ASYNCHRONOUS NEIGHBOR DISCOVERY USING COUPLED COMPRESSIVE SENSING

†*Vamsi K. Amalladinne,* †*Krishna R. Narayanan,*
†*Jean-Francois Chamberland, §Dongning Guo*

†Department of Electrical and Computer Engineering, Texas A&M University
§Department of Electrical Engineering and Computer Science, Northwestern University

ABSTRACT

The neighbor discovery paradigm finds wide application in Internet of Things networks, where the number of active devices is orders of magnitude smaller than the total device population. Designing low-complexity schemes for asynchronous neighbor discovery has recently gained significant attention from the research community. Concurrently, a divide-and-conquer framework, referred to as coupled compressive sensing, has been introduced for the synchronous massive random access channel. This work adapts this novel algorithm to the problem of asynchronous neighbor discovery with unknown transmission delays. Simulation results suggest that the proposed scheme requires much fewer transmissions to achieve a performance level akin to that of state-of-the-art techniques.

Index Terms— Neighbor discovery, random access, compressive sensing, forward error correction, asynchronous schemes.

1. INTRODUCTION

Neighbor discovery is a frequently occurring task in Internet of Things (IoT) networks [1]. In this context, a node within a wireless network seeks to identify all neighbors with which it can communicate directly. The total number of potential devices in a typical IoT network is on the order of billions. Hence, there is a pressing need to design algorithms whose computational complexity scales moderately with the number of devices and bandwidth [1–7]. Further, because devices are at different geographical locations, their transmissions are typically asynchronous as perceived by the receiver. It is thus important to design coding schemes that perform well in the presence of asynchronous transmissions.

The interest in finding good solution to neighbor discovery is evinced by the large number of contributions on this topic. In particular, tools from compressive sensing have been employed to design algorithms for neighbor discovery in [1, 5–7]. Two different paradigms are proposed for synchronous neighbor discovery in [5]. The first algorithm employs random on-off signatures and the query node discovers its neighbors using a variation of group testing. The other algorithm is based on second order Reed-Muller codes followed by chirp reconstruction. In [6], the authors employ sparse Kerdock matrices as codebooks for synchronous neighbor discovery. Still, the schemes developed in [5, 6] rely on symbol synchronicity across transmissions, and they cannot be used when signals are asynchronous. In [7], LASSO is employed for node identification in an asynchronous random access scenario. Regrettably, the schemes mentioned above do not scale well when the device address space becomes huge (e.g., 48-bit IEEE 802.11 MAC addresses [8]). In [1], an asynchronous neighbor discovery scheme based on sparse orthogonal frequency division multiplexing (S-OFDM) is proposed. This approach is shown to scale well with the device address space. That is, the computational complexity of this scheme scales linearly with the number of active devices, and logarithmically with the cardinality of the address space.

In this paper, we develop a scalable, asynchronous neighbor discovery algorithm tailored to large address spaces. This novel approach requires substantially fewer transmissions (smaller bandwidth) than previously published alternatives [1] to achieve a comparable error performance. The proposed scheme builds on our earlier work on coupled compressive sensing (CCS), which first appeared in the treatment of the unsourced and uncoordinated multiple access channel [9]. Herein, we demonstrate that a variant of the CCS algorithm outperforms state-of-the-art schemes for asynchronous neighbor discovery as well. There are a few key features that distinguish the massive random access paradigm considered in [9] and the neighbor discovery problem at hand. Only additive white Gaussian noise (AWGN) appears in [9], whereas the present article considers a flat fading environment where channel coefficients are unknown at the receiver. Also, transmissions are synchronous in [9]; the modified CCS algorithm must thence be extended to asynchronous cases.

