . Different from them, AIR targets on a different issue that discovering neighboring devices and utilizes algorithms to achieve a shared purpose without emitting a tone.

III. Overview

The study of neighbor discovery problems focuses on supporting proximity-based applications, especially when users want to operate in a purely local fashion [12]. The key insight that inspires AIR is that, acoustic information are similar among neighboring devices in close vicinity, almost every kind of mobile devices, for instance, smartphones, tablets, iPods and PSPs, has microphones for recording sound and the microphone is an energy-efficient component of mobile devices compared with wireless modules. Some measurements indicate that microphones consume less than 300 mW power [27], which is one fifth that of WiFi interface [13]. Even the latest Bluetooth works at over 600 mW [14] twice microphone power. Consequently, some pioneering work [14] has explored the idea of utilizing microphones in novel energy-efficient mobile applications in substitution of wireless interfaces, and achieved 5.5X lower than the latest Bluetooth 4.0 protocols, respectively.

Also, by independently detecting and carefully selecting the same *audio events*, neighbors consent upon a pre-determined waiting time, and turn on the radio simultaneously for neighbor discovery. To codify the above idea into a practical system, AIR has to tackle the following major technical challenges.

- **Extracting Meaningful Acoustic Features and Detecting Unknown Audio Events.** Since microphones have no prior knowledge of the nature of the recorded audio, that is, whether it is a piece of melody or a segment of speech, we need generic acoustic features that characterize a wide range of ambient sounds. In addition, as we aim to detect representative audio events rather than uncertain background noise for radio-on scheduling, we need acoustic features that are noise-resistant and quantify how informative and meaningful the audio records are.
- **Distributed Consensus and Localization of Audio Events.** To ensure turn on their radios simultaneously, it is crucial for the devices to select the same audio event without negotiation. However, device diversity and location differences potentially degrade the quality of audio recording and induce unknown distortion of the extracted features, thus leading to different radio-on schedules.

Figure 1 shows the overall AIR architecture. At a high level, AIR addresses the above challenges by conducting the following steps:

1) **Low Power Audio Sensing.** After initiation, AIR enters a low power audio sensing state to save energy. It periodically opens its microphone, records a short duration of acoustic information and stores it for further analysis. The sampling frequency is tuned to capture a wide range of daily sound events, for example, walking, speaking, music, and so on.
2) **Framed Acoustic Spectrum Processing.** To extract appropriate features to detect audio events, AIR transforms each short duration of audio samples into the frequency domain, and calculates a sequence of informative and representative spectral features for audio event detection.
3) **Event Coding and Matching.** To mitigate potential event misalignments, the sequences of spectral features are coded into a single binary sequence, where candidate events are transformed into consecutive 1's. A maximum length matching scheme is then employed to ensure that independent neighbors select the same representative audio event with high probability.
4) **Radio-on Scheduling.** To fix the duty cycle of neighbor discovery, a radio-on schedule is assigned only after a certain amount of audio recording. Each device then locates the detected event position on the timeline and assigns radio-on states to the timeslots with a pre-defined delay to the selected event. Thus their radio-on timeslots are aligned as long as the selected events are identical.

These steps can be extended to other environmental sensing methods as a framework. When mobile devices are widely equipped with a more energy efficient module, we are able to adopt such a framework to those devices easily. Next, we elaborate our design in detail in the following section.

