On Robust Neighbor Discovery in Mobile Wireless Networks

Tong Meng[†], Fan Wu[†]*, Aijing Li[§], Guihai Chen[†], Nitin H. Vaidya[‡]

[†]Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[§]PLA University of Science and Technology, China
[‡]Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, US
mengtong@sjtu.edu.cn, fwu@cs.sjtu.edu.cn, lishan.wh@gmail.com
gchen@cs.sjtu.edu.cn, nhv@illinois.edu

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 ReCorder has been evalueated using the USRP-N210 testbed. Our evaluation results show that ReCorder can achieve robust neighbor discovery at an SINR lower than the existing beaconing and decoding based neighbor discovery protocols by almost 10dB. Furthermore, ReCorder can avoid degrading the decoding of background IEEE 802.11a/g transmissions with BPSK modulation, which is important for its co-existence with concurrent wireless streams.

*Fan Wu is the corresponding author.

CoNEXT '15, Dec. 1–4, 2015, Heidelberg, Germany
 ISBN 0-12345-67-8/90/01…$15.00
DOI: 10.475/123_4

CCS Concepts

•**Networks** → **Mobile networks;** *Cross-layer protocols;*

Keywords

Neighbor Discovery; Wireless Networks; Smart Devices; Experimentation

1. 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, 22, 26]). 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 H-ARQ [18], 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. Moreover, just like people can recognize each other through tunes of voices, ReCorder uses well-defined signatures to distinguish different neighboring nodes. 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 expoiting the correlation property of pseudo-radom sequences, which contain a pratical estimation of the SINR level, as well.
- We analyze and investigate the influence of ReCorder on background OFDM transmissions using the IEEE 802.11a/g protocol. We conclude that ReCorder can co-exist with low bit-rate management frames, and minimize its impact on the decoding of OFDM packets by occupying at least the same bandwidth as OFDM. We also discuss the combination of existing works with OFDM and ReCorder to bootstrap their co-existence.
- 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. What's more, ReCorder enables shorter neighbor discovery messages, which is more energy-efficient with less transmission power and no decoding overheads, and brings no degradation to IEEE 802.11a/g protocol with BPSK modulation.

The rest of the paper is organized as follows. In Section 2, we discuss the related works. In Section 3, we introduce our motivation as well as the preliminary knowledge on wireless communication. The overview and design details of ReCorder are presented in Section 4, which is followed by the evaluation results in Section 5. Then, we discuss several practical issues in Section 6, and conclude the paper in Section 7.

2. RELATED WORKS

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

2.1 Neighbor Discovery Protocols

The problem of neighbor discovery has been extensively investigated in both sensor networks and mobile wireless networks. 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. What's more, 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 *et al.* [21] is the foundation of most of the following probabilistic neighbor discovery protocols (*e.g.*, [16, 27, 29]). 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 [13] and [17], each cycle of a node is regarded as a quorum. Zheng *et al.* [31] applied optimal block design. However, these works are mainly restricted to symmetric duty cycle. To overcome such a limitation, primed-based protocols, such as Disco [7] and U-Connect [14], implement primes to generate active-sleep pattern. Besides, Searchlight [4] and Hello [26] leverage the regular relation between the probing schedules of different nodes. Furthermore, by exploiting asynchronization, Meng *et al.* [22,23] derived the lower bound of discovery latency, and designed (A)Diff-Codes.

Most of previous neighbor discovery protocols make efforts to realize high energy and time efficiency, 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.

2.2 Related Works on Cross-Correlation

Cross-correlation is usually implemented to recognize some known pseudo-random sequences. For example, Sen *et al.* [25] proposed CSMA/CN that utilizes the correlation property of a pseudo-random signature to notify the detection of collision. Wu *et al.* [28] built a Side Channel for efficient medium access by the correlation of some intended patterns.

What's more, Zhang *et al.* [30] designed E-MiLi that enables downclocked radios through the correlation of M-preambles. Magistretti *et al.* [20] designed 802.11ec, and replaced the control messages in IEEE 802.11 with correlatable symbol sequences. Both [30] and [20] 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.

