This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Have You Recorded My Voice: Toward Robust Neighbor Discovery in Mobile Wireless Networks

Fan Wu, *Member, IEEE*, Tong Meng, *Student Member, ACM*, Aijing Li, *Member, IEEE*, Guihai Chen, *Senior Member, IEEE*, and Nitin H. Vaidya, *Fellow, IEEE*

***Abstract*— The surge of proximity-based applications on mobile devices has promoted the need for effective neighbor discovery protocols in mobile wireless networks. In contrast to existing works, which can achieve energy efficient neighbor discovery with bounded latency only in the scenario without strong interference, we aim at designing techniques for practical and robust neighbor discovery. We propose ReCorder to achieve robust neighbor discovery in mobile wireless networks despite the "noisy" communication media. Specifically, we exploit the cross-correlation property of pseudo-random sequences to eliminate the necessity of beacon decoding in existing neighbor discovery protocols. In ReCorder, a neighbor discovery message can be detected through cross-correlation on an RCover preamble, and contains a ReCord identity signature, which is unique for each of the nodes. We also design algorithms for RCover detection and ReCord recognization. The performance of the ReCorder has been evaluated using the USRP-N210 testbed. Our evaluation results show that the ReCorder can achieve robust neighbor discovery at an SINR lower than the existing beaconing and decoding-based neighbor discovery protocols by almost 10 dB. Furthermore, the ReCorder can avoid degrading the decoding of background IEEE 802.11 a/g transmissions with BPSK modulation, which is important for its co-existence with concurrent wireless streams, and it only induces limited throughput degradation to background data flows.**

***Index Terms*— Neighbor discovery, wireless networks, experimentation.**

I. Introduction

NOWADAYS, thanks to the increasing communication and computation capabilities of mobile wireless devices (*e.g.*, smartphones and tablets), users can enjoy the convenience of diverse proximity-based applications. For instance, on the trip to a French travel resort, one can have a rest at a street coffee house by playing video games using her Sony's Vita [1] with nearby users. It has been demonstrated that the ability of discovering neighbors within a mobile device's wireless communication range can exert the full potential of such proximity-based applications. That has motivated works on neighbor discovery in mobile wireless networks (*e.g.*, [4], [7], [34], [44]). Considering the limited energy budgets on mobile devices and the unpredictable mobility of device users, most of existing works focus on designing energy and time efficient neighbor discovery protocols. Neighbor discovery protocols need not only to avoid the energy bottleneck, but also to capture the short contact periods between neighboring nodes. Thus, during the process of neighbor discovery, each mobile device has to conform to a relatively low duty cycle owing to limited battery power. In the meanwhile, the device transforms its state between active and power-saving according to a deterministic schedule subject to the duty cycle, which guarantees the worst-case bound of discovery latency.

Even though a number of neighbor discovery protocols have been proposed and proven to achieve good performance theoretically, most of them ignore an important characteristic of mobile wireless communication environment, *i.e.*, the busy communication media. Specifically, existing protocols simply use beacons as the messages for neighbor discovery, *i.e.*, each node sends beacons when it is active, and decodes the received beacons to obtain the identity of its neighbors. However, in mobile wireless networks, the existence of many interfering wireless signals, such as file transfer from a laptop to a smartphone and delivery of a webpage to a tablet, can easily impair the possibility of beacon decoding. Even with carrier sensing, neighbor discovery beacons may still collide with other signals due to various reasons, *e.g.*, hidden terminal. Moreover, the beacons have a much smaller size (around 30 bytes) than regular data frames (up to 4095 bytes in IEEE 802.11 OFDM [3]). They are likely to be hidden in the shadow of other packets once there are collisions. That means existing beaconing and decoding based neighbor discovery protocols tend to fail unless nodes can receive the beacons without strong interference. Such shortage restricts their robustness in the existence of interfering signals, and thus, undermines their performance when applied in practical mobile wireless networks. Consequently, it is vital to design techniques to improve the robustness of neighbor discovery protocols.

Unfortunately, simply adding reliability to the decoding of beacons cannot satisfy the requirements of neighbor discovery. On one hand, because each node turns active and sends beacons according to a deterministic schedule, a nodes cannot distinguish between the scenario with no active neighbors and beacon lost. Moreover, considering the low duty cycle, the acknowledgement/retransmission schemes, such as

Manuscript received January 6, 2016; revised January 25, 2017, August 1, 2017, and March 21, 2018; accepted April 4, 2018; approved by IEEE/ACM Transactions on Networking Editor D. Goeckel. This work was supported in part by the State Key Development Program for Basic Research of China (973) under Project 2014CB340303, in part by China NSF under Grant 61672348, Grant 61672353, and Grant 61472252, and in part by the Shanghai Science and Technology Fund under Grant 15220721300. *(Corresponding author: Fan Wu.)*

F. Wu, T. Meng, and G. Chen are with the Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: fwu@cs.sjtu.edu.cn; mengtong@sjtu.edu.cn; gchen@cs.sjtu.edu.cn).

A. Li is with the PLA University of Science and Technology, Nanjing 210007, China (e-mail: lishan.wh@gmail.com).

N. H. Vaidya is with the University of Illinois at Urbana–Champaign, Champaign, IL 61820 USA (e-mail: nhv@illinois.edu).

Digital Object Identifier 10.1109/TNET.2018.2824848

H-ARQ [30], will induce too many unnecessary transmissions. On the other hand, it will increase the energy burden if some difficult coding schemes are adopted. Besides, to achieve robust neighbor discovery in practice, we need to cope with two additional major challenges.

- As the first step in neighbor discovery, we need a way to detect neighbor discovery messages among the other concurrent transmissions, considering the complicated wireless communication environments. Similarly, a mobile device should be able to recognize the identities of different neighbors. Thus, instead of using beacons, the messages should have a well-designed structure that is specific to neighbor discovery.
- Moreover, neighbor discovery should be able to co-exist with the decoding of other packets. On one hand, the robustness of neighbor discovery requires that it should not be impaired by interfering data transmissions. On the other hand, the decoding of other packets should also not be impeded by the neighbor discovery messages.

To tackle the above challenges, we utilize the correlation property of pseudo-random sequences, and propose a novel and robust neighbor discovery scheme named ReCorder. In ReCorder, we use a pseudo-random preamble to distinguish neighbor discovery messages, which is named as RCover. Moreover, just like people can recognize each other through tunes of voices, ReCorder uses well-defined signatures called ReCord to distinguish different neighboring nodes.[1] Both the detection of the preamble and the recognition of identity signatures exploit cross-correlation. Therefore, decoding is not needed in the process of neighbor discovery.

The detailed contributions are listed in the following.

- To the best of our knowledge, ReCorder is the first to enable effective neighbor discovery despite interference in the communication media. We propose algorithms for RCover detection and ReCord recognition by exploiting the correlation property of pseudo-random sequences, which contain a practical estimation of the SINR level, as well.
- We prototype ReCorder on a USRP-N210 testbed. The evaluation results show that ReCorder can improve the robustness of neighbor discovery protocols significantly, *i.e.*, it can successfully detect the RCover, and recognize the ReCord in the neighbor discovery message in more than 90% of times at an SINR of -6dB, which is about 10dB lower than the existing beaconing and decoding based neighbor discovery protocols. Furthermore, ReCorder enables shorter neighbor discovery messages, which is more energy-efficient with less transmission power and no decoding overheads.
- We analyze and evaluate the influence of ReCorder on background OFDM transmissions using the IEEE 802.11a/g protocol. We conclude that ReCorder can bring no degradation to IEEE 802.11a/g protocol with BPSK modulation, and minimize its impact on the decoding of OFDM packets by occupying at least the same bandwidth as OFDM. Additionally, the simulation results show that ReCorder does not obviously reduce the throughput of the background flow.

[1]The proposed neighbor discovery scheme is named as ReCorder, inspired from the fact that we use a "*recorder*" to record different voice. Similarly, the names RCover and ReCord are used out of the consideration that RCover is like the front cover of some music record, and ReCord is like the actual voice information.

The rest of the paper is organized as follows. In Section II, we discuss the related works. In Section III, we introduce our motivation as well as the preliminary knowledge on wireless communication. The overview and design details of ReCorder are presented in Section IV, which is followed by the evaluation results in Section V. Then, we discuss several practical issues in Section VI, and conclude the paper in Section VII.

II. Related Works

In this section, we briefly introduce existing works on neighbor discovery, and discuss related works that implement cross-correlation.

A. Neighbor Discovery Protocols

The problem of neighbor discovery has been extensively investigated in both sensor networks [18], [41] and mobile wireless networks [45]. Most of the existing works focus on designing efficient neighbor discovery protocols. They divide time into slots, and restrict each node by some duty cycle. Generally, existing neighbor discovery protocols fall into two categories: probabilistic protocols and deterministic protocols. In addition, if all the nodes have the same duty cycle, it is called symmetric neighbor discovery. Otherwise, it is called asymmetric neighbor discovery.

In probabilistic protocols, each node probabilistically determines to transmit, receive, or sleep in each slot. Birthday protocol proposed by McGlynn and Borbash [33] is the foundation of most of the following probabilistic neighbor discovery protocols. Based on [28] and [33] deals with neighbor discovery under the reception feedback mechanism. References [26], [48], and [54] further extend the neighbor discovery to multi-hop, multi-channel, and multi-packet reception networks, respecticely. References [8] and [49] propose neighbor discovery algorithms using directional antennas. Vasudevan *et al.* [50] reduce the probabilistic neighbor discovery algorithm to the Coupon Collector's Problem. In addition, [32] designs a neighbor discovery protocol for power harvesting transceivers. Those probabilistic protocols can support both symmetric and asymmetric cases, but cannot guarantee the bound on discovery latency in the worst case.

In deterministic protocols, there is a fixed active-sleep pattern scheduling nodes' periodic state transformation. In [47] and [24], each cycle of a node is regarded as a quorum. By exploiting the randomized combinatorial characteristics of the quorum schedule, [6] managed to improve the quorum construction for neighbor discovery. Zheng *et al.* [56] applied optimal block design. However, these works are mainly restricted to symmetric duty cycle. Although [29] is then designed to support asymmetric cases, it is still restricted to only two different duty cycles. Galluzzi *et al.* proposed a straightforward pattern with at least half of the slots in each period being active, but such an active-sleep pattern leads to excessive energy consumption. To overcome such a limitation, many protocols are designed to spread the active slots to multiple consecutive cycles. For instance, primed-based protocols, such as Disco [12] and U-Connect [25], implement primes to generate active-sleep pattern. Besides, Searchlight [4] and Hello [44] leverage the regular relation between the probing schedules of different nodes. Furthermore, by exploiting asynchronization, Meng *et al.* [34], [35] derived the lower bound of discovery latency, and designed

(A)Diff-Codes. Chen *et al.* [9] improved the design of active-sleep schedule by a non-integer, continuous-time model.