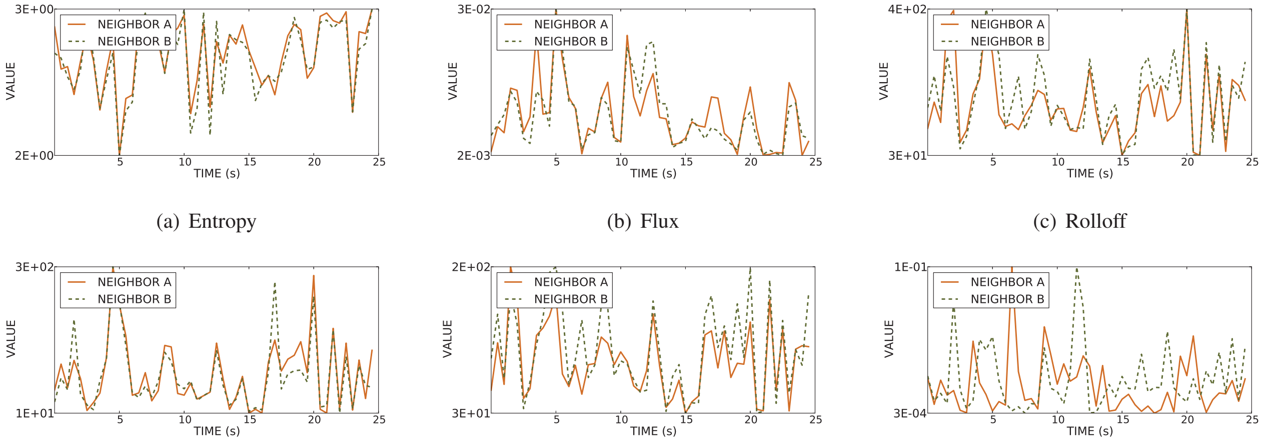
IV. System Design

A. Low-power Audio Sensing

The rationale for employing lower-power audio sensing is twofold:

- **Ubiquitous Support:** Microphones are pervasively embedded in commodity mobile phones, tablets, laptops, and so on. Therefore no extra infrastructure is needed to operate the AIR protocol.
- **Energy Efficiency:** Audio interfaces usually consume low energy [13]. In addition, instead of continuously collecting acoustic information, we adopt a low-power listening strategy to further reduce unnecessary energy consumption.

Unlike sensor radios, mobile wireless module (e.g., WiFi and Bluetooth) require much longer time (a few seconds) to bring the wireless interface up [12]. Therefore, we are unable to set the length of the timeslot as small as on sensor nodes, which employ ZigBee [28] for communication. Here we assume the length of a timeslot is 2 seconds and this is also the default setting in our experiments. In the future, if the latency for starting a wireless interface is shortened, we



(a) Entropy

(b) Flux

(c) Rolloff

(d) Centroid

(e) Bandwidth

(f) NWPD

Fig. 2. Feature Values between Two Neighbor Devices.

could alter the parameters of our design accordingly to keep it feasible. To capture common audio events with minimal energy consumption, we set the audio sensing frequency at 2Hz and the frame size as 2048 sample points to shorten the sound recording duration without degrading the audio content stability [29] [30]. Hence AIR can capture most sound events such as speaking, cars beeping and screaming. Given an audio sampling rate of 44.1KHz, this is equal to around 40ms of recording, and consequently, the audio interface is working on quite a low duty cycle. From our experiments on lower sampling rates, AIR allows the rate to be lowered down 8K for greatly energy saving.

Although such low sensing frequency and short sampling duration may capture transient or even noisy audio events, the spectral features we extracted (Section IV-B) and the maximal length based event matching scheme (Section IV-C) ensure that only notable and lasting audio events are selected and aligned for radio-on scheduling.

B. Framed Acoustic Spectrum Processing

To detect audio events that are representative and noise-resistant, we conduct framed acrostic spectrum analysis on a sequence of audio frames, and extract appropriate spectral features from each frame. Audio events are then detected from a sequence of spectral feature points. In this section, we explore candidate spectral features and select robust features for audio event generation.

1) Candidate Features: Denote $f_i(u)$ as the normalized magnitude of the u^{th} frequency bin of the spectrum over the i^{th} audio frame via an M-point FFT. Several prevalent spectral features from the acrostic signal processing literature are enumerated below.

Spectral Entropy

$$H_i = -\sum_{u=1}^{M} f_i(u) \log f_i(u) \qquad (1)$$

TABLE I. SPATIAL CORRELATIONS OF FEATURES

Entropy	Flux	Rolloff	Centroid	Bandwidth	NWPD
0.843	0.803	0.801	0.824	0.796	0.029

Spectral entropy characterizes the flatness of the acoustic spectrum shape [29]. A high entropy indicates a flat spectrum while a low entropy usually represents certain notable spectrum patterns. Thus, when an audio event occurs, entropy tends to drop sharply, since the audio event possibly dominates the acoustic environment and exhibits a specific spectrum pattern with respect to a relative flat spectrum.

Spectral Flux

$$SF_i = \sum_{u=1}^{M} (f_i(u) - f_{i-1}(u))^2 \qquad (2)$$

Spectral flux [31] quantifies the shape change of two successive acoustic spectra, which is defined as a L_2-norm of the amplitude difference vector of two adjacent frames. When flux increases substantially, it indicates a significant change of the ambient sound, which is potentially induced in an audio event.

Spectral Rolloff Frequency

$$SRF_i = \max(h | \sum_{u=1}^{h} f_i(u) < TH \cdot \sum_{u=1}^{M} f_i(u)) \qquad (3)$$

Spectral rolloff frequency [32] is designed to measure the skewness of the spectral distribution. A large value indicates a right-skewed distribution, for example, music signals which contain a large series of high frequency components. In practice, however, it is hard to set an appropriate threshold TH.