There are also works on cooperative packet recovery (*e.g.*, [5,10]) 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 [10] require controllable collisions, which does not fit the unpredictable interference in mobile wireless networks.

3. MOTIVATION AND PRELIMINARIES

In this section, we verify the necessity of designing robust neighbor discovery technique by showing a motivating example. The results demonstrate that the transmission of small size wireless packets with low packet generation rate, which is similar to the scenario of neighbor discovery, can be severely impacted by concurrent wireless flows. Then, we briefly introduce the preliminaries on wireless communication.

3.1 Motivation

Plenty of works (*e.g.*, [20, 25, 28]) have proven that the carrier sensing mechanism in the 802.11 networks cannot escape from interference. In the busy communication media of mobile wireless networks, concurrent wireless transmissions in the same collision domain can interfere with each other, leading to packet decoding degradations. For instance, one may suffer from poor WiFi accessing experience sometimes, even when she is the only client connected to a high-RSS router that is set up by herself. One of the underlying reasons is the interference by background wireless transmissions, which is common in mobile wireless networks.

What's more, two important characteristics of neighbor discovery in mobile wireless networks make it more vulnerable to interference, which are the short message length and the low duty cycle. As mentioned in Section 1, existing neighbor discovery protocols use beacons as neighbor discovery messages. A beacon usually has much smaller packet size than a regular data frame. In case of interference, a beacon tends to be shadowed by concurrent wireless packets, and cannot be decoded. Besides, neighbor discovery has low duty cycle owing to the limited battery power of mobile devices. A node sends beacons infrequently, while keeping asleep most of the time. Thus, 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.

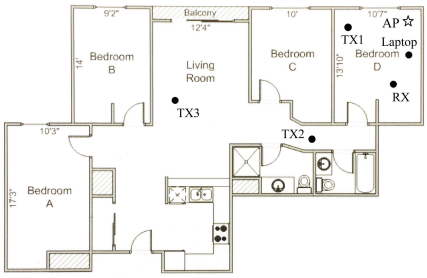


Figure 1: Floor Plan of An Apartment Room for Our Motivating Example

We give a motivating example in the environment of an apartment room as above to verify such vulnerability. As shown in Figure 1, we set up a wireless router in a bedroom, and connect a laptop to it. In addition, we establish a UDP tunnel between two smartphones connected through WiFi-Direct. One of the smartphones sends 30-byte packets (similar size as beacon) to the other at the rate of 1 packet per second. The receiving phone is placed in the same bedroom as the wireless router and the laptop, and three different places for the transmitting phone are considered. We download a large file on the laptop to provide interfering signals,[1] and close all the doors to create the effect of hidden terminal. We note that the experiments are conducted late at night to minimize the influence of wireless transmissions from neighboring rooms.

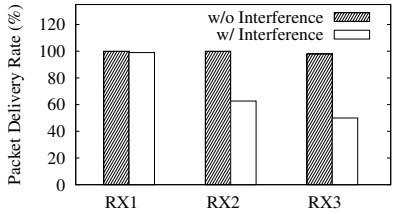


Figure 2: Packet Delivery Rate with and without Interference

Figure 2 presents the packet delivery rates between the two smartphones with and without interference, which are computed over more than 500 transmitted packets in each set of experiments. Apparently, the transmission between the two phones receives severe degradation with the existence of the downloading interference, *i.e.*, the packet delivery rates at RX2 and RX3 fall from nearly 100% to 62.7% and 50.0%, respectively.

[1]Throughout the experiments, the WiFi downloading speed does not exhibit any obvious fluctuations due to the UDP transmissions between the smartphones, because the laptop is close to the router. We will discuss the influence of neighbor discovery on background transmissions in Section 6.

The experiment results indicate that the beacon decoding based neighbor discovery protocols may indeed be impeded due to interfering transmissions, and a robust technique for neighbor discovery in practical mobile wireless networks is highly needed.