Most of previous neighbor discovery protocols make efforts to realize different trade-off relations between high energy and time efficiency [17], by improving the active-sleep pattern. They all rely on the beaconing mechanism. The decoding of beacons is necessary in discovering neighbors. Nevertheless, in this work, we argue that such beaconing and decoding based protocols lack robustness, and do not work well in practice owing to wireless noise and possible interfering signals in mobile wireless networks.

B. Related Works on Cross-Correlation

Cross-correlation is usually implemented to recognize some known pseudo-random sequences. For example, Sen *et al.* [42] proposed CSMA/CN that utilizes the correlation property of a pseudo-random signature to notify the detection of collision. Wu *et al.* [52] built a Side Channel for efficient medium access by the correlation of some intended patterns.

Furthermore, Zhang and Shin [55] designed E-MiLi that enables downclocked radios through the correlation of M-preambles. Magistretti *et al.* [31] designed 802.11ec, and replaced the control messages in IEEE 802.11 with correlatable symbol sequences. Both [31] and [55] include addressing information in pseudo-random sequences. Specifically, E-MiLi uses different sequence lengths to convey addresses implicitly, while 802.11ec allocates multiple correlatable symbol sequences to each node for selection. However, E-MiLi assumes limited size of the networks, and 802.11ec requires each node to have the knowledge of its neighboring nodes. Thus, they cannot be applied in neighbor discovery, where each node in the networks needs a unique identity signature.

In addition, there are works on message detection in different scenarios such as [15] and [23]. They utilize the direct sequence spread spectrum (DSSS) technique [10], [38], [43], spreading the data bits using pseudo-random codes, which are then correlated at the receiver to detect the appearance of specific data bits. Similar to the pseudo-random sequence correlation in ReCorder, the correlation of pseudo-random codes in DSSS improves the interference tolerance, as well. Nevertheless, as the node identity in ReCorder, both [15] and [23] use spread-spectrum to convey only limited information, *i.e.*, wireless control messages and collision detection, respectively. Therefore, unlike the decoding in DSSS-based mechanisms (*e.g.*, IEEE 802.11b), ReCorder, as well as such related works, can still achieve high data efficiency despite the spreading.

There are also works on cooperative packet recovery (*e.g.*, [5], [19]) utilizing correlation together with interference cancellation. Unfortunately, they fail to rescue the existing neighbor discovery protocols from insufficient robustness. On one hand, with cooperative packet recovery, the decoding of a collided beacon needs multiple receptions from the same neighbor, leading to longer discovery latency. On the other hand, [5] and [19] require controllable collisions, which does not fit the unpredictable interference in mobile wireless networks.

III. Motivation and Preliminaries

In this section, we verify the necessity of designing robust neighbor discovery technique. Then, we briefly introduce the preliminaries on wireless communication.

A. Motivation for Robustness in Neighbor Discovery

Most existing neighbor discovery protocols adopt beacons as neighbor discovery messages. In IEEE 802.11 OFDM [36], a beacon starts with specially designed training preambles, followed by its packet header and data payload. To accomplish the discovery process, a node first uses autocorrelation on the training preambles to discover the beacon packet, and then decodes the payload to extract the MAC address of a neighboring node. Moreover, before any decoding, the node needs to rely on the training preambles for frequency and symbol-level synchronization.

However, the existence of interfering signals will severely impact the accuracy of the above synchronization, and pollute the received beacon symbols, as well. Thus, the decoding of MAC address in the beacon can be easily impeded due to interfering transmissions, which cannot be escaped in practice even by carrier sensing mechanism [31], [42], [52].

Besides, considering the low duty cycle of neighbor discovery, failing to decode even a single received beacon can result in much longer discovery latency. In the worst case, such unexpected delay may miss the short contact opportunity between two mobile nodes.

Therefore, the lack of robustness of existing beaconing and decoding mechanism for neighbor discovery may significantly restrict its practical implementation. A robust technique for neighbor discovery in practical mobile wireless networks is highly needed.

B. Preliminaries on Wireless Communication

Wireless signals are typically streams of discrete complex symbols. Specifically, a wireless transmitter modulates the binary bits of a packet into complex constellation points before sending the packet on a wireless channel. According to the implemented digital modulation scheme, every fixed number of binary bits are transformed into a single complex symbol. For example, in BPSK modulation, bit 0 is mapped to $e^{j\pi} = -1$, and bit 1 to $e^{j0} = 1$.[2]

In particular, after a packet $\mathbf{x}$ is transmitted, the i-th received complex symbol $\mathbf{y}_i$, which corresponds to the i-th transmitted complex symbol $\mathbf{x}_i$, can be represented as,

$$\mathbf{y}_i = \mathbf{h}_i \mathbf{x}_i + \mathbf{n}_i, \tag{1}$$

where $\mathbf{n}_i$ includes the random noise, as well as the other possible interfering signals, and $\mathbf{h}_i$ is the channel coefficient between the transmitter and the receiver. The magnitude and angle of $\mathbf{h}_i$ capture the channel attenuation and the phase shift of the i-th symbol, respectively.

In wireless communications, a node can detect a known pseudo-random pattern $\mathbf{s}$ composed of L complex symbols by performing cross-correlation [27] between the received signal and the known pattern. For example, the DSSS technique adopts cross-correlation on pre-defined pseudo-random codes for data bits decoding.

Specifically, given the received signal $\mathbf{y}$, its cross-correlation with the pattern $\mathbf{s}$ at position Δ is computed as,

$$C(\mathbf{s}, \mathbf{y}, \Delta) = \sum_{i=1}^{L} \left(\mathbf{s}_i^{*} \cdot \mathbf{y}_{i+\Delta} \right), \tag{2}$$

[2] The actual transmitted complex symbols should be normalized according to the transmission power.

where $\mathbf{s}_i^*$ is the complex conjugate of the i-th symbol in $\mathbf{s}$. Considering that the pattern is pseudo-random, it is independent of the noise and possibly the interfering signals. Hence, the magnitude of $C(\mathbf{s}, \mathbf{y}, \Delta)$ is quite small, except when the received signal $\mathbf{y}$ contains an aligned copy of $\mathbf{s}$, *i.e.*, the copy of the pattern starts at position Δ. In that case, we have,

$$\begin{aligned} C(\mathbf{s}, \mathbf{y}, \Delta) &= \sum_{i=1}^{L} (\mathbf{s}_i^* \cdot \mathbf{y}_{i+\Delta}) \\ &= \sum_{i=1}^{L} [\mathbf{s}_i^* \cdot (\mathbf{h}_{i+\Delta}\mathbf{s}_i + \mathbf{n}_{i+\Delta})] \\ &\approx \sum_{i=1}^{L} \left(\mathbf{h}_{i+\Delta} \cdot |\mathbf{s}_i|^2\right). \end{aligned} \tag{3}$$

The above result approximately reflects the total energy level in the received pattern, and is extraordinarily large. Therefore, in practice, a wireless receiver continuously computes the cross-correlation between the known pattern and the most recent L received complex symbols, until a peak magnitude is observed. The peak in the correlation result indicates the appearance of a pattern $\mathbf{s}$.

Provided the correlation property of pseudo-random symbol pattern, a detection threshold is necessary in determining whether a pattern appears in the received signal. If the magnitude of cross-correlation $C(\mathbf{s}, \mathbf{y}, \Delta)$ exceeds the threshold, it implies that a pattern starts at position Δ in $\mathbf{y}$. A larger detection threshold yields a higher false negative probability, while a smaller threshold may induce false positives. By the correlation theory based on Gaussian noise [27], the optimal threshold is given by,

$$\mathit{Threshold} = Q^{-1}(Pr_{FP}) \cdot \sqrt{\frac{L \cdot P(\mathbf{s}) \cdot P(\mathbf{n})}{2}}, \tag{4}$$

where Pr_{FP} is the target false positive probability, Q is the tail probability of the standard normal function, and $P(\mathbf{s})$ ($P(\mathbf{n})$) is the power of the received pattern (random noise).

IV. Design of ReCorder

In this section, we present the design of ReCorder in detail. We first give a brief overview of how ReCorder works, and then explain the components of ReCorder in correspondence to the challenges underlying robust neighbor discovery.

A. Overview

ReCorder provides a novel technique for robust neighbor discovery, which is highly needed in mobile wireless networks. It replaces the decoding of beacons in previous works with cross-correlation, and thus can be applied to almost all the existing neighbor discovery protocols (*e.g.*, [4], [12], [25], [33], [34]) to enhance robustness.

Specifically, the message for neighbor discovery is designed to be a pseudo-random sequence, which is composed of an RCover preamble and a ReCord signature. Correspondingly, based on correlation as shown in Equation (2), two algorithms called RMix and RMix-2 are proposed for RCover and ReCord recognition, respectively. First, by exploiting the thresholding RMix algorithm, the message is filtered out by correlating the received complex symbols to a fixed pseudo-random preamble, called RCover. The RCover preamble is known to all the nodes, and can be determined beforehand. Second, a message for neighbor discovery contains the identity signature of its sender, which is named as ReCord in this work. Each ReCord is unique and provides the information of a 2-level identity. Upon detecting an instance of RCover, a wireless node should feed the following complex symbols to the RMix-2 algorithm. The RMix-2 algorithm correlates those symbols to the stored ReCord signatures, and uses threshold-based method to determine whether that is a new ReCord from a new neighboring node or not. Specifically, Fig. 1 shows the corresponding format of a message for neighbor discovery.

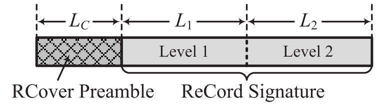


Fig. 1. Structure of messages for neighbor discovery in ReCorder.

B. RCover: Distinguishing Packets for Neighbor Discovery

In mobile wireless networks, neighbor discovery is unlikely to be the only source of wireless signals within the transmission proximity of a wireless radio on user's mobile device. At the same time with neighbor discovery, a wireless node may (over)hear other packet transmissions, *e.g.*, WiFi downloading streams, file transmissions through WiFi Direct [2], *etc*. In such situation, the decoding of incoming beacons for neighbor discovery will be easy to be corrupted owing to the low SINR. Thus, to realize robust neighbor discovery, the first challenge is to enable wireless nodes to distinguish messages for neighbor discovery efficiently without decoding.