Spectral Centroid

$$c_i = \frac{\sum_{u=1}^{M} u \cdot |f_i(u)|^2}{\sum_{u=1}^{M} |f_i(u)|^2} \qquad (4)$$

Spectral centroid [32] calculates the balancing point of the spectral power distribution. Normally, different centroid values mark different categories of audio events. For instance, music with high frequency sounds, screaming and scratching usually push the spectral centroid high, while the centroid of a human voice may be relatively low given the same frequency range. In a quiet ambient environment, however, the balancing point may fluctuate due to random noise over a relatively flat spectrum, thus posing ambiguity in detecting different kinds of audio events.

Bandwidth

$$b_i^2 = \frac{\sum_{u=1}^{M}(u-c_i)^2 \cdot |f_i(u)|^2}{\sum_{u=1}^{M}|f_i(u)|^2} \tag{5}$$

Bandwidth [32] depicts the concentration of a spectrum around its centroid. A flat spectrum normally has a high bandwidth. Meanwhile, for music spanning a broad frequency range, it also has a large bandwidth. Hence, bandwidth suffers similar limitations to spectral centroid.

Normalized Weighted Phase Deviation

$$nwpd_i = \sum_{u=1}^{M} f_i(u) \cdot \phi_i'' \tag{6}$$

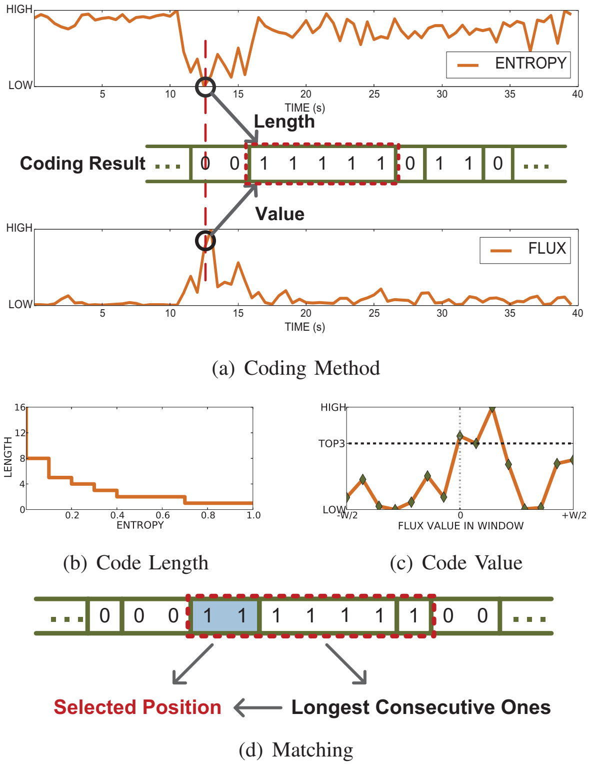
where ϕ'' is the second derivative of the phase of u^{th} frequency bin. This feature expresses the phase deviations weighted by their magnitude and both ambient sound and music usually have a small value of this feature [33].

The above features characterize different aspects of acoustic information. In terms of neighbor discovery, an effective feature should have a high correlation among neighboring devices even if the audio samples are recorded independently. Fig. 2 plots an example of the above features obtained from the audio samples recorded by two devices placed 10m apart, and TABLE I calculates the average spatial correlation coefficients of different spectral features under 3 different ambient acoustic environments, for example, playing music, speaking and walking.

As shown, in most cases, spectral entropy has the highest correlation coefficient compared with other features and it indicates the amount of acoustic information. Except for entropy, we select flux as another candidate feature because it represents the change in spectrum shape. When an audio event happens, the shape of the spectrum is definitely different from the earlier one, which introduces a large flux value. So that the flux value is more appropriate for detecting the same audio event. Taking the physical significance and limitations of different features into consideration as well, we finally pick spectral entropy and flux as the candidate features and propose combining them via a novel coding scheme for audio event detection and localization.

2) Feature Extraction: As illustrated in Fig. 1, each audio frame of 2048 samples has been obtained via a period of low power audio sensing, and is converted into a corresponding spectrum by FFT. The spectrum is then filtered by a bandpass filter for the following reasons.

- High frequency tones tend to experience strong energy attenuation [14], which potentially creates audio quality deviation.
- Common audio events seldom contain signals of too low frequencies.