3.2 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 $\vec{x}$ is transmitted, the i-th received complex symbol $\vec{y}_i$, which corresponds to the i-th transmitted complex symbol $\vec{x}_i$, can be represented as,

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

where $\vec{n}_i$ includes the random noise, as well as the other possible interfering signals, and $\vec{h}_i$ is the channel coefficient between the transmitter and the receiver. The magnitude and angle of $\vec{h}_i$ 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 $\vec{s}$ composed of L complex symbols by performing cross-correlation [15] between the received signal and the known pattern. Given the received signal $\vec{y}$, its cross-correlation with the pattern $\vec{s}$ at position Δ is computed as,

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

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

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

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

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 $\vec{s}$.

What's more, a detection threshold is necessary in determining whether a pattern appears in the received signal. If the magnitude of cross-correlation $C(\vec{s}, \vec{y}, \Delta)$ exceeds the threshold, it implies that a pattern starts at position Δ in $\vec{y}$. 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 [15], the optimal threshold is given by,

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

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

4. 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.

4.1 Overview

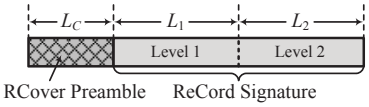


Figure 3: Structure of Messages for Neighbor Discovery in ReCorder

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, 7, 14, 21, 22]) to enhance robustness.

The newly designed message for neighbor discovery is a pseudo-random sequence. First, by exploiting the proposed RMix algorithm, the message is distinguished from other packets by a fixed pseudo-random symbol sequence, called RCover. Thus, a wireless node can detect the neighbor discovery messages through correlation as detecting a known symbol pattern. 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 acquire the following complex symbols corresponding to the ReCord in the message. Moreover, based on our RMix-2 algorithm, a wireless node can determine whether a newly received ReCord is from a new neighboring node or not by correlating it with the stored ones. Specifically, Figure 3 shows the corresponding format of a message for neighbor discovery.

4.2 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 Figure 3). 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 [8] 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 $\vec{s_C}$. A ReCord signature starts at the position where the correlation result spikes. There are two practical issues (frequency offset and detection threshold setting) existing in implementing this idea.

4.2.1 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,

$$\vec{y_i} = \vec{h_i}\vec{x_i}e^{j2\pi Ti\delta f} + \vec{n_i}, \qquad (5)$$

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 complensated. However, the frequency offset is generally small enough (e.g., less than 4 KHz [25]), so that it

does not obviously influence the RCover detection, provided that we keep the length of the preamble small.

4.2.2 Detection Threshold

As explained in Section 3, 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 [10], or assumes a large enough SINR [30]. 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,

$$SINR_c = \frac{|C(\vec{s}_C, \vec{y}, 0)| - \mathcal{C}(k-1)}{C(\vec{y}, \vec{y}, 0) - |C(\vec{s}_C, \vec{y}, 0)| + \mathcal{C}(k-1)}, \quad (6)$$

where $\vec{y}$ stores the most recently received L_C complex symbols 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,

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

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

What's more, 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,

$$SINR_e = \frac{C(\vec{y}, \vec{y}, 0) - E_s(k - L_C)}{E_s(k - L_C)}. \quad (8)$$

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

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

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,

$$T_C = \beta_1 \cdot \sqrt{L_C \cdot \frac{(|C(\vec{s}_C, \vec{y}, 0)| - \mathcal{C}(k-1))^2}{SINR_c}} + \beta_2 \cdot \mathcal{C}(k-1), \quad (10)$$

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.

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 $10 \lg SINR_e$ 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 $E_s(k)$,

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

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

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

where r is a constant approximate to 1, and is 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 $\vec{s_C}$. 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.

4.3 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 4.2, 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

Algorithm 1: RMix Algorithm for RCover Detection

Input: The received L_C symbols $\vec{y}$ at sampling point k, a copy of the RCover sequence $\vec{s}_C$.
Output: A flag indicating whether an RCover is detected at sampling point k.