To this end, we prepend an RCover preamble to each message for neighbor discovery (as shown in Fig. 1). The RCover preamble is a pseudo-random sequence with L_C complex symbols. It is known to all the nodes, and can be detected by cross-correlation with the received symbols. The RCover preamble should have good correlation property, *i.e.*, the magnitude of correlation spikes only when it is correlated with exactly itself. We select the Gold code [14] as the choice of RCover preamble, and apply BPSK to modulate the binary Gold code into complex symbols for transmission. Then, during the process of neighbor discovery, when the wireless interface of the node is turned on, it tries to detect neighbor discovery messages by continuously correlating the recently received L_C complex symbols with the local copy of RCover sequence $\mathbf{s_C}$. A ReCord signature starts at the position where the correlation result spikes. There are three practical issues (carrier frequency offset, phase offset, and detection threshold setting) existing in implementing this idea.

1) Frequency Offset: Two different wireless radios may have an offset in their center frequency (denoted by δf). This frequency offset leads to a phase rotation in the received symbols between a pair of wireless transmitter and receiver,

$$\mathbf{y}_i = \mathbf{h}_i \mathbf{x}_i e^{j2\pi T i \delta f} + \mathbf{n}_i, \tag{5}$$

where T is the sampling period at the receiver. The equation shows that the induced phase rotation accumulates over time. In a packet decoding system, such accumulated rotation may lead to decoding errors if not compensated. However, the frequency offset is generally small enough (e.g., less than 4 KHz [42]), so that it does not obviously influence the RCover detection, provided that we keep the length of the preamble small [31].

2) Phase Offset: Due to lack of perfectly aligned close phases, two wireless radios can have a phase offset, generating a fixed rotation of the received symbols. Such a carrier phase offset may also influence the received symbols, but it can be easily avoided by calculating correlation *magnitude* as in [31], which according to [31] only induces limited penalty (*e.g.*, no greater than 0.5dB in theory [40]).

3) Detection Threshold: As explained in Section III, a threshold is necessary for RCover detection judgement. However, the threshold in Equation (4) requires a non-trivial estimation of the SINR value. From the perspective of energy efficiency, it is impractical to measure the actual signal power to calculate the SINR. However, SINR estimation in existing works either relies on previous decoding results [19], or assumes a large enough SINR [55]. By contrast, the detection of RCover uses cross-correlation without any decoding, and is designed to adapt to interfering signals. Thus, previous methods are inapplicable for RCover detection.

Nevertheless, we notice that when the received symbols contain RCover, the magnitude of their correlation with the copy of RCover sequence approximates to the received power of RCover. In the meanwhile, the self-correlation of the received L_C symbols is a coarse approximation of their energy level. Inspired by that, we can estimate the SINR regarding the received RCover by,

$$\mathit{SINR}_c = \frac{|C(\mathbf{s}_C, \mathbf{y}, 0)| - \mathcal{C}(k-1)}{C(\mathbf{y}, \mathbf{y}, 0) - |C(\mathbf{s}_C, \mathbf{y}, 0)| + \mathcal{C}(k-1)}, \tag{6}$$

where $\mathbf{y}$ stores the most recently received L_C complex symbols[3] at sampling point k, and $\mathcal{C}(k-1)$ is the moving average of cross-correlation magnitude at previous sampling point $(k-1)$. We calculate $\mathcal{C}(k)$ as,

$$\mathcal{C}(k) = (1-\eta_s)\cdot \mathcal{C}(k-1) + \eta_s \cdot |C(\mathbf{s}_C, \mathbf{y}, 0)|, \tag{7}$$

where η_s is the learning rate. In this work, we take the value of η_s to be around $(L_C)^{-1}$.

Alternatively, considering the short period of the RCover preamble, the energy level of the received symbols preceding an instance of RCover tends to reflect the corresponding noise and possible interference. So we can also calculate the SINR at sampling point k as,

$$\mathit{SINR}_e = \frac{C(\mathbf{y}, \mathbf{y}, 0) - E_s(k-L_C)}{E_s(k-L_C)}. \tag{8}$$

Similar to Equation (7), we maintain a weighted average of the received energy level,

$$E_s(k) = (1-\eta_s)\cdot E_s(k-1) + \eta_s \cdot C(\mathbf{y}, \mathbf{y}, 0). \tag{9}$$

In this work, we use SINR_c to determine the threshold for RCover detection. Specifically, referring to Equation (4), the detection threshold for RCover detection is set to be,

$$T_C = \beta_1 \cdot \sqrt{L_C \cdot \frac{(|C(\mathbf{s}_C, \mathbf{y}, 0)| - \mathcal{C}(k-1))^2}{\mathit{SINR}_c}} + \beta_2 \cdot \mathcal{C}(k-1), \tag{10}$$

where β_1 and β_2 are both constants balancing false positives and false negatives. Besides, we add a second term $(\beta_2 \cdot \mathcal{C}(k-1))$ on above to avoid false positives.

[3]In our experiments, we normalize the symbols in $\mathbf{y}$ before correlating it with $\mathbf{s}_C$.

Algorithm 1 RMix Algorithm for RCover Detection

Input: The received L_C symbols $\mathbf{y}$ at sampling point k, a copy of the RCover sequence $\mathbf{s}_C$.

Output: A flag indicating whether an RCover is detected at sampling point k.

1 $E_1 \leftarrow C(\mathbf{y}, \mathbf{y}, 0)$; $C_1 \leftarrow |C(\mathbf{s}_C, \mathbf{y}, 0)|$;
2 $E_s(k) \leftarrow (1-\eta_s)\cdot E_s(k-1) + \eta_s \cdot E_1$;
3 $E_f(k) \leftarrow (1-\eta_f)\cdot E_f(k-1) + \eta_f \cdot E_1$;
4 $\mathcal{C}(k) \leftarrow (1-\eta_s)\cdot \mathcal{C}(k-1) + \eta_s \cdot C_1$;
5 $\mathit{SINR}_c \leftarrow [C_1 - \mathcal{C}(k-1)] \,/\, [E_1 - C_1 + \mathcal{C}(k-1)]$;
6 $\mathit{SINR}_e \leftarrow [E_1 - E_s(k-L_C)] \,/ E_s(k-L_C)$;
7 **if** $\mathit{SINR}_c > 0$ **and** $\mathit{SINR}_e > 0$ **and** $10\lg \mathrm{SINR}_e > H_L$ **and** $r < E_f(k)/E_1 < r^{-1}$ **then**
8 $\quad T_C \leftarrow \beta_1 \cdot \sqrt{L_C(C_1 - \mathcal{C}(k-1))^2/\mathit{SINR}_c} + \beta_2 \cdot \mathcal{C}(k-1)$;
9 $\quad$ **if** $C_1 > T_C$ **then return True** ;
10 $\quad$;
11 **end**
12 **return False** ;

However, despite the effectiveness of Equation (10) in the existence of RCover, we may also falsely detect some non-existing RCover sequences. For instance, the threshold can be quite low during the idle period of the channel, in which the received energy level is close to zero. To avoid false positives like that, we set a lower bound on the received signal strength that we aim to support with SINR_e. To be specific, the thresholding examination using T_C is only triggered when $10\lg \mathit{SINR}_e$ is above H_L (set to -10dB in this work). In addition, we only calculate the threshold for detection when the average energy level is close to the energy level of received symbols. Hence, we maintain an average energy level with a faster learning rate (*i.e.*, $\eta_f > 2\eta_s$) than $E_s(k)$,

$$E_f(k) = (1-\eta_f)\cdot E_f(k-1) + \eta_f \cdot C(\mathbf{y}, \mathbf{y}, 0). \tag{11}$$

Then we have the following judgement before turning to the threshold T_C,

$$r < \frac{E_f(k)}{C(\mathbf{y}, \mathbf{y}, 0)} < r^{-1}, \tag{12}$$

where r is a constant approximate to 1, and is empirically set to 0.8 in our evaluation. With Equation (12), we can filter out those short jitters of energy level due to the changing wireless channel.

On the above basis, we propose the RMix algorithm for RCover detection. The pseudo-code of RMix is shown in Algorithm 1. In the beginning of the algorithm, it calculates the energy level of the newly received L_C complex symbols, as well as their correlation with $\mathbf{s_C}$. Then the moving averages, as well as the SINR estimations, are updated. Thereafter, RMix determines whether the thresholding judgement should be triggered according to the given rules (line 7), and calculates the detection threshold if necessary (line 8). Because at each sampling point, the RMix algorithm only involves several single step computations such as updating the moving averages and calculating the threshold if necessary, it is of linear complexity with respect to the length of RCover (L_C). That can be easily satisfied by the processor in practice.

In fact, we find that when the SINR is high enough, the multiple of the average correlation magnitude can simply replace the threshold T_C computed by Equation (10). In this work, we compare the correlation magnitude directly with $\beta_3 \cdot \mathcal{C}(k-1)$ (*e.g.*, $\beta_3 = 5$ gives satisfying performance in our evaluation) to determine the appearance of an RCover preamble, as long as $10\lg SINR_e$ is above 0dB. That can further reduce the complexity of RCover detection.

It is important to note that, the selected Gold sequence with length $L = 2^l - 1$ guarantees a self-correlation magnitude that is at least $2^{\frac{l-1}{2}}$ times higher than any secondary correlation peak when it correlates with any shifted version of itself [14]. Thus, the RMix algorithm will not be impeded by *aperiodic auto-correlation* with partial, unaligned RCover sub-sequence in the received symbols.

C. ReCord: Identity Signature of Neighboring Nodes

The detection of RCover sequence is the first step of robust neighbor discovery in mobile wireless networks. Another challenge is to recognize different neighboring nodes. Most of the established neighbor discovery protocols employ the MAC address in the beacons to convey identity information. Whereas due to the same reasons as specified in Section IV-B, the decoding of beacons is sometimes infeasible for practical applications in mobile wireless networks. Thus, we prefer correlation rather than decoding for neighbor recognization. However, neighbor recognization is more complicated than RCover detection in that, instead of detecting a known sequence, each node needs to distinguish various neighbors. Therefore, we design the unique ReCord identity signature for each node, and propose RMix-2 algorithm to distinguish different ReCord signatures. The details are as follows.

1) 2-Level Identity Information: MAC address is generally used as the identity of each node in existing neighbor discovery protocols. However, the 48-bit (12 hexadecimal digits) MAC address has poor correlation property compared with Gold code. For example, if the MAC addresses of two nodes differ by only one or two digits, their correlation magnitude will generate a peak, which means that the two nodes may be falsely regarded as the same one.