(a) Coding Method

(b) Code Length

(c) Code Value

(d) Matching

Fig. 3. Coding and Matching Method. a) Coding Overview b) Code Length Function c) Code Value Decision d) the Longest Consecutive 1-sequence Locates the Event Position

We empirically select a pass band of 500Hz to 11kHz in our design. After employed to calculate spectral entropy and flux as in Eqn. 1 and Eqn. 2, we obtain two series of entropy and flux values from a sequence of audio frames. For complexity and energy-efficient considerations, in this process, we adopt FFT on such small length of frames with some other simple calculations, and each frame has no overlap with another. Therefore, it will not cause any stall and only consumes little energy for the computation.

C. Event localization

The event localization module is tasked to locate the audio events from the two sequences of spectral entropy and flux values to ensure that the devices that detect the audio events can independently select the same audio event. Under ideal conditions, this can be achieved by simply choosing the maximal or minimal point in a certain sequence of acoustic spectrum features, since different devices close to each other tend to record quite similar audio samples. In reality, however, hardware constraints, device diversity and spatial differences usually induce unknown deviation to the audio records, and hence the above event positioning scheme simply fails. To overcome the above challenges, we propose a novel audio event positioning scheme which consists of a coding procedure and a matching algorithm as illustrated in Fig. 3.

1) Event Coding: At a high level, our scheme codes the patterns of entropy and flux into a variable-length binary sequence. Each feature point of entropy and flux is transformed to a number of 1s or 0s. More specifically, the entropy decides the length while the flux determines whether the bit is coded into 1 or 0.

Code Length: As previously discussed, entropy quantifies the amount of information contained in the acoustic spectrum. A large entropy indicates a flat spectrum, which is probably ambient noise, while a small one suggests a potential audio event. Therefore, we set a long code length to a small entropy to magnify such a difference, and instead of a linear assignment, we employ the following scheme to highlight the audio event.

As shown in Fig. 3(b), we regulate the sequence of entropy values to 10 bins of equal length as $\frac{E_{max}-E_{min}}{10}$, and translate the entropy values to the integers corresponding to the index of bins which it falls into. In order to amplify the influence of low entropy values and limit the computation complexity, we adopt an inverse proportional function to the sequence of index integers as $L = \lfloor \frac{\alpha}{I_i} \rfloor$. From our experiment experience, choosing $\alpha = 16$ performs well.

Hence, the maximum length is equal to the integer α and when $I_i > k = 4$, it is no longer than 0.25α, which is much shorter than α. Therefore, the meaningful part of the samples is substantially emphasized and amplified into the coding result.

Code Value: After determining the length of coding for each feature point, the next question is to decide the coding value. Since our goal is to highlight the happening of audio events, we should keep the number of code bit 1 as a small amount. Referring to Fig. 3(c), for each feature point, we check whether it is in the top-3 of flux values within a fixed window of size 15. The choosing of these parameters comes from a series of tests. If it belongs to the top-3, we assign it as 1, otherwise 0. Theoretically, within a period length of L, $\frac{L}{5}$ feature points would be assigned as 1 in expectation.

2) Event Matching: Now, we obtain a binary sequence with a varied length, and then we search for the longest consecutive 1s to set their starting point as the selected position. Based on the above coding scheme, each sequence of successive 1s may be the result of either a single feature point or multiple feature points. Such a matching scheme is applicable to both cases.

- **Single Feature Point Case:** The output of the matching scheme is actually the starting point of a strong audio event, since it is the longest sequence containing only one feature point.
- **Multiple Feature Points Case:** When multiple feature points are integrated into a single long sequence of 1s, it indicates that the audio event is lasting and continuously changing. The starting point of this series of changing events is thus output from the matching scheme.

In summary, the matching outputs the start position of a significant audio event. Note that the above matching scheme can be naturally extended to output multiple start positions of several significant audio events as the 2nd and 3rd candidates. This is useful when the duty cycle is high, and hence we need to assign more radio-on timeslots.

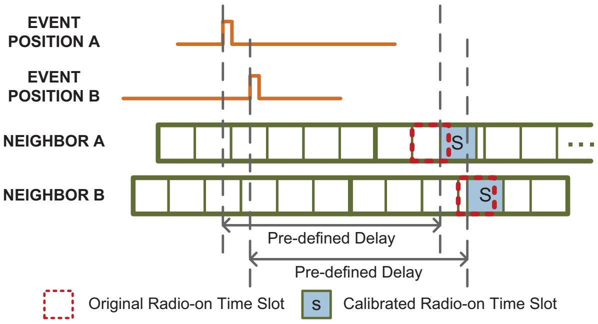


Fig. 4. Scheduling Example.