$E_1 \leftarrow C(\vec{y}, \vec{y}, 0)$; $C_1 \leftarrow |C(\vec{s}_C, \vec{y}, 0)|$;
$E_s(k) \leftarrow (1-\eta_s) \cdot E_s(k-1) + \eta_s \cdot E_1$;
$E_f(k) \leftarrow (1-\eta_f) \cdot E_f(k-1) + \eta_f \cdot E_1$;
$\mathcal{C}(k) \leftarrow (1-\eta_s) \cdot \mathcal{C}(k-1) + \eta_s \cdot C_1$;
$SINR_c \leftarrow [C_1 - \mathcal{C}(k-1)] / [E_1 - C_1 + \mathcal{C}(k-1)]$;
$SINR_e \leftarrow [E_1 - E_s(k-L_C)] / E_s(k-L_C)$;
if $SINR_c > 0$ **and** $SINR_e > 0$ **and** $10 \lg \mathrm{SINR}_e > H_L$ **and** $r < E_f(k)/E_1 < r^{-1}$ **then**
  $T_C \leftarrow \beta_1 \cdot \sqrt{L_C (C_1 - \mathcal{C}(k-1))^2 / SINR_c} + \beta_2 \cdot \mathcal{C}(k-1)$;
  if $C_1 > T_C$ **then return True** ;
end
return False ;

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.

4.3.1 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 [8] 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 [8]. 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.

4.3.2 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 [10]. 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 $\vec{x}$ twice, and a receiver hears $\vec{y}$ and $\vec{y}'$ successively, then there is,

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

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.

Algorithm 2: RMix-2 Algorithm for ReCord Recognization

Input: The complex symbols following a newly detected RCover, including $\vec{y}_1$ of length L_1 and $\vec{y}_2$ of length L_2, and the local copy $\vec{s}_L$ 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 $Cr \leftarrow |C(\vec{s}_L, \vec{y}_1, i)|$; $C_{avg} \leftarrow C_{avg} + Cr$;
4 **if** $Cr > C_{\max}$ **then**
5 $C_{max} \leftarrow Cr$; $pos \leftarrow i$;
6 **end**
7 **end**
8 $C_{avg} \leftarrow (C_{avg} - C_{max})/(L_1 - 1)$;
9 $S_1 \leftarrow C_{max} - C_{avg}$; $I_1 \leftarrow C(\vec{y}_1, \vec{y}_1, 0) - S_1$;
10 **if** $C_{max} < \beta_1 \cdot \sqrt{L_1 \cdot S_1 \cdot I_1} + \beta_2 \cdot C_{avg}$ **then**
11 **return** $\mathcal{T}$;
12 **end**
13 **foreach** $< pos, \vec{s}_1, \vec{s}_2, sinr > \in \mathcal{T}$ **do**
14 $C_2 \leftarrow |C(\vec{s}_2, \vec{y}_2, 0)|$;
15 $C_{avg2} \leftarrow C_{avg} \cdot \sqrt{C(\vec{s}_2, \vec{s}_2, 0)/C(\vec{s}_L, \vec{s}_L, 0)}$;
16 $S_2 \leftarrow C_2 - C_{avg2}$; $I_2 \leftarrow C(\vec{y}_2, \vec{y}_2, 0) - S_2$;
17 **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 **continue** ;
19 **end**
20 $C' \leftarrow |C(\vec{s}_1|\vec{s}_2, \vec{y}_1|\vec{y}_2, 0)|$;
21 $C'_{avg} \leftarrow C_{avg} \cdot \sqrt{C(\vec{s}_1|\vec{s}_2, \vec{s}_1|\vec{s}_2, 0)/C(\vec{s}_L, \vec{s}_L, 0)}$;
22 $S' \leftarrow C' - C'_{avg}$; $I' \leftarrow C(\vec{y}_1|\vec{y}_2, \vec{y}_1|\vec{y}_2, 0) - S'$;
23 **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 $U \leftarrow U \cup \{< pos, \vec{s}_1, \vec{s}_2, sinr >\}$;
25 **end**
26 **end**
27 **if** $|U| = 0$ **then**
28 **return** $\mathcal{T} \cup \{< pos, \vec{y}_1, \vec{y}_2, S_1/I_1 >\}$;
29 **else if** $|U| = 1$ **and** $S_1/I_1 > SINR(U)$ **then**
30 **return** $\mathcal{T} \cup \{< pos, \vec{y}_1, \vec{y}_2, S_1/I_1 >\} \setminus U$;
31 **else**
32 **return** $\mathcal{T}$;
33 **end**