Having reviewed existing contributions, we turn to the presentation of our scheme. Throughout, we employ the following notation. $\mathbf{DFT}_k$ symbolizes the $k \times k$ discrete-Fourier transform matrix. The cardinality of set $\mathcal{S}$ is represented by $|\mathcal{S}|$. We employ $\mathbb{C}$, $\mathbb{Z}$, and $\mathbb{Z}_+$ to denote complex numbers, integers, and non-negative integers, respectively. For any $a, b \in \mathbb{Z}$, $[a : b] = \{c \in \mathbb{Z} : a \leq c \leq b\}$. For a vector $\mathbf{x} = [x_1 \ x_2 \cdots x_n] \in \mathbb{C}^n$, $\mathrm{supp}(\mathbf{x}) = \{k \in [1 : n] : x_k \neq 0\}$, $\|\mathbf{x}\|_0 = |\mathrm{supp}(\mathbf{x})|$, and $\|\mathbf{x}\|_p$ is the standard ℓ_p norm with $p \geq 1$. We write $X \sim \mathbb{U}[a, b]$ to indicate that X is uniformly distributed between a and b. Symbols $\mathbf{0}_n$, $\mathbf{0}_{k \times n}$ denote the all-zero column vector of length n and the $k \times n$ all-zero matrix, respectively. We use $\delta_{i,j}$ to denote the standard Kronecker delta function.

2. SYSTEM MODEL

Let $\mathcal{K}_{\mathrm{tot}}$ be the collection of all nodes and $\mathcal{K} \subseteq \mathcal{K}_{\mathrm{tot}}$ be the subset of active nodes within a network. For convenience, we assign parameters to the cardinalities of these sets, namely $|\mathcal{K}_{\mathrm{tot}}| = K_{\mathrm{tot}}$ and $|\mathcal{K}| = K$. An active device $k \in \mathcal{K}$ transmits a frame (codeword) $\mathbf{x}_k = [x_{k,0} \ x_{k,1} \cdots x_{k,N-1}]^{\mathrm{T}} \in \mathbb{C}^N$ of N complex symbols with $\mathbb{E}[\|\mathbf{x}_k\|_2^2] = N$. The signal received by the query node at time instant i is given by

$$y_i = \sum_{k \in \mathcal{K}} \sqrt{P} h_k x_{k,i-\tau_k} + w_i, \qquad (1)$$

where h_k denotes the complex fading coefficient of the link between node k and the query node, τ_k is the discrete delay in terms of symbols experienced by the signal sent by device k, P represents the transmission power, and w_i is complex additive white Gaussian noise with zero mean and unit variance. We explore two fading models.

Fading Model I: To facilitate a fair comparison between the proposed scheme and the approach of [1], we assume that the magnitude of each fading coefficient is bounded above and below by two constants, $\underline{h}$ and $2\underline{h}$, and its phase is distributed uniform between 0 and 2π. Specifically, $|h_k| \sim \mathbb{U}[\underline{h}, 2\underline{h}]$ and $\angle h_k \sim \mathbb{U}[0, 2\pi]$ for all $k \in \mathcal{K}$. The signal-to-noise ratio (SNR) for this model is equal to $\mathsf{SNR} = P|\underline{h}|^2/2$.

Fading Model II: In this model, we incorporate the effects of network topology and signal propagation loss into fading. Similar to [5], we assume that the channel gains follow a Pareto distribution with scale parameter $\eta > 0$ and shape parameter $\alpha > 1$[1]. The transmission SNR for this model is given by $\mathsf{SNR} = P\eta/2$.

The delays are specific to each communication link, and remain constant over the duration of a frame. Further, they are bounded above by a known constant $\mathcal{T}$, i.e., $\mathcal{T} = \max_{k \in \mathcal{K}} \tau_k$, since we are interested only in discovering neighbors who fall within a bounded distance. We refer

