Neighbor Discovery in Wireless Ad Hoc Networks Based on Group Testing

Jun Luo and Dongning Guo

Department of Electrical Engineering & Computer Science
Northwestern University
Evanston, IL 60208, USA

***Abstract*—Fast and efficient discovery of all neighboring nodes by a node new to a neighborhood is critical to the deployment of wireless ad hoc networks. Different than the conventional ALOHA-type random access discovery schemes, this paper assumes that all nodes in the neighborhood simultaneously send their unique on-off signatures known to the receive node. In particular, a transmitter does not transmit any energy during an "off" mini-slot in its signature sequence. The received signal can be viewed as the outcome of a sequence of tests over the mini-slots, where the outcome of a test is positive if there is energy at the corresponding mini-slot from at least one neighbor, and negative if none of the neighboring nodes transmits energy during the mini-slot. The neighbor discovery problem is thus equivalent to a classical group testing problem. Two practical and scalable detection algorithms are developed from the group testing viewpoint. Unlike some previous neighbor discovery schemes using coherent multiuser detection, which are difficult to implement due to lack of training, the proposed scheme requires only non-coherent energy detection. The proposed algorithms are shown to achieve faster and more reliable discovery than existing random access schemes.**

I. Introduction

The emerging wireless ad hoc network paradigm enables a new type of network in which collaborating devices relay packets from one device to another across multiple wireless links in a self-organizing manner. A number of applications based on this type of network have been established or are expected in the near future, such as environmental and building monitoring, disaster relief and military battlefield communication. Due to the self-organizing nature of ad hoc networks, every node in the network can be alternately functioning as a transmitter or a receiver. Oftentimes, a node can communicate directly with only several other nodes around itself, which are called its "neighbors". In absence of a central controller, every node has to discover its neighbors before efficient routing is possible. The process for a node to identify all its neighbors is called *neighbor discovery*, which is a crucial first step of constructing reliable wireless ad hoc networks.

Neighbor discovery in ad hoc networks is a critical and non-trivial task. Algorithms such as "birthday protocol" [1], directional antenna neighbor discovery [2], [3] and slotted random transmission and reception [4] have been proposed to enable all nodes in a network to find out their neighbors either synchronously or asynchronously. These algorithms can be categorized as random access discovery, which requires nodes to be randomly in a "transmitting" or "listening" state in each time slot so that each node gets a chance to hear every neighbor for at least once in a sufficient amount of time. Such random access discovery schemes allow one transmission to be successful at a time, and hence generally require a large number of time slots until reliable neighbor discovery is achieved.

Timely discovery of a node's neighbors is a critical issue in wireless networks, especially when the nodes are mobile. References [5]–[7] suggest solution of the neighbor discovery problem from the multiuser detection perspective. The idea is to let all neighbors simultaneously send their unique signature waveforms which identify themselves, and let the center node detect which signatures are at presence. The advantage is rapid detection achieved using multiuser detectors, which are well-understood in the context of code-division multiple access (CDMA). However, the difficulties of scaling the scheme as well as implementing coherent detection without training have not been adequately addressed (training for channel estimation is evidently impossible before the discovery of neighbors).

In this work, we propose a novel scheme based on *group testing*, which is highly scalable, only requires

simple non-coherent (energy) detection and incurs small overhead. A CDMA-like on-off signaling is proposed, where the signature of each user is a randomly produced binary sequence of 0's and 1's. The difference with the usual direct-sequence CDMA with frequency- or phase-shifted keying spreading sequence is that, during the *chips* or *mini-slots* corresponding to 0's in the sequence, the node transmits zero energy. The receive node simply detects whether there is energy in each chip, and infer about which nodes are present as neighbors based on the overall on-off pattern. The underlying assumption is of course that transmitters can switch on and off as frequently as the chip rate. This is feasible using today's technology because amplifiers have sharp response time.