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 $\vec{y}_1$ and the Gold code sequence $\vec{s}_L$ 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 potential offset of $\vec{y}_1$. 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). 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 $O(L_1^2)$.

5. 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.

5.1 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 bandwidth. 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 bea-

coning mechanism with various neighbor disovery protocols using our testbed prototype, including Disco [7], U-Connect [14], Searchlight [4], Hello [26] and Diff-Code [22]. To be specific, we calculate the cumulative distribution of the latencies to discover four neighbors at one receiver over 200 runs. Again an additional node is set to provide interfering signals.

Finally, 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. We fix the OFDM link and adjust the transmission gain on the other link for neighbor discovery, so as to investigate the change of OFDM packet decoding rate under different SINR of ReCorder. 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.

5.2 Experiment Results

RCover Length	β_1
63	0.05
127	0.03
255	0.017

Table 1: Value of Parameter β_1

5.2.1 *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 1. 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.

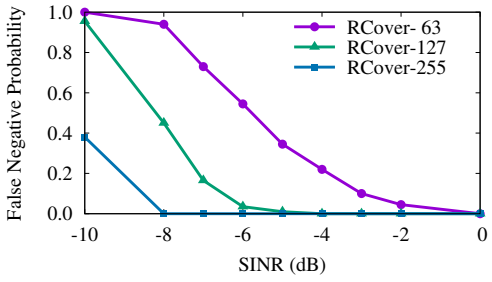


Figure 4: RCover Detection: False Negative Probability

In Figure 4, 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 Figure 4, 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.

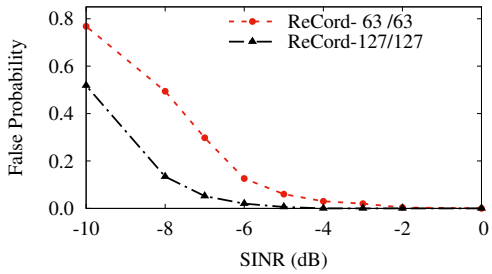
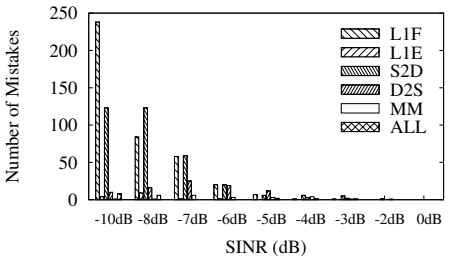


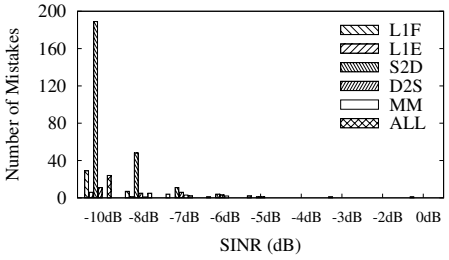
Figure 5: ReCord Recognization: False Recognization Probability

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 Figure 5. 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 owign to mismatching on the whole signature level. Among these six types, L1F, MM and ALL will increse the discovery latencies, and the type of L1E may lead to the discovery of un-existed neighbors. In addition, S2D results in du-