D. Radio-on Scheduling

In order to guarantee the low-duty cycle operation, AIR locates specified number of audio events in each period according to the duty-cycle. Such step operates period by period. Thus, after successfully locating the same audio event, neighboring devices finally turn on their radios for discovery after a pre-defined delay, which equals to the length of the period. We first discuss two important parameters, for example, the length of one timeslot and the predefined delay , before elaborating on the scheduling scheme in detail.

1) Length of Timeslot: In sensor node based protocols, the length of timeslots is usually set at the scale of milliseconds [10] or even microseconds [11]. Nevertheless, a timeslot usually lasts much longer for mobile device based protocols, due to the intrinsic constraints of their relatively long wireless interface switch time. In AIR, we adopt a timeslot length of 2 seconds, which is consistent with that in mainstream mobile device based protocols [12]. As we discussed in Section IV, if the latency for starting a wireless interface is shortened, AIR is still workable by easily changing some parameters correspondingly. Based on such a setting, AIR achieves a significantly smaller average discovery latency than existing deterministic protocols.

2) Pre-defined Delay: AIR sets the pre-defined delay equal to the length of the period, and we configure the period length as 20 timeslots based on the following considerations.

- **Latency:** For rapid neighbor discovery, the earliest radio-on timeslot the devices can select is in the next period. A large period will lead to long waiting time, which is not preferable.
- **Duty Cycle:** A period length of more timeslots provides more flexibility for users to select the desired working duty cycles, as well as a lower minimal duty cycle.

With a period length of 20 timeslots, AIR could retain the same low duty cycle of 5% and provide reasonably amounts of duty cycle choices in practical usage.

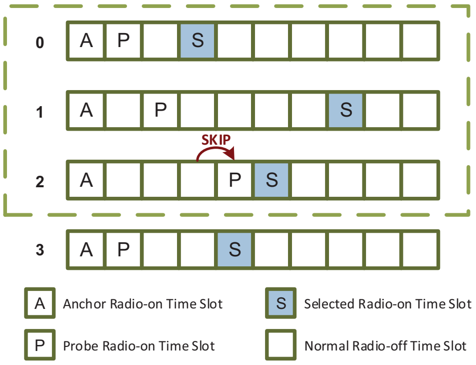

0
A
P
S
1
A
P
S
SKIP
2
A
P
S
3
A
P
S
A Anchor Radio-on Time Slot
S Selected Radio-on Time Slot
P Probe Radio-on Time Slot
Normal Radio-off Time Slot

Fig. 5. Combination with Searchlight.

After deciding the pre-defined delay, we elaborate on the detailed radio-on scheduling as follows.

3) Scheduling Scheme: When an audio event located at position x is selected, AIR re-calibrates the radio-on timeslot starting from the position with a pre-defined delay (which is one period length in our scheme) to x in the next period, and lasting for the same length of one timeslot. Recall that the sample frequency is 2Hz while each timeslot occupies 2 seconds. Therefore after re-calibration, the scheduled radio-on timeslots promise overlaps for more than 0.5s, which is sufficient for discovering neighboring devices, as long as the selected event positions of different devices deviate within 1.5s (See Fig. 4 for an example). When the length of a timeslot is shortened, we could slightly increase the sample frequency correspondingly. In effect, the re-calibration successfully remedies some potential failure cases due to the misaligned timeslot and the offsets of the selected event positions of different devices (e.g., the red dotted boxes in Fig. 4). To support different working duty cycles in AIR, devices still adopt the same lengthed period but select different quantities of radio-on timeslots according to their desired duty cycles.

4) Combination with Deterministic Protocol: Note that our assignment of radio-on timeslots can be regarded as a special kind of probe timeslots. It is thus convenient to incorporate AIR with prevalent deterministic protocols such as Searchlight [12], which provides a worst-case bound for discovery latency. Fig. 5 illustrates an example of such an incorporation. As is shown, by combining AIR with Searchlight, the worst-case latency bound is also shortened. This is because when the probe slot's position has been occupied by a selected radio-on timeslot of AIR in a certain period before, the next probe slot could skip this position according to the purpose of probe slots (See the third period in Fig. 5).

V. Evaluation

In this section, we evaluate the performance of AIR in various scenarios.

A. Methodology

We employ several LG Nexus 4 mobile phones to sense ambient acoustic information. All devices sense acoustic information at the sampling rate of 44.1KHz and use WiFi interfaces for discovering with the duty cycle of 5%. Other parameters are chosen the same as we state in Section IV. Since AIR can not guarantee a worst-case discovery latency, it has a long tail to achieve 100% discovery ratio. For all following comparisons, the failure of neighbor discovery is regarded as not happening within the first three periods of scheduling.