Hence, instead of using the MAC address, the ReCord signature is designed to be a pseudo-random sequence, as well. To be specific, there are two levels of identity information in a ReCord. We implement Gold code [14] again as the level-1 identity. All the nodes use the same Gold code of length L_1, but pick different cyclic shift offsets randomly to generate their own level-1 ReCord signatures. As for the second level, each node randomly generates a sequence of length L_2. A hash function can be applied to map the MAC address of a node to its level-2 identity, so that each ReCord is guaranteed to be unique on the second level of identity information. We note that the reasons of such 2-level design of ReCord signature are twofold.

- First, the level-1 identity cannot exclude duplications. Given a fixed length L_1, the number of available cyclic shift offset is also limited by L_1. Considering the huge amount of mobile devices, it is possible that two neighboring nodes select the same offset, in which case they cannot be distinguished only by the level-1 ReCords.
- Second, the correlation property of the level-2 identity is inferior to the level-1 identity. To be specific, on the first level, the self-correlation peak of Gold code with length $L_1 = 2^l - 1$ is at least $2^{\frac{l-1}{2}}$ times higher than the secondary peak [14]. By contrast, the randomly generated level-2 identity fails to guarantee a bounded secondary peak, when correlated with its shifted sequence. Thus, the second level in ReCord acts as a supplement to the first level, in case that two nodes have the identical level-1 ReCord sometimes.

2) Recognizing ReCord Signatures: In ReCorder, each node maintains a table of received ReCord signatures, each of which represents a neighboring node without duplication. During the process of neighbor discovery, each time when a node discovers a neighbor discovery message, it should compare the newly received ReCord sequence in the message with the stored ones by means of cross-correlation. After determining whether the new ReCord is from a new neighbor or not, the node updates its local ReCord table accordingly.

For the recognization of ReCord, the cross-correlation between different ReCord signatures is not bothered by the frequency offset between nodes. To be specific, the frequency offset between two nodes is stable even over long periods of time [19]. Therefore, as long as a node receives two ReCords from the same neighboring node, these two ReCords will experience similar phase rotation, and their correlation will cancel out the effect of frequency offset. Actually, the frequency offset also contributes to the peak of the correlation magnitude. Mathematically, if we assume that a transmitter sends an L-symbol complex sequence $\mathbf{x}$ twice, and a receiver hears $\mathbf{y}$ and $\mathbf{y}'$ successively, then there is,

$$\begin{aligned} C(\mathbf{y}, \mathbf{y}', 0) &= \sum_{i=1}^{L} (\mathbf{y}_i^* \cdot \mathbf{y}'_i) \\ &= \sum_{i=1}^{L} \left(\mathbf{h}_i \mathbf{x}_i e^{j2\pi T i \delta f} + \mathbf{n}_i\right)^* \cdot \left(\mathbf{h}'_i \mathbf{x}_i e^{j2\pi T i \delta f} + \mathbf{n}'_i\right) \\ &\approx \sum_{i=1}^{L} \left(\mathbf{h}_i \mathbf{h}'_i \cdot \left|\mathbf{x}_i \, e^{j2\pi T i \delta f}\right|^2\right). \end{aligned} \tag{13}$$

The overall process of ReCord recognization is outlined as below. The node will first correlate the level-1 in the received ReCord with the Gold code for level-1 signature generation. That helps the node to determine the cyclic shift offset of the level-1 ReCord. After that, the node searches in the local table for ReCord signatures with the same cyclic shift offset on the first level. If such ReCords exist, it turns to the second level. Only if two ReCords match each other on both levels will the node conclude that they are from the same neighbors. Referring to equation (6), we have to know the average magnitude of cross-correlation. We estimate that approximately using the correlation results between the level-1 of the newly received ReCord and the known Gold code for level-1 signature generation. In practice, when a duplicated ReCord is received, the node should update the stored ReCord to be the one with higher SINR value. Otherwise, a new neighbor is discovered, and its ReCord is stored.

More details on the recognization of a newly received ReCord signature are summarized in the RMix-2 algorithm. The pseudo-code of RMix-2 is shown in Algorithm 2. In the beginning of Algorithm 2, it calculates the correlation between the received sequence $\mathbf{y}_1$ and the Gold code sequence $\mathbf{s}_L$ for level-1 signature generation under all the possible cyclic shift offsets (line 2-7). The algorithm takes the cyclic shift offset where the correlation magnitude is maximized to be the poten-

Algorithm 2 RMix-2 Algorithm for ReCord Recognization

Input: The complex symbols following a newly detected RCover, including $\mathbf{y}_1$ of length L_1 and $\mathbf{y}_2$ of length L_2, and the local copy $\mathbf{s}_L$ of Gold code for level-1 signature, and the stored ReCord table $\mathcal{T}$.

Output: The updated ReCord table.

1 $C_{avg}, C_{max} \leftarrow 0$; $U \leftarrow \phi$;
2 **for** *i from 0 to* $L_1 - 1$ **do**
3 $\quad Cr \leftarrow |C(\mathbf{s}_L, \mathbf{y}_1, i)|$; $C_{avg} \leftarrow C_{avg} + Cr$;
4 $\quad$ **if** $Cr > C_{\max}$ **then**
5 $\quad\quad C_{max} \leftarrow Cr$; $pos \leftarrow i$;
6 $\quad$ **end**
7 **end**
8 $C_{avg} \leftarrow (C_{avg} - C_{max})/(L_1 - 1)$;
9 $S_1 \leftarrow C_{max} - C_{avg}$; $I_1 \leftarrow C(\mathbf{y}_1, \mathbf{y}_1, 0) - S_1$;
10 **if** $C_{max} < \beta_1 \cdot \sqrt{L_1 \cdot S_1 \cdot I_1} + \beta_2 \cdot C_{avg}$ **then**
11 $\quad$ **return** $\mathcal{T}$;
12 **end**
13 **foreach** $< pos, \mathbf{s}_1, \mathbf{s}_2, sinr > \in \mathcal{T}$ **do**
14 $\quad C_2 \leftarrow |C(\mathbf{s}_2, \mathbf{y}_2, 0)|$;
15 $\quad C_{avg2} \leftarrow C_{avg} \cdot \sqrt{C(\mathbf{s}_2, \mathbf{s}_2, 0)/C(\mathbf{s}_L, \mathbf{s}_L, 0)}$;
16 $\quad S_2 \leftarrow C_2 - C_{avg2}$; $I_2 \leftarrow C(\mathbf{y}_2, \mathbf{y}_2, 0) - S_2$;
17 $\quad$ **if** $C_2 < C_{avg2}$ **or** $C_2 < \beta_1 \cdot \sqrt{L_2 \cdot S_2 \cdot I_2} + \beta_2 \cdot C_{avg2}$ **then**
18 $\quad\quad$ **continue** ;
19 $\quad$ **end**
20 $\quad C' \leftarrow |C(\mathbf{s}_1|\mathbf{s}_2, \mathbf{y}_1|\mathbf{y}_2, 0)|$;
21 $\quad C'_{avg} \leftarrow C_{avg} \cdot \sqrt{C(\mathbf{s}_1|\mathbf{s}_2, \mathbf{s}_1|\mathbf{s}_2, 0)/C(\mathbf{s}_L, \mathbf{s}_L, 0)}$;
22 $\quad S' \leftarrow C' - C'_{avg}$; $I' \leftarrow C(\mathbf{y}_1|\mathbf{y}_2, \mathbf{y}_1|\mathbf{y}_2, 0) - S'$;
23 $\quad$ **if** $C' > C'_{avg}$ **and** $C' > \beta_1 \cdot \sqrt{(L_1 + L_2) \cdot S' \cdot I'} + \beta_2 \cdot C'_{avg}$ **then**
24 $\quad\quad U \leftarrow U \cup \{< pos, \mathbf{s}_1, \mathbf{s}_2, sinr >\}$;
25 $\quad$ **end**
26 **end**
27 **if** $|U| = 0$ **then**
28 $\quad$ **return** $\mathcal{T} \cup \{< pos, \mathbf{y}_1, \mathbf{y}_2, S_1/I_1 >\}$;
29 **else if** $|U| = 1$ **and** $S_1/I_1 >$ *SINR(U)* **then**
30 $\quad$ **return** $\mathcal{T} \cup \{< pos, \mathbf{y}_1, \mathbf{y}_2, S_1/I_1 >\} \setminus U$;
31 **else**
32 $\quad$ **return** $\mathcal{T}$;
33 **end**

tial offset of $\mathbf{y}_1$. By the evaluation, that can effectively avoid false recognization. Then, RMix-2 examines the potential shift offset using the threshold computed by Equation (10) (line 10). This threshold check can also filter out false positives from the RCover detection by RMix algorithm in previous step. In the following, Algorithm 2 tries to match the received signature with the stored ReCords that have the same cyclic shift offset. Specifically, the algorithm examines the correlations on level-2 (line 14-19) and the whole signature (line 20-25), respectively, using the thresholding method. Finally, the local ReCord table is updated only if the matching results have no ambiguity, *i.e.*, the newly received sequence matches with at most one stored ReCord. Because the amount of neighbors within the wireless proximity of a node is finite, and each node generates the ReCord signature randomly, the number of stored ReCords that have identical cyclic shift offset is unlikely to far exceed the length of the signature. Therefore, the time complexity of RMix-2 is dominated by the process of getting potential cyclic shift offset, which is $O(L_1^2)$.

V. Evaluation

We have conducted comprehensive experiments to evaluate the performance of ReCorder on our USRP-N210 testbed. In this section, we first elaborate the setups of our experiments. Then, we present the evaluation results.

A. Experiment Setup

We first evaluate the performance of RCover preamble and ReCord signature, respectively. In each set of experiments, we use one USRP node as the sender of neighbor discovery messages, and another node as the receiver. Different pairs of USRP nodes are used to acquire different ReCord signatures. An interfering node is added, which keeps sending random OFDM signals. We note that all the three nodes work on the 2.4GHz spectrum band, and use the same 20MHz bandwidth unless specified otherwise. In different sets of experiments, we adjust the transmission gain and the placing of the third node to realize various SINR levels. However, it is still difficult to precisely control the SINR of neighbor discovery messages at the receiving node over the air. Therefore, in each set of experiments, we collect 500 samples of neighbor discovery messages, and take their average SINR as the SINR level for the whole set. For comparison, we also implement OFDM beacon transmission and decoding. Specifically, each beacon uses the convolutional coding rate of $1/2$, and is modulated by BPSK, corresponding to 6Mbps in IEEE 802.11a.

Furthermore, we implement ReCorder and the beaconing mechanism with various neighbor discovery protocols using our testbed prototype, including Disco [12], U-Connect [25], Searchlight [4], Hello [44] and Diff-Code [34]. To be specific, four USRP nodes are set up to transmit ReCorder's neighbor discovery messages or OFDM beacons using different neighbor discovery protocols, and an additional node is set to provide interfering signals. Then, we collect the traces at one receiver, which also operates according to the schedule of the same neighbor discovery protocol as the four neighboring nodes, and compute the discovery latency from the traces. The cumulative distribution of the latencies to discover those four neighbors at the receiver over 200 runs are presented.