Interestingly, the neighbor discovery scheme using on-off signatures can be viewed as a group testing problem. In general, the classical problem of group testing is to identify defective items out of a set of objects by exercising tests over a sequence of object pools. The aim is to discover all defective items with the fewest number of tests. Application of group testing to the design of efficient algorithms for contention resolution in random multiple-access communication systems has been studied (e.g., [8], [9]). It is shown that by querying a sequence of subsets of all the users, a central controller can identify all active users and resolve collision very quickly. Furthermore, the group testing techniques are extended to multiple-access systems with heterogeneous population of users, where different users may have different probabilities of being active [10]. Note that multiple-access based on group testing relies on a central controller to roll out an optimal plan of queries, whereas in ad hoc networks such controller is not available. Also, unlike the works in [8] and [10], the sequence of tests used in this paper is *predetermined*, which does not change over time.

The rest of the paper is organized as the following. In Section II, we describe the group testing technique and how it is applied to neighbor discovery. A direct algorithm for neighbor discovery based on group testing is proposed in Section III along with an upper bound on its error performance. A second algorithm with lower complexity is also proposed in the section. Both algorithms are shown to be efficient and effective using numerical results in Section IV. Section V concludes the paper.

II. Neighbor Discovery as Group Testing

Consider a network with $K+1$ nodes, indexed by $0, 1, \ldots, K$. Without loss of generality, consider the neighborhood of node 0. The problem of neighbor discovery is to collect the indices of the nodes which are in the neighborhood of node 0. An ALOHA type of random access discovery scheme is often considered, where each user sends its index through random access of the channel upon receipt of a beacon signal from node 0. Typically, it takes a number of transmissions to resolve contention and finish the discovery process.

In order for more rapid discovery, one can take advantage of the multiple access channel and let nodes simultaneously send their coded identity information in response to a beacon signal from node 0. The neighbor discovery problem is fundamentally a multiuser detection problem. Let X_k indicate whether node k is a neighbor of node 0, i.e., $X_k = 1$ denotes that node k is directly connected with node 0, whereas $X_k = 0$ denotes otherwise. Suppose $X_1, \ldots, X_K$ are independent and identically distributed (i.i.d.) Bernoulli random variables with parameter p. We also assume that node 0 typically has no more than a small number of neighbors, so that the vector $\boldsymbol{X} = [X_1, \ldots, X_K]^T$ is sparse. The goal of neighbor discovery is to infer about the elements of $\boldsymbol{X}$ based on the observation.

Consider the use of CDMA-type signaling, i.e., each node is assigned a signature and nodes transmit the signatures simultaneously when they receive the beacon from the center node (out-of-range nodes are not aware of the beacon signal and hence do not respond). Suppose node 0 has *a priori* knowledge of the correspondence between nodes and signatures. We use the $L \times K$ matrix $\underline{\boldsymbol{S}}$ to denote the signature matrix, i.e. its jth column $\boldsymbol{S}_j$ is the signature for node j. We use non-coherent detection at the receiver so that no channel estimation is needed. A simple and useful scheme is to use two values, 1 and 0, for elements (*chips*) of each signature. A 1 indicates some energy is transmitted during the corresponding mini-slot, while a 0 indicates no transmission during the mini-slot. Let the vector $\boldsymbol{Y}$ denote the received signal at node 0. In order to focus on the efficiency of the methodology proposed in this paper, we assume that the response from the neighbors are noiseless. Thus, node 0 detects energy in Y_j and set $Y_j = 1$ only when the jth element of at least one of the K signatures contains energy. Otherwise, Y_j is set to be 0. Therefore, $\boldsymbol{Y}$ can be regarded as the output of passing the signatures of the neighbors through an OR-channel.

Recovering $\boldsymbol{X}$ from $\boldsymbol{Y}$ can be done by exploring the results from group testing. The classical group testing problem is to identify defective items out of a set of objects by exercising pool queries whose output is

produced in the manner of an OR-channel. In group testing language, the matrix $\underline{\boldsymbol{S}}$ can be viewed as a *test plan*, where $S_{ij} = 1$ indicates node j is tested in the ith test and $S_{ij} = 0$ indicates otherwise. Accordingly, $\boldsymbol{Y}$ is considered as the *test output*. Thus, a group testing problem can be described using the following linear system,

$$\boldsymbol{Y} = \underline{\boldsymbol{S}}\boldsymbol{X}, \tag{1}$$

where the addition is the *inclusive or* operation. In general, the number of tests L can be a function of K which is to be designed. The aim is of classical group testing is to achieve perfect detection of the defective items using the fewest number of queries.