For the purpose of verifying the feasibility and robustness of AIR, we conduct experiments in different scenarios and repeat them hundreds of times for each scenario. Due to the trade-off between discovery latency and duty cycle, and AIR is the first one utilizes environment information, we compare the discovery latency between AIR and the state-of-the-art deterministic protocol Searchlight [12] at the same duty cycle. Similar to other related works, we adopt the CDF (Cumulative Distribution Function) of discovery latency to exhibit their performances. It is worth mentioning that in order to precisely display the performance of AIR, we do not make any schedule for the first period as AIR requires a period of acoustic information recording (see Fig. 6(a)). In practice use, the first period can be scheduled by random selection or kept always-on. Both options will not affect the duty cycle in long term as they only occur once.

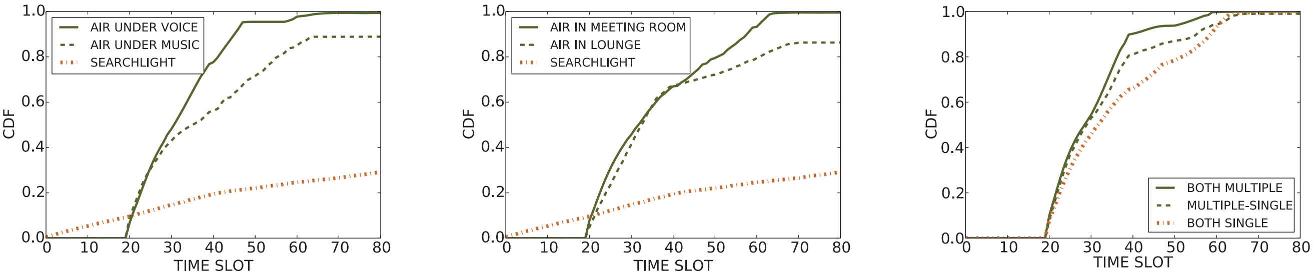
In early stage of the experiments, we have discovered that when an audio event suddenly breaks the silence, AIR can detect the event accurately and achieve a discovery rate of almost 100% in first radio-on period. On the other hand, discovering neighbors is usually required in places full of social activities, where are not very quiet and mixed with different kinds of sound. Thus, all following experiments are conducted in the environment of various sound types and random noise levels, which simulates the practical usage.

B. Discovery Performance in Different Scenarios

1) Different Ambient Sounds: We evaluate different dominating ambient sound scenarios for indoor environment such as offices and coffee shops. Inside rooms, the most common sounds are human voice and music, for example, people talk and chat with other and coffee shops play light music as background music. Thus, we classified the ambient sounds into these two types, and display the performance of AIR as two sets: voice and music(see Fig. 6(a)).

In human voice dominated sound settings, AIR achieves nearly 99% success rate, with most of the discovery taken place in the first radio-on period. In comparison, in music dominated sound environment, AIR also succeeds with a discovery rate of around 90%. The discovery rate is lower in music environment since the frequency spectra of music tend to be more complex than that of voice. As microphones on mobile devices are designed to better capture human voice, they generally have better voice detection than music detection.

2) Different Room Sizes: We then evaluate the performance of AIR in different sized rooms, that is, meeting room (3m × 5m) and lounge (10m × 15m). Both human voice and music scenarios are tested in these rooms. Neighboring devices are placed at each end of the room.



(a) Different Ambient Sound

(b) Different Room Sizes

(c) Multiple Selection of AIR

Fig. 6. Discovery CDF in Different Scenarios.

Undeniably, the success rate of experiments in the bigger room is lower (see Fig. 6(b)). Nevertheless, within the first three periods of scheduling, ratio of discovery is still over 85%. We need to clarify that since there is a TV set available in the lounge, we use it to play music instead of using the loudspeaker of a laptop. Hence, the sound quality of music in the lounge room is much better that in the meeting room, which help improve the discovery rate in the lounge case. In summary, AIR's performance in a large room is comparable with that in a small one, which verifies its ubiquity in different sized rooms as well as over different distances.

3) Multiple Active Slots Selection: We further test for multiple radio-on timeslot selection indoor. Specifically, we compare the results among three conditions: 1) both devices select a single radio-on timeslot during each period, 2) one of the devices selects a single radio-on timeslot while the other selects two, and 3) both devices select two radio-on timeslots. Selecting one and two radio-on timeslots indicate 5% and 10% of the duty cycle respectively.