Then, we evaluate ReCorder's impact on the decoding of other 802.11a/g OFDM packets. For that purpose, two links are established: one is for 800-byte OFDM packet transmissions, and the other is for neighbor discovery using ReCorder. To investigate the change of OFDM packet decoding rate under different SINR of ReCorder, we fix the OFDM link and adjust the transmission gain on the other link for neighbor discovery, which transmits discovery messages continuously. We note that different SINR levels for neighbor discovery also reflect its different extents of inference on OFDM. What's more, we set the bandwidth of OFDM to be 20 MHz, and evaluate two different bandwidths of ReCorder, which are 10 and 20 MHz.

We also use simulations to examine the influence of ReCorder-based low duty-cycled neighbor discovery on the throughput of OFDM data streams. On one hand, we consider a background fixed-rate UDP flow between a pair of nodes, transmitting 1500-byte packet under various rates. On the

TABLE I
VALUE OF PARAMETER β_1

RCover Length	β_1
63	0.05
127	0.03
255	0.017

other hand, we set a clique of nodes conducting neighbor discovery in each simulation, using time slots with duration of 20 ms. All of such nodes can interfere with the UDP transmissions. Besides, they conform to symmetric duty cycle based on Searchlight [4] for illustration. A UDP datagram can be successfully received only if it reaches the receiver without interference from concurrent discovery messages for a whole transmission time, and is regarded as lost otherwise. Specifically, we select two different discovery message lengths, which are of 253 ($L_C = 127$, $L_1 = L_2 = 63$) and 381 ($L_C = L_1 = L_2 = 127$) symbols, respectively. Their respective transmission time are 12.65 and 19.05 μs. Due to hardware jitter, we conservatively take the transmission time of a discovery message to be 14 or 20 μs. Similarly, a 1500-byte packet can be transmitted within 333.3 μs under 36 Mbps, corresponding to 335 μs plus the jitter. For comparison, we compute the average UDP packet loss ratio over 1000 trials with each simulation setup. The time slot boundaries and indexes of different neighboring nodes are randomly generated each time.

In the end of this section, we evaluate the interference within the node clique for neighbor discovery for comprehensiveness. Because two neighbor discovery messages will collide with each other if their interval is shorter than the transmission time of a single discovery message, the shorter discovery message in ReCorder should be able to mitigate the interference among neighboring nodes, especially in a crowded clique. Thus, we set a node A, together with different number of neighbors, for symmetric neighbor discovery based on Searchlight as above. The slot alignment of these nodes are randomly generated in each simulation, again using a 20-ms slot width. We use the same discovery message transmission time for ReCorder as above. For comparison, the transmission time of an OFDM beacon is set to be 50 μs.

B. Experiment Results

1) Robustness of ReCorder: In the detection of RCover, we focus on the probability of false negatives. The length of RCovers is set to 63, 127, and 255, respectively. In the calculation of detection threshold in Equation (10), we set the value of β_1 with respect to L_C in our experiments as in Table I. Moreover, the value of β_2 is tuned within the range $[0.5, 3.5]$ according to the energy level of the received symbols, *i.e.* higher energy level leads to smaller β_2. We should note that the provided value of β_1 and β_2 are specifically tuned with the USRP nodes used in our experiments. We select their value while trying to avoid false positives in the process of correlation detection.[4]

In Fig. 2, we present the false negative probability of RCover detection changing with the average SINR. It can be observed that, under the same SINR level, longer RCover sequence has smaller false negative probability. For example, under -6dB, 3.5% samples of 127-symbol RCover are missed, while the probability increases to 54.5% for 63-symbol RCover. For RCover with 255 symbols, the false negative probability even stays at 0% when the SINR comes to -8dB. Although longer RCover sequences can bring stronger robustness, they inevitably induce more transmission overheads. According to Fig. 2, 127-symbol RCover can realize a satisfying compromise between robustness and transmission overheads. In addition, among all the experiments, we only come across one instance of false positive with 63-symbol RCover under the SINR of 0dB.

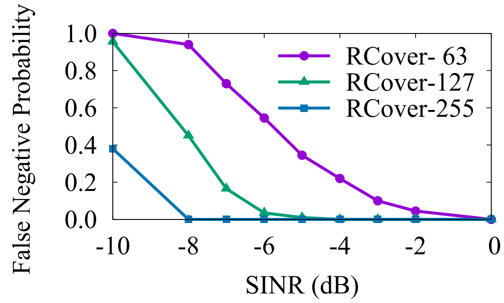


Fig. 2. RCover detection: False negative probability.

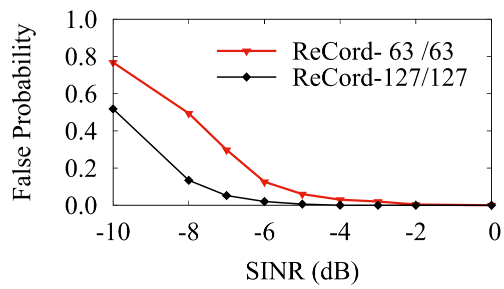


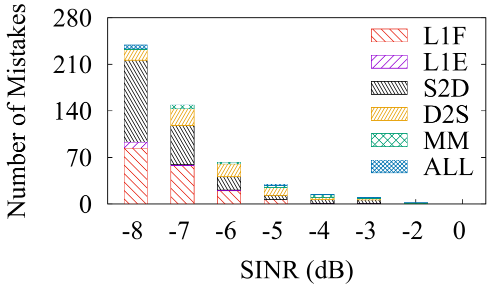
Fig. 3. ReCord false recognization probability.

Furthermore, we implement ReCord signatures with $L_1 = L_2 = 63$, and $L_1 = L_2 = 127$, respectively. For each setup of signature length, we pick 24 cyclic shift offsets on level-1 to generate 48 different ReCord signatures. Each signature is repeated by 10 times. Besides, we add 20 sequences of random symbols in the experiments to examine ReCord's resistance to noise and interference. All the 500 signatures are transmitted in random order. The results of false recognization probability are shown in Fig. 3. We can observe that the probabilities of false recognizations are relatively lower than false negatives in ReCord detection. For example, when $L_1 = L_2 = 63$, there are as few as 6.0% false recognizations under -5dB. While for ReCord-127/127, the false recognization probability is 5.2% under -7dB, and less than 2% for higher SINR value.

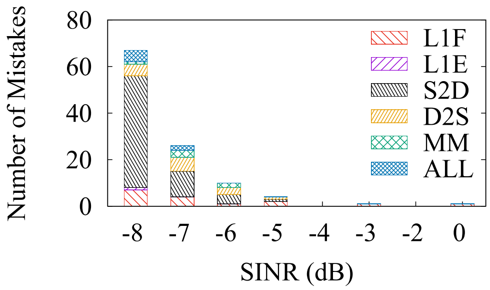
Specifically, we investigate the detailed false recognization reasons. There are six types of false recognizations in the evaluation, including (1) L1F: discard due to matching failure on the first level, and (2) L1E: matching to the wrong cyclic shift offset on the first level, and (3) S2D: mistaking a stored neighbor for a new one owing to level-2 un-matching, and (4) D2S: falsely matching two different ReCords as the same one on the second level, and (5) MM: discard of signatures due to multiple matchings, and (6) ALL: discard owing to mismatching on the whole signature level. Among these six

[4] The appropriate threshold setup for practical implementation should consider and maybe formulate the effect of signal normalization of radio hardwares [46], [53], which involves the tradeoff between false positives and false negatives. We leave the theoretical and mathematical analysis to future work.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a)



(b)

Fig. 4. ReCord false recognization details. (a) ReCord-63/63. (b) ReCord-127/127.

types, L1F, MM and ALL will increase the discovery latencies, and the type of L1E may lead to the discovery of un-existed neighbors. In addition, S2D results in duplicated discovery, while D2S induces unnecessary discard of already discovered neighbors. We present the detailed results in Fig. 4(a) and 4(b). The figures show that when the SINR is no lower than -6dB, all the five types of false recognizations are rare (*i.e.*, no more than 10%), which only appear when the signature sequences happen to be canceled by the interference.

More specifically, the 20 random symbol sequences are correctly discarded when the SINR is at least -5dB for L_1=L_2=63, and -7dB for L_1=L_2=127. In all the experiments, the random symbol sequences are falsely recognized as ReCords for less than 10 times. That shows the strong resistance of the RMix-2 algorithm to false positives. It implies that we can allow false positives to a limited extent in RCover detection, which reduces the false negative possibility without impairing the performance of ReCorder.[5] In addition, even when the level-1 identity is correctly identified, there are still comparable number of level-2 mistakes (S2D, D2S). This is due to the inferior correlation property of level-2 ReCord.

Till now we have obtained the false probabilities for both RCover detection and ReCord recognization. In Fig. 5, we integrate the above results, and demonstrate the false probabilities of ReCorder. Note that we do not use the 255-symbol RCover preamble because of its high transmission overhead. By comparison, we measure the packet error rate of 30-byte OFDM packets. Apparently, because the false recognization probabilities of ReCord signatures are extremely low, the performance of ReCorder is dominated by RCover detection. Compared with OFDM, any combination of RCover preamble and ReCord signature performs significantly better.

[5]When a false positive escapes from both RMix and RMix-2 algorithms, they can ultimately be detected through an after-discovery phase, as discussed in Section VI-D.

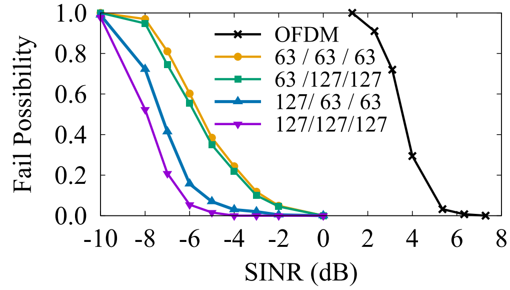


Fig. 5. Comparison of false probabilities: ReCorder *vs.* OFDM beacons.