Similar to classical group testing, the goal of this paper is to recover $\boldsymbol{X}$ based on $\boldsymbol{Y}$ with sufficiently small error using a small number of tests L. Note that in neighbor discovery L can be interpreted as detection delay, which is also proportional to the transmission power consumption of neighbors and memory usage. Due to the sparsity of $\boldsymbol{X}$, very low discovery error probabilities can be achieved by using much fewer tests than the total number of nodes in the system (i.e., $L \ll K$). This can of course be regarded also as an instance of compressed sensing [11], while the algorithms designed in this paper are specialized and simple. In particular, using a randomly generated $\underline{\boldsymbol{S}}$ and each of its elements independently following some Bernoulli distribution with parameter q which is carefully picked, the recovery error rate averaged over all realizations of $\underline{\boldsymbol{S}}$ and $\boldsymbol{X}$ can be asymptotically upper bounded by $O(K^{-1})$.

III. Algorithms

This section presents two algorithms for recovering $\boldsymbol{X}$ from the observation $\boldsymbol{Y}$. Before describing the algorithms, we introduce some terminologies used in the group testing literature. We call an element of $\boldsymbol{Y}$ which equals to 1 a *positive test* and otherwise a *negative test*.

A. A Direct Algorithm

Note that a negative test rules out all nodes probed in this test as neighbors. By checking all negative tests, we can mark the nodes probed by these tests as non-neighboring nodes. The remaining unmarked nodes are called *undetermined* nodes. Note that those positive tests with only one node tested indicate such nodes as neighbors. In principle, one can search over all hypotheses of the set of neighbors to discover definite neighbors. In this work, we simply mark all undetermined nodes as neighbors. We propose the direct algorithm as follows.

Algorithm I: *Direct Algorithm*

1: **Input:** $\boldsymbol{Y}$ and $\underline{\boldsymbol{S}}$
2: $U \leftarrow \{1, \ldots, K\}$
3: **for** $i = 1$ to L **do**
4: **if** $\boldsymbol{Y}_i = 0$ **then**
5: $U \leftarrow U \setminus \{j : S_{i,j} = 1\}$
6: **end if**
7: **end for**
8: mark nodes in U as neighbors.

It is possible that there are still some non-neighboring nodes among the undetermined ones, and we call them *false alarms*. The following proposition analyzes the average number of false alarms and concludes that this average number can be extremely small when K is large and the test plan $\underline{\boldsymbol{S}}$ is carefully designed.

Proposition 1: In the neighbor discovery problem described by (1), suppose there are $K > 3$ nodes in the network and on average c of them are neighbors of node 0. If the elements of signature matrix $\underline{\boldsymbol{S}}$ are independently generated according to Bernoulli distribution with parameter q, then the number of false alarms averaged over all possible realizations of $\underline{\boldsymbol{S}}$ and $\boldsymbol{X}$, which is denoted by $\mathcal{E}$, is upper bounded as

$$\mathcal{E} \leq K^{-1} \exp\left[c\left(\exp\left(\frac{1}{\log\log K}\right) - 1\right)\right] \tag{2}$$

if we choose

$$q = \frac{1}{2\log K \log\log K} \tag{3}$$

and the length of the signatures to be

$$L = 4(\log K)^2 \log\log K \ . \tag{4}$$

Proof of Proposition 1: Because there are on average c nodes around node 0, the parameter $p = c/K$. For a given $\underline{\boldsymbol{s}}$ and a given $\boldsymbol{x}$, let $u(\underline{\boldsymbol{s}}, \boldsymbol{x})$ denote the number of undetermined nodes. We use $\mathcal{X}^K$ to denote the set of all possible realizations of $\boldsymbol{X}$ and use $P_{\boldsymbol{X}}(\cdot)$ to denote the probability mass function of the random variable $\boldsymbol{X}$. Define

$$\tilde{u}(\underline{\boldsymbol{s}}) = \sum_{\boldsymbol{x} \in \mathcal{X}^K} u(\underline{\boldsymbol{s}}, \boldsymbol{x}) P_{\boldsymbol{X}}(\boldsymbol{x}), \tag{5}$$

which is the number of false alarms averaged over $\mathcal{X}^K$ for a given $\underline{\boldsymbol{s}}$.