Naturally, increasing the number of selected radio-on timeslots speeds up the discovery rate to some extent, which is also confirmed by our experiment results show (see Fig. 6(c)). This is consistent with our analysis in Section IV. To be specific, when both devices select two audio events for a radio-on schedule, around 90% of discoveries occur in the first period of scheduling; and when two devices select two and one respectively, the discovery rate is also over 80% (see Fig. 6(c)).

The discovery ratios of the aforementioned five scenarios are summarized in Fig. 7. Overall, AIR archives a 90% success rate of discovery, which is concentrated on the first radio-on scheduling. In terms of energy consumption, AIR consumes slightly more energy than Searchlight. Under the same low duty cycle, AIR and Searchlight takes 15% and 13% of total battery power respectively after multiple-hour experiments. The additional energy is mainly used for low-power audio sensing. Considering the large latency improvement by nearly 70%, AIR is more energy efficient in practical usage.

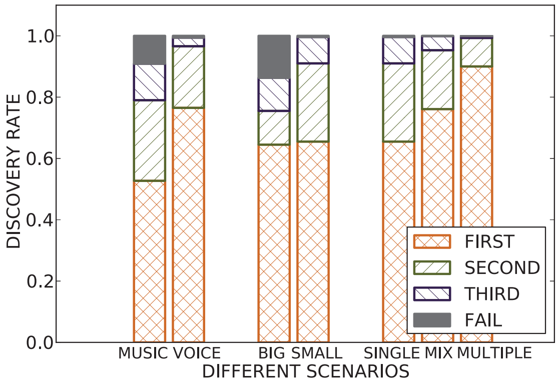


Fig. 7. Discovery Ratio in Different Period

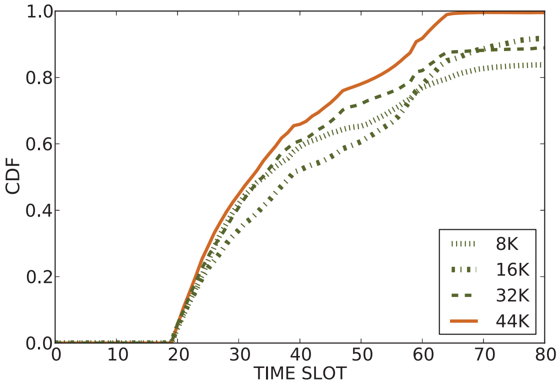


Fig. 8. Discovery Ratio under Different Sample Rate

C. Different Sample Rate

To further shrink energy consumption, we consider lowering the microphone's sampling rate. Thus, we compare the performance of AIR under different sampling rates, that is, 8K, 16K, 32K, and the normal 44.1K. In order to keep AIR recording the same length of ambient acoustic information, we decrease the number of sample points individually for fairness of comparison.

Certainly, a low sample rate loses some discovery accuracy. Surprisingly, we find that the reduction is not severe for AIR as the success rates of different sampling rates are all basically over 85%. The robustness of AIR under different sampling rates is owing to the well-designed audio event detection. Referring to Fig. 8, the sampling rate of 8K quickly discovers some neighbors in the first period of scheduling.

It performs especially well in the background of a human voice. We consider that under the sampling rate of 8K, the major frequency spectrum of human voice remains and its limited bandwidth enhances its sensitivity to voice events. Interestingly, we observe that the curve of the sample rate of 32K has a cross with that of 16K, and is surpassed in terms of the total success rate. We think that it is due to the effect of the filter where only the low frequency part is filtered. As a result, sampling rate of 32K retains some high frequency part from 11K to 16K, which causes its false detection somehow. From another point of view, it demonstrates the necessity of a bandpass filter and verifies that we set a appropriate bandpass range for the filter.

VI. Conclusion and Future Work

In this study, we explore ambient acoustic information to decide the radio-on schedule for neighbor discovery on mobile devices. AIR leverages an ubiquitous microphone to detect audio events occurring nearby. Through extensive evaluation in different scenarios, the effectivity and feasibility of AIR are verified. On average, AIR significantly reduces the discovery latency by nearly 70% compared with the state-of-art neighbor discovery protocol, which basically satisfies users' demand in practical usage.

In the future, we plan to conduct additional experiments in more complex scenarios, further enhance the robustness of AIR, and integrate AIR into real applications. In addition, we are planning to utilize more professional energy consumption tools to measure and compare the energy savings of AIR at a finer granularity.