to the special case $\mathcal{T} = 0$ as synchronous neighbor discovery. The receiver assumes that signals are only present during the frame duration, i.e., $x_{k,i} = 0$ when $i < 0$ or $i > N - 1$ for $k \in \mathcal{K}$. The receiver is tasked with finding $\hat{\mathcal{K}}$, an estimate of the unknown collection $\mathcal{K}$ of active nodes, given measurements $\mathbf{y} = [y_0 \ y_1 \ \cdots \ y_{N-1}]^{\mathrm{T}}$. To facilitate neighbor discovery, every active device needs to embed its identity in the transmission. Since the system features K_{tot} devices, nodes can be distinguished using roughly $\log_2 K_{\mathrm{tot}}$ bits, which are encoded into N complex symbols and sent. Thus, we have a multiple access system, with K active devices intending to transmit $\log_2 K_{\mathrm{tot}}$ bits each by encoding them into N complex symbols.

The error probability of the system is defined as the probability that the estimated set $\hat{\mathcal{K}}$ is not the same as the actual set $\mathcal{K}$ of active devices. Mathematically, $P_{\mathrm{e}} = \Pr[\hat{\mathcal{K}} \neq \mathcal{K}]$. Our objective is to design a low-complexity coding scheme that offers good error performance with respect to SNR at low codeword lengths.

3. PROPOSED SCHEME

Our new scheme is based on adapting the CCS algorithm [9] to the needs of neighbor discovery. We initiate the discussion with the encoding scheme. As mentioned above, the amount of information needed to identify a device is $B = \log_2 K_{\mathrm{tot}}$, and we refer to this data as the message bits. Every B-bit message is encoded into N complex symbols, and active devices concurrently transmit the resulting signals. The encoding process features an outer encoder, which is referred to as the tree encoder, and an inner CS encoder; these two parts are explained below.

Tree Encoder: The tree encoder splits the B-bit identity into several sub-blocks, each of which is amenable to CS recovery. To this end, the B-bit message is partitioned into n sub-blocks and redundancy is added to some blocks. Let m_i denote the number of message bits present in the ith sub-block, $\sum_{i=0}^{n-1} m_i = B$. Let l_i denote the number of redundant parity-check bits added to sub-block i, with $l_0 = 0$; We never add parity bits to the 0th sub-block. The length of each coded sub-block is J, i.e., $m_i + l_i = J$ for $i \in [0 : n-1]$. We denote the length of coded block by M, which yields $B + \sum_{i=0}^{n-1} l_i = M$. The parity-check bits added in the ith sub-block act as random parity check constraints for all the message bits preceding it[2].

CS Encoder: Every coded sub-block of J bits is further encoded into $\tilde{N} = \frac{N}{n}$ complex symbols by the CS encoder. That is, every M-bit output of the tree encoder is mapped into N complex symbols by the CS encoder. Specifically, each sub-block corresponding to a device is first encoded into $\tilde{N} - \mathcal{T}$ complex symbols via a matrix $\mathbf{S} = [\mathbf{s}_1 \ \mathbf{s}_2 \cdots \mathbf{s}_{2^J}] \in \mathbb{C}^{(\tilde{N}-\mathcal{T}) \times 2^J}$. This is accomplished

---

[1]See [5] for details regarding how this model is relevant to the neighbor discovery problem.

[2]See [9] for more details regarding the construction of parity-check bits, their allocation across sub-blocks and the theoretical analysis of error probability and computational complexity of the CCS scheme.

through a bijective function $f : \{0,1\}^J \mapsto \{\mathbf{s}_j, j \in [1 : 2^J]\}$ that maps sub-blocks into columns of $\mathbf{S}$. When transmissions are synchronous ($\mathcal{T} = 0$), the receiver observes superimposed sub-blocks and it can recover individual sub-blocks by implementing a CS decoding algorithm. On the other hand, when transmissions are asynchronous, there may be interference from other sub-blocks. To prevent this from occurring, each column in $\mathbf{S}$ is padded with $\mathcal{T}$ zeros to form the codebook $\mathbf{S}_{\mathrm{ZP}} = [\mathbf{S}^{\mathrm{T}} \mathbf{0}_{\mathcal{T} \times 2^J}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{\tilde{N} \times 2^J}$. In other words, each sub-block of length J bits is encoded into a column of $\mathbf{S}_{\mathrm{ZP}}$ and transmitted over the channel.