Let $\boldsymbol{s}_j$ denote the jth column of $\underline{\boldsymbol{s}}$. We adopt the convention that for two equal-length binary vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, the notation $\boldsymbol{a} \subseteq \boldsymbol{b}$ states that the set of indices

of non-zero entries in $\boldsymbol{a}$ is a subset of that in $\boldsymbol{b}$. Let $\mathcal{X}_i^K$ denote the subset of $\mathcal{X}^K$ whose elements have exactly i 1's. Thus,

$$\tilde{u}(\underline{\boldsymbol{s}}) = \sum_{j=1}^{K} \sum_{\boldsymbol{x}\in\mathcal{X}^K, \boldsymbol{s}_j\subseteq\underline{\boldsymbol{s}}\boldsymbol{x}} P_{\boldsymbol{X}}(\boldsymbol{x}) \tag{6}$$

$$= \sum_{j=1}^{K}\sum_{i=0}^{K} \sum_{\boldsymbol{x}\in\mathcal{X}_i^K, \boldsymbol{s}_j\subseteq\underline{\boldsymbol{s}}\boldsymbol{x}} P_{\boldsymbol{X}}(\boldsymbol{x}) \ . \tag{7}$$

Next we average $\tilde{u}(\underline{\boldsymbol{S}})$ over all possible realizations of $\underline{\boldsymbol{S}}$. Note that for given $j \in \{1,\ldots,K\}$ and $\boldsymbol{x} \in \mathcal{X}_i^K$, $\boldsymbol{S}_j \subseteq \underline{\boldsymbol{S}}\boldsymbol{x}$ if $i \geq 1$ and $\boldsymbol{x}_j = 1$, whereas if $\boldsymbol{x}_j = 0$ then $\boldsymbol{S}_j \subseteq \underline{\boldsymbol{S}}\boldsymbol{x}$ occurs if and only if none of the L rows of $\underline{\boldsymbol{S}}$ has a 1 in column j and a 0 in each of the i columns indexed by the 1's in $\boldsymbol{x}$. Hence, $P(\boldsymbol{S}_j \subseteq \underline{\boldsymbol{S}}\boldsymbol{x})$ equals 1 if $i \geq 1$ and $\boldsymbol{x}_j = 1$ and equals $(1-q(1-q)^i)^L$ if $\boldsymbol{x}_j = 0$. It follows that

$$\mathsf{E}\{\tilde{u}(\underline{\boldsymbol{S}})\}$$

$$= \sum_{i=0}^{K}(K-i)(1-q(1-q)^i)^L Q(i) + \sum_{i=0}^{K} iQ(i) \tag{8}$$

$$= K\sum_{i=0}^{K} Q(i)(1-q(1-q)^i)^L$$

$$+ \sum_{i=0}^{K} iQ(i)(1-(1-q(1-q)^i)^L) \tag{9}$$

where

$$Q(i) = \binom{K}{i} p^i (1-p)^{K-i}. \tag{10}$$

Because

$$(1-q(1-q)^i)^L \leq (1-q(1-iq))^L \leq e^{-q(1-iq)L}, \tag{11}$$

the first term in (9) does not exceed

$$K\sum_{i=0}^{K} Q(i) e^{-q(1-iq)L}$$

$$= Ke^{-qL}\left(1-p+pe^{q^2L}\right)^K \tag{12}$$

$$\leq Ke^{-qL+Kp(e^{q^2L}-1)}. \tag{13}$$

In addition, it is easy to see that the second term in (9) can be upper bounded by Kp. Therefore,

$$\mathsf{E}\{\tilde{u}(\underline{\boldsymbol{S}})\} \leq Ke^{-qL+Kp(e^{q^2L}-1)} + Kp \ . \tag{14}$$