(a) Details: ReCord-63/63



(b) Details: ReCord-127/127

Figure 6: ReCord Recognization: False Recognizations

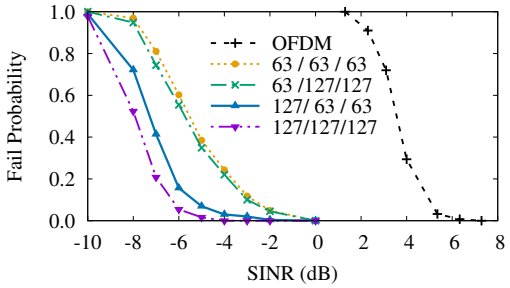


Figure 7: Comparison of False Probabilities: ReCorder *vs.* OFDM Beacons

plicated discovery, while D2S induces unnecessary discard of already discovered neighbors. We present the detailed results in Figure 6(a) and 6(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 cancelled by the interference.

What's more, 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. 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.

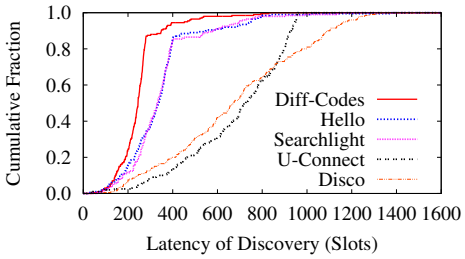
Till now we have obtained the false probabilities for both RCover detection and ReCord recognization. In Figure 7, we integrate the above results, and demonstrate the false probabilites of ReCorder. Note that we do not use the 255-symbol RCover preamble because of its high transmission overhead. By comparison, 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. 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.

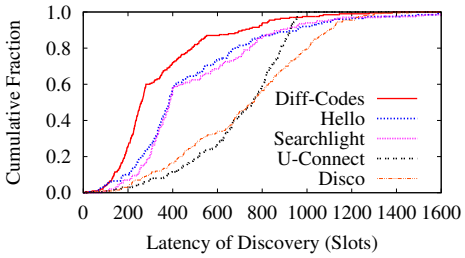
5.2.2 *Cases of Applications*

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 for ReCorder, and 4dB for OFDM. The cumulative distributions of discovery latencies are shown in Figure 8. We can see that with the same neighbor discovery protocol, ReCorder at -5dB can achieve the median and worst-case gains that are both larger than 10% over OFDM at 4dB. For example, the median and worst-case gains are 14.0% and 30% with Searchlight. That is because even at the SINR of -5dB, the false probability of ReCorder is lower than the PER of OFDM beacons at 4dB. In practice, the smaller discover latency leads to the reduction of energy consumption for neighbor discovery.

Finally, we present how the decoding of OFDM packets is impacted by neighbor discovery messages of ReCorder in Figure 9. We carry out the experiment with three bit rates in IEEE 802.11 a/g. When ReCorder and OFDM have the same bandwidth of 20 MHz, at -6dB SINR for ReCorder, the bit rates of 6Mbps, 9Mbps and 12Mbps experience no decoding degradation. When the SINR of ReCorder is increased, the performance



(a) ReCorder (127/63/63) under SINR −5dB



(b) OFDM Beacons under SINR 4dB

Figure 8: CDF of Discovery Latencies for Symmetric Duty Cycle 5%

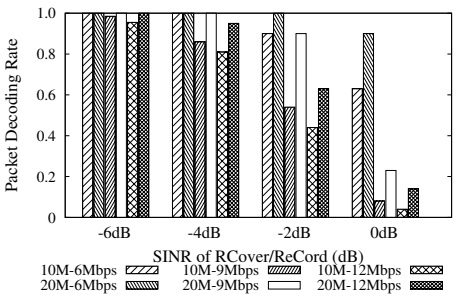


Figure 9: Impact of ReCorder on OFDM Data Packets

of OFDM with lower bit rates are relatively stable. As for higher bit 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 bit rates (*i.e.*, 6Mbps and 9Mbps), while still achieving robust performance. As explained in Section 6, that is important for the coexistence 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.