At this stage, we turn to the decoding process, which also features two components: an inner CS decoder, which recovers sub-blocks from asynchronous noisy measurements; and an outer tree decoder, which connects all the sub-blocks corresponding to a parent message.

CS decoder: A compressive sensing framework for asynchronous multiple access was first developed in [7]. Let $\mathcal{I}^{(i)} = \{i_k : k \in K\} \subset [1 : 2^J]$ denote the indices corresponding to the columns of $\mathbf{S}_{\mathrm{ZP}}$ transmitted during i^{th} sub-block. Since the maximum delay $\mathcal{T}$ is known to the decoder, the signal received during the ith sub-block can be written as

$$\mathbf{y}^{(i)} = \sum_{k \in \mathcal{K}} h_k \tilde{\mathbf{s}}_{i_k}^{(\tau_k)} + \mathbf{w}^{(i)}, \qquad (2)$$

where $\tilde{\mathbf{s}}_{i_k}^{(\tau_k)} = [\mathbf{0}_{\tau_k}^{\mathrm{T}} \ \mathbf{s}_{i_k}^{\mathrm{T}} \ \mathbf{0}_{\mathcal{T}-\tau_k}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{\tilde{N}}$, a shifted version of $\mathbf{s}_{i_k}^{\mathrm{T}}$ with zero paddings. Equation (2) can be expressed in matrix form as

$$\mathbf{y}^{(i)} = \tilde{\mathbf{S}} \tilde{\mathbf{h}}^{(i)} + \mathbf{w}^{(i)}, \qquad (3)$$

where $\tilde{\mathbf{S}} = [\tilde{\mathbf{S}}_1 \ \tilde{\mathbf{S}}_2 \cdots \tilde{\mathbf{S}}_{2^J}]$ is an $\tilde{N} \times 2^J(\mathcal{T}+1)$ matrix, and sub-matrices $\tilde{\mathbf{S}}_j = [\tilde{\mathbf{s}}_j^{(0)} \ \tilde{\mathbf{s}}_j^{(1)} \cdots \tilde{\mathbf{s}}_j^{(\mathcal{T})}] \in \mathbb{C}^{\tilde{N} \times (\mathcal{T}+1)}$ for $j \in [1 : 2^J]$. The vectors $\tilde{\mathbf{h}}^{(i)} = [\tilde{\mathbf{h}}_1^{(i)} \ \tilde{\mathbf{h}}_2^{(i)} \cdots \tilde{\mathbf{h}}_{2^J}^{(i)}]^T \in \mathbb{C}^{2^J(\mathcal{T}+1)}$ are defined by

$$\tilde{\mathbf{h}}_j^{(i)} = \sum_{k \in \mathcal{K}} [\mathbf{0}_{\tau_k}^{\mathrm{T}} \ h_k \ \mathbf{0}_{\mathcal{T}-\tau_k}^{\mathrm{T}}]^{\mathrm{T}} \delta_{j,i_k}. \qquad (4)$$

Equation (3) resembles a standard compressed sensing setting with $\mathbf{y}^{(i)}$ acting as the observation vector and $\tilde{\mathbf{S}}$ being the sensing matrix. The CS decoder constructs the sensing matrix $\tilde{\mathbf{S}}$ using the procedure outlined above. An estimate $\hat{\tilde{\mathbf{h}}}^{(i)}$ of vector $\tilde{\mathbf{h}}^{(i)}$ is first computed using measurements $\mathbf{y}^{(i)}$ by implementing any standard CS decoding technique. The final estimate of $\tilde{\mathbf{h}}^{(i)}$ is subsequently obtained by taking the best K-term approximation[3] to the vector $\hat{\tilde{\mathbf{h}}}^{(i)}$.