Note that there are on average Kp active nodes, it follows that the average number of false alarms $\mathcal{E}$ has an upper bound given by

$$\mathcal{E} \leq Ke^{-qL+Kp(e^{q^2L}-1)}. \tag{15}$$

One can pick values for L and q as given in (4) and (3) to arrive at the upper bound (2). ■

Remark: In [12], Berger and Levenshtein studied the minimum number of tests required in classical group testing by adapting methods originally developed in information theory and coding theory. The upper bound (14) in the proof of Proposition 1 was also derived in [12] to illustrate the asymptotic efficiency of group testing. Here we repeat the derivation of (14) for completeness. There are many different choices of L and q other than (4) and (3) to arrive at different versions of upper bounds according to (15). In fact [11] provides several different choices. In this paper, it suffices to consider the special case described by (4) and (3).

The computational cost of Algorithm I mainly resides on the "for" loop, which introduces complexity of $O(K(\log K)^2 \log\log K)$ in order to address all elements in $\underline{\boldsymbol{S}}$. The set operation in line 8 contributes no more than K operations in total. Therefore, the complexity of Algorithm I is $O(K(\log K)^2 \log\log K)$.

Here we give a comparison between the the neighbor discovery scheme based on group testing and random access similar to the birthday-listen-and-transmit algorithm in [1]. Consider a network with ten thousand nodes and on average six nodes around node 0. We assume time is slotted and transmission of each bit takes one slot. In group testing based neighbor discovery, we need to assign $L \approx 754$ bits to each signature by equation (4). Thus, the group testing method uses 754 slots in total. There is no need for training and power control overhead. To quantify the comparison, we consider the probability of failure as the metric, namely the probability of the event where neighbor discovery is not successful either due to introducing false alarms or missing any neighbor. In group testing based discovery, the probability of failure is upper bounded by the average number of false alarms, which equals 0.003 in the example of this comparison.

As for the random access discovery scheme, we consider the case where nodes contend to announce themselves across a sequence of slotted contention periods. In each contention period, every neighbor independently chooses either to transmit (with probability η) or to listen (with probability $1-\eta$) and the choices are independent across contention periods. In order to characterize the probability of failure, we derive a lower bound by calculating the probability of one particular node among the neighbors being missed. Using the same $Q(\cdot)$ as in (10) with $p = 0.0006$, the lower bound θ for probability of failure after T contention periods

is given by

$$\theta = \sum_{i=1}^{K} Q(i) \left[1 - \eta\,(1-\eta)^{i-1}\right]^{T}. \qquad (16)$$

The smallest T that brings θ below 0.003 is 114, which is obtained when η is set to 0.11. Due to the lack of a central scheduler, in each contention period each neighbor has to transmit its identification sequence which takes at least $\log_2(10^4) \approx 14$ bits without counting in additional overhead such as preamble and parity check bits. Thus the total time expense is at least $114 \times 14 = 1596$ slots. In this case, using group testing discovery scheme is about 50% more economic than using random access discovery.

We note that the above analysis has not included the overhead in each packet transmission. In order to send the 14 bits of identity information reliably, many more bits have to be used as preambles for synchronization, parity checks for error control, etc. The overall impact of such overhead is much more significant in the case of random access because the additional cost is multiplied to the number of contention periods.

It is also interesting to calculate the expected number of slots required to hear all neighbors. Let B_i^{Λ} denote the expected number of contention periods required until hearing i specified nodes among Λ neighbors. We have the following recursive equation

$$\begin{aligned} B_i^{\Lambda} &= i\eta(1-\eta)^{\Lambda-1}(B_{i-1}^{\Lambda}+1) \\ &\quad + \left[1 - i\eta(1-\eta)^{\Lambda-1}\right](B_i^{\Lambda}+1). \end{aligned} \qquad (17)$$

Note that $B_1^{\Lambda} = \frac{1}{\eta(1-\eta)^{\Lambda-1}}$. Thus we get

$$B_i^{\Lambda} = \frac{1}{\eta(1-\eta)^{\Lambda-1}} \sum_{k=1}^{i} \frac{1}{k}\ . \qquad (18)$$

Averaging over all possible numbers of neighbors, the overall average number of contention periods is given by

$$\sum_{l=1}^{K} Q(l) B_l^l = \sum_{l=1}^{K} Q(l) \left[\frac{1}{\eta(1-\eta)^{l-1}} \sum_{k=1}^{l} \frac{1}{k}\right]\ . \qquad (19)$$

Therefore, with $\eta = 0.11$ which minimizes the lower bound of probability of failure, 43 contention periods are required on average to discover all neighbors, which leads to a total time expense of $43 \times 14 = 602$ slots. Taking into account the overhead, which accumulates linearly with the number of contention periods, the time expense associated with random access discovery can be much larger than 754 slots.