Acknowledgment

This research was supported in part by Hong Kong RGC Grant HKUST16207714, National Natural Science Foundation of China Grant No. 61173156, and Nansha S&T Project Grant No. 2013G004.

References

[1] "Nintendo 3ds - streetpass," http://www.nintendo.com/3ds/features.

[2] "Sony ps vita - near," http://us.playstation.com/psvita/apps/psvita-app-near.html.

[3] "Nextdoor," https://nextdoor.com/.

[4] "Global games investment review," http://www.digi-capital.com/.

[5] "Mobile social networking," http://www.abiresearch.com/press/over-17-billion-mobile-social-networking-users-in-.

[6] L. Zhang, X. Ding, Z. Wan, M. Gu, and X.-Y. Li, "Wiface: a secure geosocial networking system using wifi-based multi-hop manet," in *Proc. of ACM Workshop on MCC*, 2010.

[7] T. Liu, C. M. Sadler, P. Zhang, and M. Martonosi, "Implementing software on resource-constrained mobile sensors: Experiences with impala and zebranet," in *Proc. of ACM MobiSys*, 2004.

[8] D. L. Mills, "Improved algorithms for synchronizing computer network clocks," *IEEE Transactions on Networking*, vol. 3, no. 3, pp. 245–254, 1995.

[9] M. McGlynn and S. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. of ACM MobiHoc*, 2001.

[10] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. of ACM SenSys*, 2008.

[11] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proc. of ACM IPSN*, 2010.

[12] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?" in *Proc. of ACM MobiCom*, 2012.

[13] M. Uddin and T. Nadeem, "A2PSM: audio assisted wi-fi power saving mechanism for smart devices," in *Proc. of ACM HotMobile*, 2013.

[14] Z. Sun, A. Purohit, R. Bose, and P. Zhang, "Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing," in *Proc. of ACM MobiSys*, 2013.

[15] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: mobile phone localization via ambience fingerprinting," in *Proc. of ACM MobiCom*, 2009.

[16] P. Lazik and A. Rowe, "Indoor pseudo-ranging of mobile devices using ultrasonic chirps," in *Proc. of ACM SenSys*, 2012.

[17] C. Peng, G. Shen, Y. Zhang, and S. Lu, "Point&connect: intention-based device pairing for mobile phone users," in *Proc. of ACM MobiSys*, 2009.

[18] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda, "Swordfight: enabling a new class of phone-to-phone action games on commodity phones," in *Proc. of ACM MobiSys*, 2012.

[19] P. G. Kannan, S. P. Venkatagiri, M. C. Chan, A. L. Ananda, and L.-S. Peh, "Low cost crowd counting using audio tones," in *Proc. of ACM SenSys*, 2012.

[20] Y. Tseng, C. Hsu, and T. Hsieh, "Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks," *Elsevier Computer Networks*, vol. 43, no. 3, pp. 317–337, 2003.

[21] S. Lai, "Heterogenous quorum-based wakeup scheduling for duty-cycled wireless sensor networks," Ph.D. dissertation, Virginia Polytechnic Institute and State University, 2009.

[22] D. Zhang, T. He, Y. Liu, Y. Gu, F. Ye, R. K. Ganti, and H. Lei, "Acc: generic on-demand accelerations for neighbor discovery in mobile applications," in *Proc. of ACM SenSys*, 2012.

[23] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proc. of ACM SenSys*, 2008.

[24] T. P. Gill, *The Doppler Effect*. New York: Academic, 1965.

[25] T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for ofdm," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

[26] S. Mason, C. Berger, S. Zhou, and P. Willett, "Detection, synchronization, and doppler scale estimation with multicarrier waveforms in underwater acoustic communication," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 9, pp. 1638–1649, 2008.

[27] H. Lu, A. B. Brush, B. Priyantha, A. K. Karlson, and J. Liu, "Speakersense: energy efficient unobtrusive speaker identification on mobile phones," in *Proc. of Pervasive*, 2011.

[28] "Zigbee alliance," http://www.zigbee.org/Home.aspx.

[29] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *Proc. of ACM MobiSys*, 2009.

[30] M. F. McKinney and J. Breebaart, "Features for audio and music classification," in *Proc. of ISMIR*, 2003.

[31] E. SCHEIRER and M. SLANEY, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proc. of IEEE ICASSP*, 1997.

[32] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee, "Classification of general audio data for content-based retrieval," *ElsevierPattern recognition letters*, vol. 22, no. 5, pp. 533–544, 2001.

[33] S. Dixon, "Onset detection revisited," in *Proc. of DAFx*, 2006.