Tree Decoder: The purpose of the tree decoder is to string together the message components identified by the CS decoder. Our choice algorithm is adapted from [9]. Yet, in contrast to our previous work, the concatenation process is informed by both the parity-check bits and the fading coefficients returned by the CS decoder. Integrating the soft

---

[3]The best s-term approximation to a vector $\mathbf{x} \in \mathbb{C}^n$ is defined as $\hat{\mathbf{x}}_s = \mathrm{argmin}_{\mathbf{z} \in \mathbb{C}^n, ||\mathbf{z}||_0 \le s} ||\mathbf{x} - \mathbf{z}||_2$.

estimates is a means to reduce the amount of redundancy needed in the tree code. Pragmatically, the ability to discriminate between fade levels is key in successfully leveraging soft estimates. When the fading profile is severe, the coefficients become very useful in terms of piecing together message components. However, when fade variations are minimal, the system must revert back to adding parity bits. The tree decoding algorithm is explained below.

Every candidate sub-block listed by the CS decoder at the onset of the process becomes a root of a decoding tree. Given a root sub-block, layer 1 of the tree is composed of all K sub-blocks found on the second list of the CS decoder. This process continues, adding K branches to every node, until the final CS decoding stage is reached. This results in a tree with K^{n-1} leafs. The tree decoder then proceeds iteratively to aggregate the sub-blocks that correspond to one parent message among these K^{n-1} possible paths. The vectors $\{\tilde{\mathbf{h}}^{(i)}\}_{i=0}^{n-1}$ in (3) have different support sets, yet they share the same non-zero entries because the fading coefficient corresponding to a device does not change across sub-blocks. Thus, the soft fade estimates offered by the CS decoder can act as statistical features to piece together the sub-blocks associated with one parent message. To this end, at stage i of the tree decoding process, we only retain a subset $\mathcal{L}_i \subseteq [1 : K^i]$ of nodes whose estimates of the fading coefficients are comparable to the estimate of the fade level based on the paths retained up to stage $i - 1$. We further prune the decoding tree by only keeping partial paths that satisfy the l_i constraints associated with parity check bits. At stage $i + 1$, the tree decoder focuses exclusively on the children of nodes that survived stage i. If there exists a single valid path at the end, the decoder outputs the corresponding message; otherwise, it declares a failure. Fig. 1 illustrates various stages involved in our tree decoding algorithm.
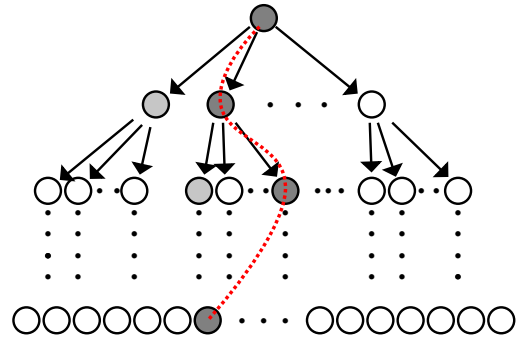

Fig. 1: This graph illustrates the tree decoding process. Fading coefficients are captured by the shade of nodes. At every stage, only nodes that are close to the fade estimate are retained. Parity check constraints are then verified for residual partial paths to further prune the tree. The highlighted path represent the unique survivor of this process, and thereby forms the basis for the decoded message.

4. SIMULATION RESULTS

In this section, we present simulation results that showcase the performance of the proposed scheme and compare it to the performance of the scheme found in [1]. A random sub-matrix of $\mathbf{DFT}_{2^J}$ of dimension $\tilde{N}-\mathcal{T}\times 2^J$ is chosen as the device codebook $\mathbf{S}$ for all simulations. These matrices are known to satisfy the restricted isometry property (RIP) and to be a good choice for noisy compressed sensing [10]. The LASSO algorithm is utilized to solve the CS sub-problems.