B. Group Testing with Binning: An Algorithm with Reduced Complexity

Although the direct algorithm performs well as the total number of nodes in the network K becomes large, e.g., over one million, its computation complexity becomes challenging. An efficient solution in case of a large K is to divide and conquer.

The key element of the algorithm proposed in this section is to use *binning* to decompose neighbor discovery among a large number of nodes into several smaller problems each of which involves much fewer candidates. We call this method *group testing with binning*. We introduce a parameter $\beta \in (0,1)$ which is the *binning exponent*. For convenience, we assume K^{β} to be an integer. The binning method contains $\lceil 1/\beta \rceil + 1$ stages. On stage $i \in \{1, \ldots, \lceil 1/\beta \rceil + 1\}$, using some mechanism, for example Hashing, we can randomly distribute all nodes into K^{β} bins with $\lceil K^{1-\beta} \rceil$ nodes in each bin. Each bin is then assigned an $L_i \times 1$ binary vector as its signature and every node in the same bin uses the bin signature as its own signature for stage i. Distributing nodes into bins as well as generating sets of bin signatures is independent across different stages. After stacking signatures for different stages together, the resulting signature for each node is a $(\sum_{i=1}^{\lceil 1/\beta \rceil + 1} L_i) \times 1$ vector. We give an example to illustrate the signature design with binning. Assuming $K = 10^6$ and $\beta = 1/2$, we have 3 stages and $\sqrt{K} = 1000$ bins with 1000 nodes in each bin on every stage. We take node 1 as an example. Suppose it is distributed into bin b_1, bin b_2 and bin b_3 on the consecutive three stages, and also suppose we generate three different sets of signatures, denoted by $\underline{\boldsymbol{S}}^1, \underline{\boldsymbol{S}}^2$ and $\underline{\boldsymbol{S}}^3$ respectively, the signature for node 1 is constructed by stacking three vectors and is represented by $[\boldsymbol{S}_{b_1}^{1\,T}, \boldsymbol{S}_{b_2}^{2\,T}, \boldsymbol{S}_{b_3}^{3\,T}]^T$.

During the neighbor discovery process, the neighboring nodes respond with their respective signatures after receiving a beacon signal from node 0. By applying the direct algorithm to the received signal associated with each stage, node 0 can find out which bins contain neighbors as well as which nodes are located in these neighbor-containing bins on each stage. According to the performance of the direct algorithm, when K^{β} is big enough, node 0 can detect the neighbor-containing bins with high accuracy. Note that for those *real* neighbors, they must be detected on every stage, while for those non-neighboring nodes, the probability that they appear in one of the detected bins on every stage can be extremely small. Thus the intersection of the nodes detected on different stages points out pretty much all neighbors. This observation inspires the group testing

with binning algorithm as is described in the following.

Algorithm II: *Group Testing with Binning*

1: **Input:** $\boldsymbol{Y}$, $\underline{\boldsymbol{S}}^1, \ldots, \underline{\boldsymbol{S}}^{\lceil 1/\beta \rceil + 1}$
2: **for** $t = 1$ to $\lceil 1/\beta \rceil + 1$ **do**
3: $P_t \leftarrow$ {the set of bin indices detected using the direct algorithm based on the part of $\boldsymbol{Y}$ corresponding to the t-th stage }
4: $U_t \leftarrow \{1 \le j \le K : \exists l \in P_t$, such that node j is located in bin $l\}$
5: **end for**
6: $I \leftarrow \bigcap_{t=1}^{\lceil 1/\beta \rceil + 1} U_t$
7: mark the nodes in I as neighbors