TABLE II
SEARCHLIGHT: DISCOVERY LATENCIES COMPARISON

Protocol	ReCorder	Beacon	ReCorder	Beacon
SINR	3 dB	3 dB	-5 dB	5 dB
Median Case	319	1018	332	335
Wosrt Case	707	6297	759	787

To be specific, when ReCorder uses a 127-symbol RCover and a $(63+63)$-symbol ReCord, it can guarantee the false probabilities of less than 10% and 20% at the SINR of -5dB and -6dB, respectively. By contrast, OFDM misses 29.6% packets at the SINR of 4dB. Thus, ReCorder can achieve a robustness gain of nearly 10dB in terms of SINR compared with the beaconing and decoding mechanism in existing works. We conclude that ReCorder with $L_C = 127$ and $L_1 = L_2 = 63$ can realize a good compromise between robustness and transmission overheads in practice.

It should be pointed out that the SINR of -6dB happens when the receiving device is close to an interfering transmitter. For data frame transmission, the receiving device can use carrier sensing combined with RTS-CTS to contend for media access. However, it is a high overhead for duty-cycled neighbor discovery. Therefore, the robustness under such a low SINR is necessary.

2) Cases of Applications: We compare ReCorder with the beaconing and decoding mechanism by implementing five state-of-the-art neighbor discovery protocols. We evaluate the symmetric duty cycle of 5%. The SINR is set to be -5dB and 3dB for ReCorder, and 3dB and 5dB for OFDM. The cumulative distributions of discovery latencies are shown in Fig. 6. We can see that for each neighbor discovery protocol, ReCorder at -5dB can achieve almost the same performance as the OFDM beacon-decoding mechanism at 5dB, while ReCorder outperforms OFDM beacon-decoding tremendously when they both work under 3dB. For illustration, we summarize the median and worst-case latencies of Searchlight in different cases in Table II. Its discovery latencies with ReCorder at -5dB are comparable to those with OFDM at 5dB. That is because even at the SINR of -5dB, the false probability of ReCorder is similar to the packet error rate of OFDM beacons at 5dB. However, under the SINR of 3dB, ReCorder's median and worst-case gains are as high as 68.7% and 88.8%, respectively, which again suggest the necessity of robust neighbor discovery in mobile wireless networks. More importantly, in practice, the smaller discover latency leads to the reduction of energy consumption for neighbor discovery.

Then, we present how the decoding of OFDM packets is impacted by neighbor discovery messages of ReCorder

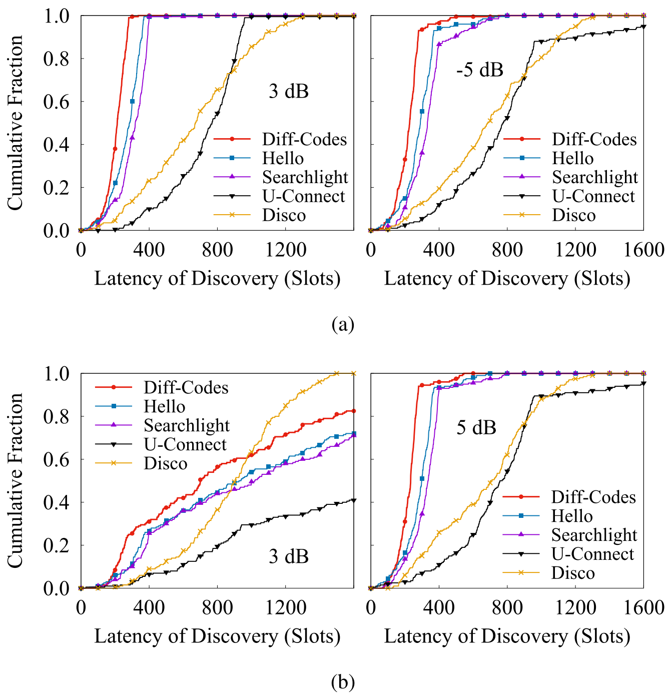


Fig. 6. CDF of discovery latencies for symmetric duty cycle 5%. (a) ReCorder. (b) OFDM beacons.

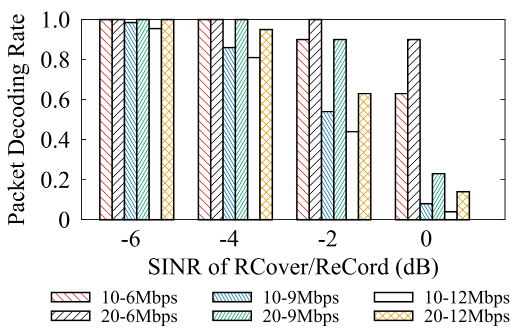


Fig. 7. ReCorder's impact on OFDM packets.

in Fig. 7. We carry out the experiment with three data rates in IEEE 802.11 a/g. When ReCorder and OFDM have the same bandwidth of 20 MHz, at -6dB SINR for ReCorder, the data rates of 6Mbps, 9Mbps and 12Mbps experience no decoding degradation. When the SINR of ReCorder is increased, the performance of OFDM with lower data rates are relatively stable. As for higher rates, the decoding of OFDM packets may be impeded due to neighbor discovery messages. For instance, the packet decoding rate of 12Mbps is below 15% when the SINR of ReCorder is as high as 0dB. Therefore, ReCorder can avoid its impact on OFDM packet decoding at low data rates (*i.e.*, 6Mbps and 9Mbps), while still achieving robust performance. As explained in Section VI, that is important for the co-existence of ReCorder and background OFDM transmissions. What's more, from the experiment results of ReCorder with 10 MHz bandwidth, we validate that the impact of ReCorder on OFDM packet decoding can be mitigated by occupying higher bandwidth.

However, in practice, considering the low duty cycle of neighbor discovery and the short message length in ReCorder, not all the data packets of a background stream will be lost owing to the interference of discovery messages. Fig. 8 presents the average packet loss ratio of a UDP flow with the existence of neighbor discovery using the 253-symbol discovery message. As shown in the figure, we compare two symmetric duty cycles for neighbor discovery, and two IEEE 802.11 data rates for UDP transmissions. We can observe that the packet loss probability is positively correlated with both the duty cycle and the clique size of neighbor discovery, and is negatively correlated with the data rate of background data transmissions. To be detailed, a higher duty cycle, as well as a larger clique size, produces more frequent discovery messages in the wireless communication media, and thus, more interference to the UDP flow; while a larger data rate reduces the transmission time of each UDP datagram, which makes it less vulnerable to the interfering discovery messages. Moreover, according to Fig. 8, the degradation of the UDP throughput is limited, *e.g.*, even when there are 50 nodes conducting neighbor discovery, the UDP packet loss ratio still falls below 8.36%, and is as low as 1.17% provided the symmetric duty cycle of 1% and the IEEE 802.11 data rate of 54 Mbps.

To further explore the influence of discovery message length on background UDP transmissions with various rates, we fix a 50-node clique and the symmetric duty cycle of 1%, and get the results as in Fig. 9. We find that changing from the 253-symbol discovery message to the 381-symbol one only slightly increases the UDP packet loss probability. Besides, there are similar results when the UDP flow has a larger rate. Specifically, the packet loss ratio almost stays the same after the UDP rate increases over 4 Mbps. This is because the packet interval for the UDP flow is relatively large compared with the low duty cycle of neighbor discovery messages.

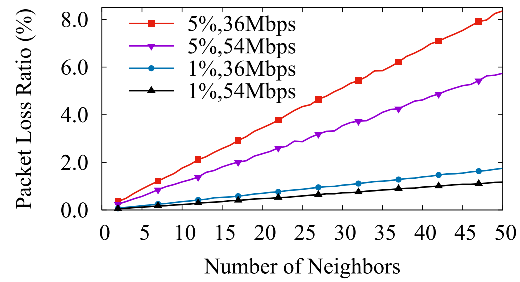


Fig. 8. UDP packet loss ratio under increasing neighbor discovery clique size, with 253-symbol discovery message.

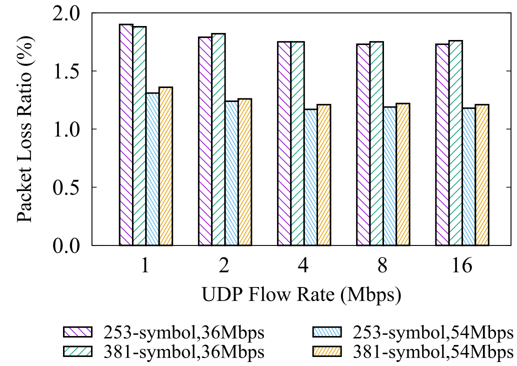


Fig. 9. UDP packet loss ratio under different discovery message lengths and 802.11 rates, with 50 neighboring nodes and 1% symmetric duty cycle.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

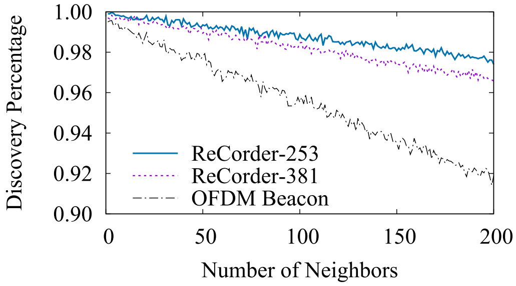


Fig. 10. Percentage of discovered neighboring nodes.

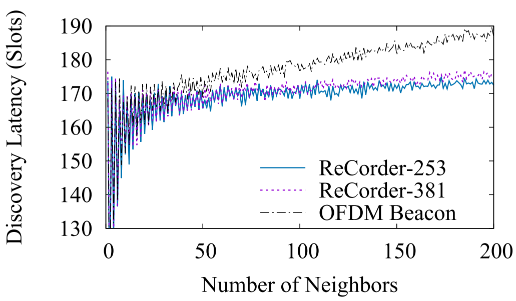


Fig. 11. Latencies to discover 50% of neighbors.

At last, besides interfering with background wireless transmissions, neighbor discovery messages may also collide with each other, *i.e.*, two nodes cannot accomplish mutual discovery, if the interleaving of their slot boundaries is smaller than the transmission time of the discovery message. Fig. 10 presents the average percentage of neighbors that a node can discover over 1000 trials, with the symmetric duty cycle of 5%. Although the beacon decoding-based mechanism also guarantees a discovery proportion of more than 90% even when there are 200 neighbors, ReCorder can increase the percentage by at least 5%.

From the perspective of discovery latency, less collisions between discovery messages enable a node to discover a specific number of neighbors with shorter latency. To illustrate that, we give the average latencies to discover at least half of all the neighboring nodes in Fig. 11. It can be seen that as the size of the neighbor discovery clique increases, the reduction of such latencies gradually approaches 7%. However, we should note that in common cases, there are not that many neighboring nodes, and the above difference between ReCorder and the beacon-decoding based mechanism regarding the collisions between discovery messages may be unapparent. Additionally, the fluctuations in Fig. 11 are due to the fact that the number of 50% neighboring nodes may not increase when the overall neighbors increase. For example, a node needs to discover 3 nodes when it has both 5 and 6 neighboring nodes. In that case, the discover latency with 6 neighbors tends to be smaller, leading to the latency fluctuations.