Fading Model I: We consider a network with $K_{\text{tot}} = 2^{38}$ devices and an active device population with $K \in \{10, 100\}$. The number of sub-blocks, the length of each sub-block, and the number of parity check bits in each sub-block are chosen as $n = 6$, $J = 10$, $[l_0 \ l_1 \cdots l_5] = [0 \ 0 \ 0 \ 2 \ 10 \ 10]$ for $K = 10$; and $n = 10$, $J = 11$, $[l_0 \ l_1 \cdots l_9] = [0 \ 5 \ 7 \ 7 \ 7 \ 7 \ 7 \ 10 \ 11 \ 11]$ for $K = 100$. For this first fading model, the channel gains are distributed within a fairly small range. As such, the reconstruction error of the soft channel estimates will largely be uninformative. Thus, in this case, tree decoding only relies on parity-check bits to stitch the sub-blocks of a message together. In Fig. 2, the error probability of synchronous neighbor discovery is plotted as a function of SNR. For the same number of channel uses, the proposed scheme provides about 10 dB improvement over the results in [1]. Similarly, for a same error performance, the required codeword length is reduced by approximately 75%.
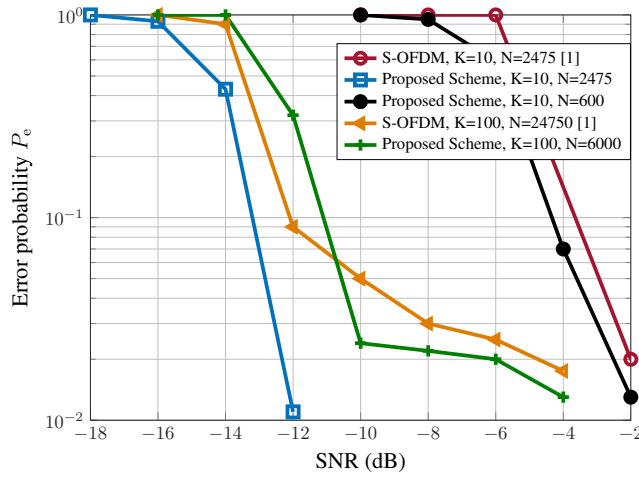

Fig. 2: This graph shows the error probability of synchronous neighbor discovery as a function of SNR, and hints at the superior performance of the proposed scheme.

Figure 3 illustrates the performance of asynchronous neighbor discovery. The maximum transmission delay is set to $\mathcal{T} = 20$ symbols, and the delay τ_k of active device k is drawn uniformly at random from $[0 : \mathcal{T}]$. Again, we see a substantial performance improvement for the proposed scheme, compared to [1]. A given error probability can be
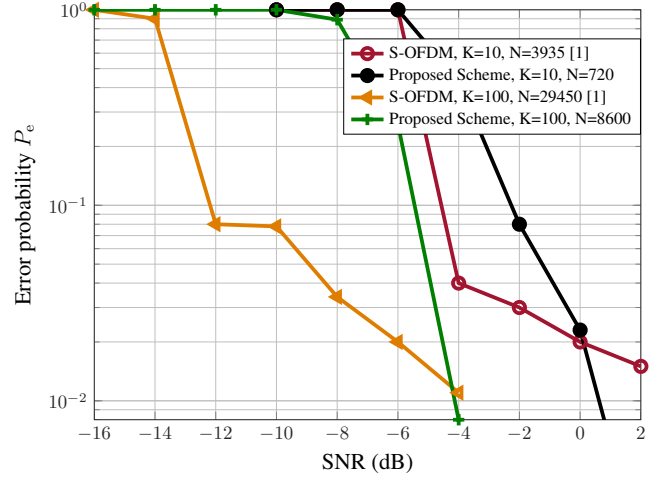

Fig. 3: The error probability of asynchronous neighbor discovery is compared in this graph for fading model I.

achieved with a reduction in codeword length of approximately 82% for $K = 10$, and 70% for $K = 100$.