We give an estimate of the computational complexity of Algorithm II. Particularly, we can use an identical signature length $L_i = 4(\log K^\beta)^2 \log\log K^\beta$ for all i. The detection on $\lceil 1/\beta \rceil + 1$ stages results in a computational cost $O(\lceil 1/\beta \rceil K^\beta (\log K^\beta)^2 \log\log K^\beta)$ ($\lceil 1/\beta \rceil$ is counted because its value may depend on K). Another major cost comes from the intersection operation in line 6 of the algorithm. Because we can order the node indices in each bin on every stage at the initialization of the system which should be done once for all, hence sorting the nodes detected on each stage costs $O(\lceil K^{1-\beta} \rceil)$, and finding the intersection of sets of detected nodes on different stages costs $O(\lceil 1/\beta \rceil \lceil K^{1-\beta} \rceil)$. Therefore, the complexity of Algorithm II is given by $O(\lceil 1/\beta \rceil \max\{K^\beta (\log K^\beta)^2 \log\log K^\beta, \lceil K^{1-\beta} \rceil\})$. Oftentimes, one may design a system such that K^β is a constant. In this case, we can rewrite the complexity as $O(K \log K)$.

C. Comparison of Two Algorithms

In this section, we compare the two proposed algorithms using an example. Consider a network with one million nodes and on average six neighboring nodes. We choose $\beta = 2/3$ and have 3 stages as well as 10000 bins on each stage. Table I summarizes the comparison. It is clear that although the second algorithm with binning technique require 13% increase in the signature length, its computational cost is only 1.13% of that of the direct algorithm.

IV. Numerical Results

Numerical results are provided in this section in order to illustrate the efficiency of the two proposed algorithms. Consider a network with on average six neighbors for node 0. The random signatures are generated with length L given in (4) and parameter q given in (3). Fig. 1 plots the number of false alarms averaged over 2000 random instances of neighbor discovery using the direct algorithm against the total number of nodes in the network. Given the small number of total transmissions required by this algorithm, the discovery scheme is seen to be highly efficient. Fig. 2 plots the simulation results using group testing with binning. One can see the algorithm performs efficiently in large networks.

TABLE I
Comparison of Direct Algorithm and Group Testing with Binning Algorithm

	signature length (bits)	computation cost (operations)
direct algorithm	2.0×10^3	5.0×10^8
with binning	2.26×10^3	5.65×10^6

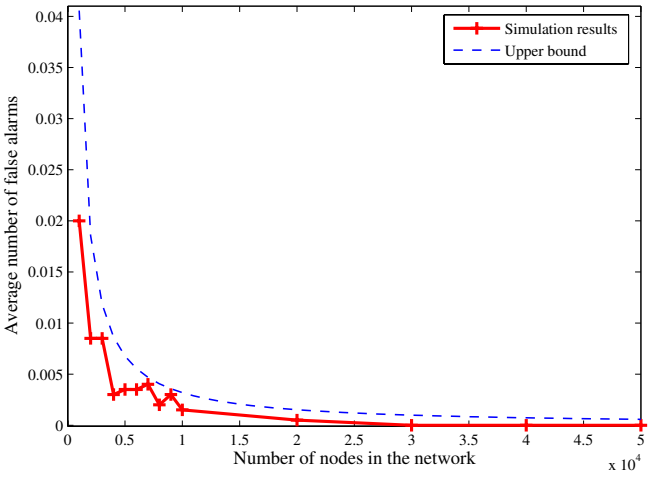

Fig. 1. Plot of the number of false alarms averaged over 2000 instances against the total number of nodes using the direct algorithm. The blue dashed curve is the upper bound computed according to Proposition 1 and the red curve with cross markers is the average number calculated from simulation.

V. Conclusion and Discussion

Efficient neighbor discovery scheme based on group testing has been proposed and analyzed in this paper. Unlike conventional multiuser detection methods, the proposed scheme requires only non-coherent energy detection and incurs little overhead. Two algorithms are developed to implement the proposed scheme, both of which achieve high discovery accuracy much more rapidly than random access discovery schemes.