VI. Discussion

In this section, we discuss some important practical issues on the implementation of ReCorder.

A. Signature Collision

As mentioned in Section IV-C, the level-1 ReCord adopts Gold code of length L_1. Therefore, the number of distinct level-1 signatures also equals L_1 [14], [31]. Due to Birthday paradox, ReCorder may suffer from collisions of level-1 ReCord signatures, especially in a relatively congested proximity. To deal with that, we can include multiple sequences in a single level-1 signature, *i.e.*, a node randomly picks m ($m \geq 1$) cyclic shift offsets of the same Gold code to form its level-1 identity. Then, two neighboring nodes will not be mixed up, unless they select the same m offsets. In that case, a larger m leads to a smaller collision probability. To formalize, given N nodes, the probability of level-1 signature collision is,

$$P_{l_1}(N, m, L_1) = 1 - \binom{L_1^m}{N} \cdot \left(\frac{1}{L_1^m}\right)^N .$$

According to the above equation, provided the level-1 ReCord with $L_1 = 127$ and a network with $N = 50$ nodes, we have $P_{l_1}(50, 1, 127) = 99.9987\% \approx 1$ and $P_{l_1}(50, 2, 127) = 7.321\%$. Obviously, the number of level-1 collisions can be significantly reduced by including multiple shifted sequences, *e.g.*, by picking 2 shift offsets, the collision probability can be restricted to an acceptable range.

B. Energy Efficiency

ReCorder reduced the length of neighbor discovery messages compared with the beacons used by existing protocols. In IEEE 802.11a/g, under the bit rate of 6Mbps, a 30-byte packet will produce about 1000 complex samples including the packet preamble. By contrast, ReCorder performs well with the 253-symbol symbol sequence as neighbor discovery message when $m = 1$, and 316-symbol when $m = 2$ ($L_1 = L_2 = 63$). The shorter neighbor discovery messages consumes at least $2/3$ less transmission energy on the sending side. Furthermore, in IEEE 802.11a/g (OFDM), a node needs the process of FFT$^{(-1)}$/FFT to transmit or receive a beacon. ReCorder can save such computation resources. On the receiving side, compared with the decoding of OFDM beacons, which should also be based on the correlation of packet preamble during beacon detection, ReCorder only conducts correlation, and eliminates the CPU overhead from packet decoding.

Besides, as demonstrated in Section V-B2, the reduced message length in ReCorder brings an additional effect, *i.e.*, it can lower the collision probabilities among multiple neighboring nodes, which, together with the increased robustness, leads to less neighbor discovery message losses. That can effectively reduce the discovery latency, as well as the energy consumption in practice.

Furthermore, although we have adopted simple BPSK modulation for both RCover and ReCord, these two sequences can actually be modulated together as the I and Q components, respectively. That is illustrated in Fig. 12, where $L_C = L_1 + L_2$. In such case, the correlation processes for RCover detection and ReCord recognization need to be conducted separately on the basis of I and Q signals. However, because of the phase distortion of wireless channel and hardware, we cannot correlate the received I or Q signals directly. Fortunately, because of the relatively small sequence length with respect to the coherence time of the wireless channel, all the samples within a single sequence tend to go through similar phase distortion. Thus, we can utilize the known RCover preamble to estimate the phase offset, which is similar to the channel estimation mechanism in existing

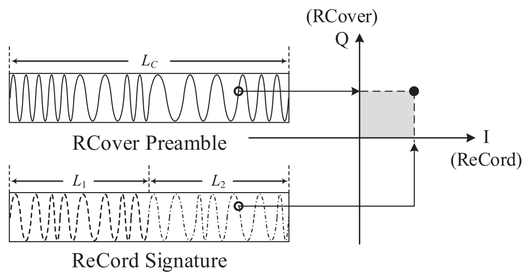


Fig. 12. IQ modulation of RCover and ReCord.

WiFi communication system.[6] Such *I/Q-decoupled* correlation reduces the length of neighbor discovery messages, which contributes to higher energy efficiency, as well. We leave it to the future work due to limited space. Since BPSK and possible QPSK modulations already achieve satisfying energy efficiency, we do not consider higher-order modulations, which may have lower tolerance to interference.

C. Co-Existence With Concurrent Transmissions

It is expected that neighbor discovery messages of ReCorder should not impact other background streams. According to existing works, it is not a concern for Direct-Sequence Spread Spectrum (DSSS) based physical layer standard, such as IEEE 802.15.4 and 802.11b. To be specific, Wu *et al.* [52] have shown that interference with short duration will not affect other data transmissions obviously, provided the redundant tolerance in the physical layer implementations. When it comes to the OFDM standard that is widely adopted in wireless networks (*e.g.*, IEEE 802.11a/g), we have shown by the experiments that ReCorder does not impact the decoding of BPSK modulated OFDM packets. In fact, this is crucial for the co-existence between neighbor discovery and background transmissions. On one hand, ReCorder can directly co-exist with low bit-rate WiFi control and management frames, so that it does not harm the regular operations of a WiFi network (*e.g.*, WiFi client-AP association), even though it may induce data packet loss. On the other hand, because of the low duty cycle, the neighbor discovery messages are only transmitted infrequently within a specific proximity. Therefore, the data packet loss due to ReCorder appears as a form of random wireless packet loss. Considering that the state-of-the-art TCP congestion control architecture [11] and works on multipath TCP [21], [51], which can resist random packet loss, and maintain the end-to-end throughput without additional hardware support, the co-existence between ReCorder and background TCP transmissions should be similarly promising compared with the UDP co-existence as presented in Section V-B2. Furthermore, we can implement quite a few existing works such as rateless code (*e.g.*, [20], [37]) and partial packet recovery (*e.g.*, [22]), to rescue those collided data packets with neighbor discovery messages, which can further reduce the impact of neighbor discovery by ReCorder.

Besides, the bandwidths employed by neighbor discovery messages and OFDM packets can affect their co-existence, as well. Provided the same transmission power, if ReCorder uses a smaller bandwidth, it will induce larger interference on OFDM packets near the center frequency in the frequency domain. So ReCorder should use at least the same bandwidth as OFDM to minimize its impact on OFDM packet decoding. Moreover, recent works on downclocking the OFDM [16] have provided the potential to enable ReCorder to occupy higher bandwidth than OFDM, in which case its impact on the decoding of background OFDM packets is further reduced.

[6]We do not need to consider such phase offset when RCover and ReCord are modulated separately, because we can calculate the correlation magnitude in such case, as presented in Section IV, which only induces limited penalty on the processing gain [31].

D. Practical Implementation

The hardware implementation of ReCorder involves two aspects. First, it needs to adopt the active-sleep schedule of some specific neighbor discovery protocol. According to existing works [12], [35], the neighbor discovery protocol can be implemented using slot counter on basis of low-power listening protocols (*e.g.*, [39]). Second, the detection of RCover and recognition of ReCord require the existence of signal filters and correlators, which are readily available in current off-the-shelf 802.11 chipsets. Therefore, similar to other works utilizing cross-correlation (*e.g.*, [31]), ReCorder can be implemented by replicating existing components in present chipsets.

Moreover, ReCorder uses a pseudo-random ReCord signature to distinguish neighboring nodes. However, before two neighbors can start packet transmissions, they still need the MAC address of each other. To bridge that gap, upon each successful discovery, a node can keep its wireless interface on for several slots, and send beacons at the same time. Because the node conducts neighbor discovery with a low duty cycle, an after-discovery mechanism like that will induce restricted energy overheads. Besides, such an after-discovery mechanism can be used as the last step in filtering out detection false positives. In addition, a node can even switch between ReCorder and the traditional beaconing mechanism based on the existence of interfering signals.

VII. Conclusion and Future Work

In this work, we have designed ReCorder for practical and robust neighbor discovery. We have established a novel structure for neighbor discovery messages instead of using beacons as existing works. To be specific, each neighbor discovery message is distinguished from other data packets by a pre-defined preamble named RCover. Each sender of neighbor discovery messages has a unique ReCord identity signature. Both RCover and ReCord are pseudo-random sequences, and can be recognized through cross-correlation by the RMix and RMix-2 algorithms, respectively. ReCorder not only eliminates the decoding of beacons in existing works, but also reduces the length of neighbor discovery messages by nearly $2/3$. Furthermore, we have prototyped ReCorder using USRP-N210. The evaluation results show that compared with the beacon-decoding mechanism, ReCorder can realize a 10dB gain of robustness in terms of SINR. In addition, ReCorder can avoid impairing the decoding of management and control frames in the 802.11 networks, which facilitates its co-existence with background wireless transmissions. Its influence on background UDP throughput is also restrictive (*e.g.*, as low as 1.17% reduction even with 50 neighboring nodes). In the future, we will further improve the robustness of ReCorder by exploring the similarities of multiple neighbor discovery messages to construct better correlation structures [13], and to improve the evidence of a neighboring node's identity.

Acknowledgments

The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

References

[1] *Sony PS Vita-Near*. Accessed: 2017. [Online]. Available: http://us.playstation.com/psvita/

[2] *Wi-Fi Direct*. Accessed: 2017. [Online]. Available: http://www.wi-fi.org/discover-and-learn/wi-fi-direct/

[3] *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11-2012, 2012.

[4] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: Won't you be my neighbor?" in *ACM MobiCom*, 2012, pp. 185–196.

[5] T. Bansal, B. Chen, P. Sinha, and K. Srinivasan, "Symphony: Cooperative packet recovery over the wired backbone in enterprise WLANs," in *Proc. ACM MobiCom*, 2013, pp. 351–362.

[6] M. Bradonjić, E. Kohler, and R. Ostrovsky, "Near-optimal radio use for wireless network synchronization," in *Proc. ALGOSENSORS*, 2009, pp. 15–28.

[7] L. Chen, K. Bian, and M. Zheng, "Never live without neighbors: From single- to multi-channel neighbor discovery for mobile sensing applications," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 3148–3161, Oct. 2015.

[8] L. Chen, Y. Li, and A. V. Vasilakos, "Oblivious neighbor discovery for wireless devices with directional antennas," in *Proc. IEEE INFOCOM*, Apr. 2016, pp. 1–9.

[9] S. Chen *et al.*, "Asynchronous neighbor discovery on duty-cycled mobile devices: Integer and non-integer schedules," in *Proc. ACM MobiHoc*, 2015, pp. 47–56.

[10] A. Chouly, A. Brajal, and S. Jourdan, "Orthogonal multicarrier techniques applied to direct sequence spread spectrum CDMA systems," in *Proc. IEEE GLOBECOM*, Nov. 1993, pp. 1723–1728.