Fading Model II: We consider a network with $K_{\text{tot}} = 2^{38}$, $K = 20$ and $\mathcal{T} = 0$. The number of sub-blocks $n = 4$, each sub-block is of length 12 and the parity-check bits are allocated as $[l_0 \ l_1 \ l_2 \ l_3] = [0 \ 0 \ 0 \ 10]$. The scale parameter of Pareto distribution is set to $\eta = 0.05$. Since the Pareto distribution has a heavy tail, the channel gains are distributed over a larger range and their estimates serve as an outer code in this model. Figure 4 shows the probability of missed detection for various values of the shape parameter α. It can be seen that increasing α worsens the performance, since the channel gains tend to concentrate within a small range. Note that these are not comparison curves in this case because this fading scenario was not treated in [1].
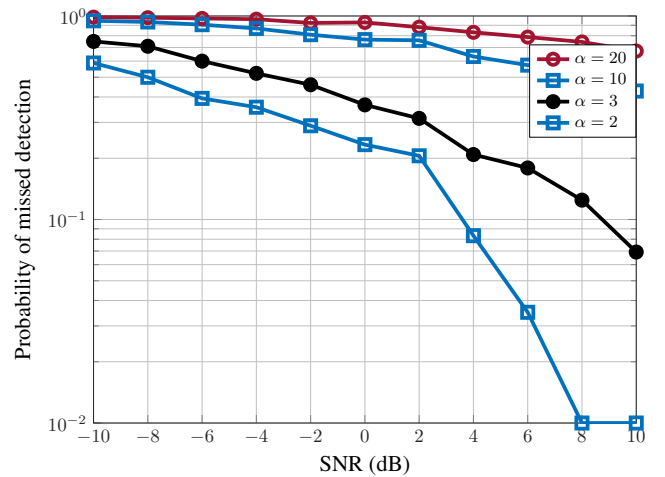

Fig. 4: This plot showcases the feasibility of leveraging fading coefficients to stitch sub-blocks together.

References

[1] Xu Chen, Dongning Guo, and Gregory W Wornell, "Sparse OFDM: A compressive sensing approach to asynchronous neighbor discovery," *arXiv preprint arXiv:1706.09387*, 2017.

[2] Sudarshan Vasudevan, Donald Towsley, Dennis Goeckel, and Ramin Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 181–192.

[3] Steven A Borbash, Anthony Ephremides, and Michael J McGlynn, "An asynchronous neighbor discovery algorithm for wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 7, pp. 998–1016, 2007.

[4] Daniele Angelosante, Ezio Biglieri, and Marco Lops, "Neighbor discovery in wireless networks: A multiuser-detection approach," *Physical Communication*, vol. 3, no. 1, pp. 28–36, 2010.

[5] Lei Zhang, Jun Luo, and Dongning Guo, "Neighbor discovery for wireless networks via compressed sensing," *Performance Evaluation*, vol. 70, no. 7-8, pp. 457–471, 2013.

[6] Andrew Thompson and Robert Calderbank, "Compressed neighbour discovery using sparse kerdock matrices," *arXiv preprint arXiv:1801.04537*, 2018.

[7] Lorne Applebaum, Waheed U Bajwa, Marco F Duarte, and Robert Calderbank, "Asynchronous code-division random access using convex optimization," *Physical Communication*, vol. 5, no. 2, pp. 129–147, 2012.

[8] IEEE Computer Society LAN MAN Standards Committee et al., "Wireless lan medium access control (mac) and physical layer (phy) specifications," *ANSI/IEEE Std. 802.11-1999*, 1999.

[9] Vamsi K Amalladinne, Avinash Vem, Dileep Kumar Soma, Krishna R Narayanan, and Jean-Francois Chamberland, "A coupled compressive sensing scheme for uncoordinated multiple access," *arXiv preprint arXiv:1809.04745*, 2018.

[10] Ishay Haviv and Oded Regev, "The restricted isometry property of subsampled Fourier matrices," in *Geometric Aspects of Functional Analysis*, pp. 163–179. Springer, 2017.