Although this paper only considers neighbor discovery for one particular node, the group testing scheme

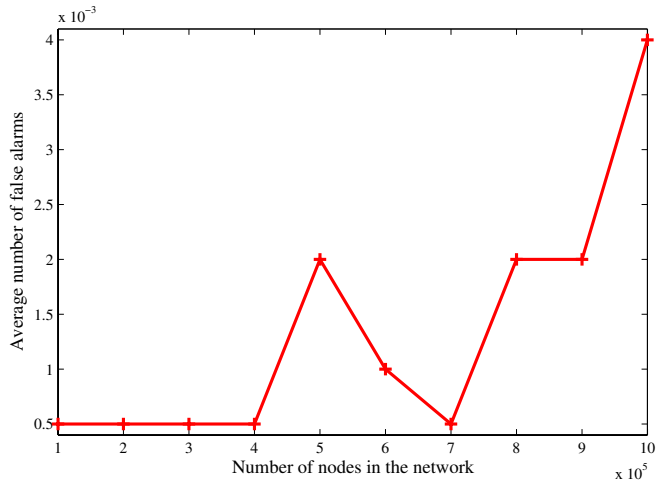

Fig. 2. Plot of the number of false alarms averaged over 2000 instances against the total number of nodes using the group testing with binning algorithm.

can be extended to neighbor discovery for all nodes in a network using cross-layer design. Integration of the group testing scheme in physical layer and the birthday-listen-and-transmit scheme [1] in medium access control layer serves as a promising solution.

Note that when only a small portion of all nodes in the network are neighbors, neighbor discovery can also be regarded as a compressed sensing problem, which studies the recovery of a sparse signal from its random projection which is of a much lower dimension than the original signal [11]. The group testing scheme proposed in this work may imply interesting applications in compressed sensing.

Finally, throughout the work we have assumed noiseless transmissions. Implementation of group testing in presence of noise is a direction of ongoing work.

REFERENCES

[1] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proceedings of the 2nd ACM International Symposium on Mobile Ad hoc Networking & Computing*, pp. 137–145, Long Beach, CA, USA, Oct. 2001.

[2] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," in *Proceedings of IEEE Infocom*, vol. 4, pp. 2502 – 2512, Amherst, MA, USA, Mar. 2005.

[3] Z. Zhang, "Performance of neighbor discovery algorithms in mobile ad hoc self-configuring networks with directional antennas," in *Proc. IEEE Military Communications Conference*, vol. 5, pp. 3162–3168, Oct. 2005.

[4] S. A. Borbash, A. Ephremides, and M. J. McGlynn, "An asynchronous neighbor discovery algorithm for wireless sensor networks," *Ad Hoc Networks*, vol. 5, pp. 998–1016, Sept. 2007.

[5] D. D. Lin and T. J. Lim, "Subspace-based active user identification for a collision-free slotted ad hoc network," *IEEE Trans. Commun.*, vol. 52, pp. 612–621, Apr. 2004.

[6] D. Angelosante, E. Biglieri, and M. Lops, "Neighbor discovery in wireless networks: A multiuser-detection approach," in *Proc. Information Theory and Applications Workshop*, pp. 46–53, Feb. 2007.

[7] D. Angelosante, E. Biglieri, and M. Lops, "A simple algorithm for neighbor discovery in wireless networks," in *Proc. IEEE Int'l Conf. Acoustics, Speech and Signal Processing*, vol. 3, pp. 169–172, April 2007.

[8] T. Berger, N. Mehravari, D. Towsley, and J. Wolf, "Random multiple-access communication and group testing," *IEEE Trans. Commun.*, vol. 32, pp. 769–779, July 1984.

[9] D. Stevens and M. Ammar, "Strategies for group testing in multihop radio networks," vol. 2, pp. 1285–1289, May 1993.

[10] D. Kurtz and M. Sidi, "Multiple-access algorithms via group testing for heterogeneous population of users," *IEEE Trans. Commun.*, vol. 36, pp. 1316–1323, December 1988.

[11] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Trans. Inform. Theory*, vol. 52, pp. 4036–4048, Sept. 2006.

[12] T. Berger and V. I. Levenshtein, "Asymptotic efficiency of two-state disjunctive testing," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1741–1749, July 2002.