[11] M. Dong, Q. Li, D. Zarchy, B. Godfrey, and M. Schapira, "PCC: Re-architecting congestion control for consistent high performance," in *Proc. USENIX NSDI*, 2014, pp. 395–408.

[12] P. Dutta and D. E. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. ACM SenSys*, 2008, pp. 71–84.

[13] C. F. M. E. Silva *et al.*, "Network-assisted neighbor discovery based on power vectors for D2D communications," in *Proc. VTC Spring*, 2015, pp. 1–5.

[14] P. Fan and M. Darnell, *Sequence Design for Communications Applications*. Taunton, U.K.: Research Studies Press, 1996.

[15] B. Feng, C. Zhang, L. Wei, and Y. Fang, "Piggybacking lightweight control messages on physical layer for multicarrier wireless LANs," in *Proc. WASA*, 2016, pp. 68–79.

[16] F. Lu, P. Ling, G. M. Voelker, and A. C. Snoeren, "Enfold: Downclocking OFDM in WiFi," in *Proc. ACM MobiCom*, 2014, pp. 129–140.

[17] L. Galluccio, G. Morabito, and S. Palazzo, "Analytical evaluation of a tradeoff between energy efficiency and responsiveness of neighbor discovery in self-organizing ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 7, pp. 1167–1182, Sep. 2004.

[18] V. Galluzzi and T. Herman, "Survey: Discovery in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 1, p. 271860, 2012.

[19] S. Gollakota and D. Katabi, "Zigzag decoding: Combating hidden terminals in wireless networks," in *Proc. ACM SIGCOMM*, 2008, pp. 159–170.

[20] A. Gudipati and S. Katti, "Strider: Automatic rate adaptation and collision handling," in *Proc. ACM SIGCOMM*, 2011, pp. 158–169.

[21] B. Han, F. Qian, S. Hao, L. Ji, and N. Bedminster, "An anatomy of mobile Web performance over multipath TCP," in *Proc. ACM CoNEXT*, 2015, Art. no. 5.

[22] K. Jamieson and H. Balakrishnan, "PPR: Partial packet recovery for wireless networks," in *Proc. ACM SIGCOMM*, 2007, pp. 409–420.

[23] X. Ji *et al.*, "On improving wireless channel utilization: A collision tolerance-based approach," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 787–800, Mar. 2017.

[24] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for IEEE 802.11 ad hoc networks," *Mobile Netw. Appl.*, vol. 10, nos. 1–2, pp. 169–181, 2005.

[25] A. Kandhalu, K. Lakshmanan, and R. Rajkumar, "U-Connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proc. ACM IPSN*, 2010, pp. 350–361.

[26] N. Karowski, A. C. Viana, and A. Wolisz, "Optimized asynchronous multi-channel neighbor discovery," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 536–540.

[27] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1998.

[28] R. Khalili, D. L. Goeckel, D. Towsley, and A. Swami, "Neighbor discovery with reception status feedback to transmitters," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[29] S. Lai, B. Ravindran, and H. Cho, "Heterogenous quorum-based wake-up scheduling in wireless sensor networks," *IEEE Trans. Comput.*, vol. 59, no. 11, pp. 1562–1575, Nov. 2010.

[30] C. Lott, O. Milenkovic, and E. Soljanin, "Hybrid ARQ: Theory, state of the art and future directions," in *Proc. IEEE Inf. Theory Workshop Inf. Theory Wireless Netw.*, Jul. 2007, pp. 1–5.

[31] E. Magistretti, O. Gurewitz, and E. W. Knightly, "802.11ec: Collision avoidance without control messages," in *Proc. ACM MobiCom*, 2012, pp. 65–76.

[32] R. Margolies, G. Grebla, T. Chen, D. Rubenstein, and G. Zussman, "Panda: Neighbor discovery on a power harvesting budget," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3606–3619, Dec. 2016.

[33] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. ACM MobiHoc*, 2001, pp. 137–145.

[34] T. Meng, F. Wu, and G. Chen, "On designing neighbor discovery protocols: A code-based approach," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 1689–1697.

[35] T. Meng, F. Wu, and G. Chen, "Code-based neighbor discovery protocols in mobile wireless networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 806–819, Apr. 2016.

[36] E. Perahia and R. Stacey, *Next Generation Wireless LANs*. Cambridge, U.K.: Cambridge Univ. Press, 2013.

[37] J. Perry, P. Iannucci, K. Fleming, H. Balakrishnan, and D. Shah, "Spinal codes," in *Proc. ACM SIGCOMM*, 2012, pp. 49–60.

[38] R. L. Peterson, R. E. Ziemer, and D. E. Borth, *Introduction to Spread-Spectrum Communications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.

[39] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM SenSys*, 2004, pp. 95–107.

[40] M. A. Richards, *Fundamentals of Radar Signal Processing*. New York, NY, USA: McGraw-Hill, 2005.

[41] S. E. Roslin, C. Gomathy, and P. Bhuvaneshwari, "A survey on neighborhood dependant topology control in wireless sensor networks," *Int. J. Comput. Sci. Commun.*, vol. 1, no. 1, pp. 185–188, 2010.

[42] S. Sen, R. R. Choudhury, and S. Nelakuditi, "CSMA/CN: Carrier sense multiple access with collision notification," *IEEE/ACM Trans. Netw.*, vol. 20, no. 2, pp. 544–556, Apr. 2012.

[43] M. Spuhler, D. Giustiniano, V. Lenders, M. Wilhelm, and J. B. Schmitt, "Detection of reactive jamming in DSSS-based wireless communications," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1593–1603, Mar. 2014.

[44] W. Sun, Z. Yang, K. Wang, and Y. Liu, "Hello: A generic flexible protocol for neighbor discovery," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 540–548.

[45] W. Sun, Z. Yang, X. Zhang, and Y. Liu, "Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1448–1459, 3rd Quart., 2014.

[46] A. Swaminathan and D. L. Noneaker, "A technique to improve the performance of serial, matched-filter acquisition in direct-sequence spread-spectrum packet radio communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 5, pp. 909–919, May 2005.

[47] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks," *Comput. Netw.*, vol. 43, no. 3, pp. 317–337, 2003.

[48] S. Vasudevan, M. Adler, D. Goeckel, and D. Towsley, "Efficient algorithms for neighbor discovery in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 1, pp. 69–83, Feb. 2013.

[49] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 2502–2512.

[50] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *Proc. ACM MobiCom*, 2009, pp. 181–192.

[51] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," in *Proc. USENIX NSDI*, 2011, pp. 1–14.

[52] K. Wu *et al.*, "Side channel: Bits over interference," in *Proc. ACM MobiCom*, 2010, pp. 13–24.

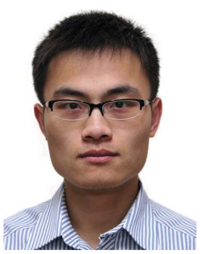
[53] Z. Xiao, D. Jin, and N. Ge, "GLRT for packet detection with practical analog AGC," *IEEE Trans. Veh. Technol.*, vol. 63, no. 4, pp. 1749–1758, May 2014.

[54] W. Zeng *et al.*, "Neighbor discovery in wireless networks with multi-packet reception," in *Proc. ACM MobiHoc*, 2011, Art. no. 3.

[55] X. Zhang and K. G. Shin, "E-MiLi: Energy-minimizing idle listening in wireless networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 9, pp. 1441–1454, Sep. 2012.

[56] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proc. ACM MobiHoc*, 2003, pp. 35–45.

Fan Wu received the B.S. degree in computer science from Nanjing University in 2004, and the Ph.D. degree in computer science and engineering from the State University of New York at Buffalo in 2009. He has visited the University of Illinois at Urbana–Champaign as a Post-Doctoral Research Associate. He is currently a Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He has published over 150 peer-reviewed papers in technical journals and conference proceedings. His research interests include wireless networking and mobile computing, algorithmic game theory and its applications, and privacy preservation. He was a recipient of the first class prize for the Natural Science Award of the China Ministry of Education, the NSFC Excellent Young Scholars Program, the ACM China Rising Star Award, the CCF-Tencent Rhinoceros bird Outstanding Award, the CCF-Intel Young Faculty Researcher Program Award, the Pujiang Scholarship, and the Tang Scholarship. He has served as the Chair for CCF YOCSEF Shanghai, on the Editorial Board of *Computer Communications*, and as a member for technical program committees of over 60 academic conferences.



Tong Meng received the B.S. degree in computer science from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, in 2013. His research interests include neighbor discovery, routing in wireless networks, and mobile social networks. He is a Student Member of ACM and CCF.

Aijing Li (M'15) received the Ph.D. degree in computer science and technology from the PLA University of Science and Technology, Nanjing, China. She joined the College of Communications Engineering, PLA University of Science and Technology, in 2015. Her current research interests include wireless communications, cognitive radio networking, and heterogeneous networking.

Guihai Chen received the B.S. degree from Nanjing University, the M.E. degree from Southeast University, and the Ph.D. degree from The University of Hong Kong. He visited the Kyushu Institute of Technology, Japan, in 1998, as a Research Fellow, and the University of Queensland, Australia, in 2000, as a Visiting Professor. From 2001 to 2003, he was a Visiting Professor with Wayne State University. He is currently a Distinguished Professor and a Deputy Chair with the Department of Computer Science, Shanghai Jiao Tong University. He has published over 200 papers in peer-reviewed journals and refereed conference proceedings in the areas of wireless sensor networks, high-performance computer architecture, peer-to-peer computing, and performance evaluation. He is a member of the IEEE Computer Society. He has served on technical program committees of numerous international conferences.

Nitin H. Vaidya (F'11) received the Ph.D. degree from the University of Massachusetts at Amherst. He has held visiting positions at Microsoft Research, Sun Microsystems, and the IIT-Bombay. He is currently an Associate Professor of electrical and computer engineering with the University of Illinois at Urbana–Champaign. He has coauthored papers that received awards at the ACM MobiCom and Personal Wireless Communications conferences. His research has been funded by various agencies, including the U.S. National Science Foundation, DARPA, Motorola, Microsoft Research, and Sun Microsystems. His current research interests include wireless networking and mobile computing. He is a member of the ACM. He was a recipient of a CAREER Award from the U.S. National Science Foundation. He has served on the committees of several conferences, including as a Program Co-Chair for the 2003 ACM MobiCom and a General Chair for the 2001 ACM MobiHoc. He has served as an editor for several journals and has been serving as the Editor-in-Chief for the IEEE Transactions on Mobile Computing since 2005.