6. DISCUSSION

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

6.1 Signature Collision

As mentioned in Section 4.3, the level-1 ReCord adopts Gold code of length L_1. Therefore, the number of distinct level-1 signatures also equals L_1 [8, 20]. 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,

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

According to the above equation, provided the level-1 ReCord with $L_1 = 127$ and a network with $N = 50$ nodes, we have $P(50, 1, 127) = 99.9987\% \approx 1$ and $P(50, 2, 127) = 7.321\%$. 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.

6.2 Energy Efficiency

Due to the increased robustness, ReCorder suffers from less neighbor discovery message losses. That can reduce the discovery latency in practice, which brings higher energy efficiency compared with the beaconing and decoding way.

In addition, ReCorder also 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-bit symbol sequence as neighbor discovery message when $m = 1$, and 362-bit when $m = 2$. The shorter neighbor discovery messages consumes at least 2/3 less transmission energy on the sending side. What's more, in IEEE 802.11a/g (OFDM), a node needs the process of $\text{FFT}^{(-1)}$/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.

6.3 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 physi-

cal layer standard, such as IEEE 802.15.4 and 802.11b. To be specific, the authors in [28] 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 bitrate WiFi control and management frames, so that it does not harm the regular operations of a WiFi network, 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 works on TCP has proposed the congestion control architecture [6], which is able to resist random packet loss, and maintain the end-to-end throughput without additional hardware support, the co-existence between ReCorder and background transmissions is promising in practice. Furthermore, we can implement quite a few existing works such as rateless code (*e.g.*, [11, 24]) and partial packet recovery (*e.g.*, [12]), 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 [9] 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.

7. 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 3/4. 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. 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 [19], and to improve the evidence of a neighboring node's identity.

8. ACKNOWLEDGMENTS

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2012CB316201), in part by China NSF grant 61422208, 61472252, 61272443 and 61133006, in part by CCF-Intel Young Faculty Researcher Program and CCF-Tencent Open Fund, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, and in part by Jiangsu Future Network Research Project No. BY2013095-1-10. 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.

9. REFERENCES

[1] Sony PS Vita-Near. http://us.playstation.com/psvita/.

[2] Wi-Fi Direct. http://www.wi-fi.org/discover-and-learn/wi-fi-direct/.

[3] IEEE Std 802.11-2012 - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE*, 2012.

[4] Mehedi Bakht, Matt Trower, and Robin Hillary Kravets. Searchlight: won't you be my neighbor? In *Proceedings of ACM MobiCom*, 2012.

[5] Tarun Bansal, Bo Chen, Prasun Sinha, and Kannan Srinivasan. Symphony: cooperative packet recovery over the wired backbone in enterprise wlans. In *Proceedings of ACM MobiCom*, 2013.

[6] Mo Dong, Qingxi Li, Doron Zarchy, Brighten Godfrey, and Michael Schapira. Pcc: Re-architecting congestion control for consistent high performance. In *Proceedings of USENIX NSDI*, 2014.

[7] Prabal Dutta and David E. Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *Proceedings of ACM SenSys*, 2008.

[8] Pingzhi Fan and Michael Darnell. *Sequence design for communications applications*, volume 30. Research Studies Press Taunton, 1996.

[9] Geoffrey M. Voelker Feng Lu, Patrick Ling and Alex C. Snoeren. Enfold: Downclocking ofdm in wifi. In *Proceedings of ACM MobiCom*, 2014.

[10] Shyamnath Gollakota and Dina Katabi. Zigzag decoding: combating hidden terminals in wireless networks. In *Proceedings of ACM SIGCOMM*, 2008.

[11] Aditya Gudipati and Sachin Katti. Strider: automatic rate adaptation and collision handling. In *Proceedings of ACM SIGCOMM*, 2011.

[12] Kyle Jamieson and Hari Balakrishnan. PPR: Partial packet recovery for wireless networks. In *Proceedings of ACM SIGCOMM*, 2007.

[13] Jehn-Ruey Jiang, Yu-Chee Tseng, Chih-Shun Hsu, and Ten-Hwang Lai. Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks. *Mobile Networks and Applications*, 10(1-2):169–181, 2005.

[14] Arvind Kandhalu, Karthik Lakshmanan, and Ragunathan Rajkumar. U-Connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proceedings of ACM IPSN*, 2010.

[15] Steven M Kay. *Fundamentals of statistical signal processing: detection theory*. Prentice-hall, 1998.

[16] Ramin Khalili, Dennis L Goeckel, Don Towsley, and Ananthram Swami. Neighbor discovery with reception status feedback to transmitters. In *Proceedings of IEEE INFOCOM*, 2010.

[17] Shouwen Lai, Binoy Ravindran, and Hyeonjoong Cho. Heterogenous quorum-based wake-up scheduling in wireless sensor networks. *IEEE Transactions on Computers*, 59(11):1562–1575, 2010.

[18] Christopher Lott, Olgica Milenkovic, and Emina Soljanin. Hybrid arq: theory, state of the art and future directions. In *IEEE Information Theory Workshop on Information Theory for Wireless Networks*, 2007.

[19] Tarcisio F Maciel, Rodrigo L Batista, Lunider Elias, Alexandre Robson, and Francisco RP Cavalcanti. Network-assisted neighbor discovery based on power vectors for d2d communications. In *Proceedings of VTC Spring*, 2015.

[20] Eugenio Magistretti, Omer Gurewitz, and Edward W Knightly. 802.11 ec: collision avoidance without control messages. In *Proceedings of ACM MobiCom*, 2012.

[21] Michael J. McGlynn and Steven A. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *Proceedings of ACM MobiHoc*, 2001.

[22] Tong Meng, Fan Wu, and Guihai Chen. On designing neighbor discovery protocols: a code-based approach. In *Proceedings of IEEE INFOCOM*, 2014.

[23] Tong Meng, Fan Wu, and Guihai Chen. Code-based neighbor discovery protocols in mobile wireless networks. *IEEE/ACM Transactions on Networking*, 2015. doi: 10.1109/TNET.2015.2388534.

[24] Jonathan Perry, Peter Iannucci, Kermin Fleming, Hari Balakrishnan, and Devavrat Shah. Spinal codes. In *Proceedings of ACM SIGCOMM*, 2012.

[25] Souvik Sen, Romit Roy Choudhury, and Srihari Nelakuditi. CSMA/CN: carrier sense multiple access with collision notification. *IEEE/ACM Transactions on Networking*, 20(2):544–556, 2012.

[26] Wei Sun, Zheng Yang, Keyu Wang, and Yunhao Liu. Hello: A generic flexible protocol for neighbor discovery. In *Proceedings of IEEE INFOCOM*, 2014.

[27] Sudarshan Vasudevan, Donald Towsley, Dennis Goeckel, and Ramin Khalili. Neighbor discovery in wireless networks and the coupon collector's problem. In *Proceedings of ACM MobiCom*, 2009.

[28] Kaishun Wu, Haoyu Tan, Yunhuai Liu, Jin Zhang, Qian Zhang, and Lionel M. Ni. Side channel: bits over interference. In *Proceedings of ACM MobiCom*, 2010.

[29] Wei Zeng, Sudarshan Vasudevan, Xian Chen, Bing Wang, Alexander Russell, and Wei Wei. Neighbor discovery in wireless networks with multipacket reception. In *Proceedings of ACM MobiHoc*, 2011.

[30] Xinyu Zhang and Kang G Shin. E-mili: energy-minimizing idle listening in wireless networks. *IEEE Transactions on Mobile Computing*, 11(9):1441–1454, 2012.

[31] Rong Zheng, Jennifer C. Hou, and Lui Sha. Asynchronous wakeup for ad hoc networks. In *Proceedings of ACM MobiHoc*, 2003.