ALOHA-Like Neighbor Discovery in Low-Duty-Cycle Wireless Sensor Networks

Lizhao You†, Zimu Yuan‡, Panlong Yang§, Guihai Chen†
†State Key Laboratory for Novel Software Technology, Nanjing University, China
‡Institute of Computing Technology, CAS, China
§Institute of Communication Engineering, PLA University of Science and Technology, China
youlizhao.nju@gmail.com, yuanzimu@ict.ac.cn, panlongyang@gmail.com, gchen@nju.edu.cn

***Abstract*—Neighbor discovery is an essential step for the self-organization of wireless sensor networks. Many algorithms have been proposed for efficient neighbor discovery. However, most of those algorithms need nodes to keep active during the process of neighbor discovery, which might be difficult for low-duty-cycle wireless sensor networks in many real deployments. In this paper, we investigate the problem of neighbor discovery in low-duty-cycle wireless sensor networks. We give an ALOHA-like algorithm and analyze the expected time to discover all $n-1$ neighbors for each node. By reducing the analysis to the classical *K Coupon Collector's Problem*, we show that the upper bound is $ne(\log_2 n + (3\log_2 n - 1)\log_2\log_2 n + c)$ with high probability, for some constant c, where e is the base of natural logarithm. Furthermore, not knowing number of neighbors leads to no more than a factor of two slowdown in the algorithm performance. Then, we validate our theoretical results by extensive simulations, and explore the performance of different algorithms in duty-cycle and non-duty-cycle networks. Finally, we apply our approach to analyze the scenario of unreliable links in low-duty-cycle wireless sensor networks.**

***Index Terms*—wireless sensor networks; duty cycle; neighbor discovery; performance analysis;**

I. INTRODUCTION

Wireless Sensor Networks (WSN) have been used for many long-term applications, such as environment monitoring [1], and science exploration [2]. Without the support of central infrastructures, sensor nodes must coordinate among themselves to form a multi-hop network. Neighbor discovery is one of the most fundamental coordinations in the self-configuration of WSNs. Knowledge of one-hop neighbors is especially useful for further operations, such as topology control, medium access control protocols.

The probabilistic neighbor discovery algorithms [3]-[6] are most commonly used in wireless networks, in which each node transmits at randomly chosen times and discovers all its neighbors by a given time with high probability [6]. However, all the proposed algorithms need network nodes to keep active during the process of neighbor discovery (named *all-node-active assumption*), which is unrealistic for long-term WSNs in real deployments. In those applications, in order to prolong the network lifetime, sensor nodes are always expected to work under duty-cycle [7] to reduce the energy waste, switching alternately between the dormant and active state.

For low-duty-cycle WSNs, one aggressive way to do neighbor discovery is to wake up all nodes, and then apply exiting discovery algorithms, which is hard to achieve. First of all, in practical scenarios featuring a time-consuming deployment phase, predicting the exact duration of the deployment process is usually hard [8]. Second, the mode switch from duty-cycle to all-node-active, then from all-node-active to duty-cycle, can be very hard due to the synchronization and coordination problem in a large-scale network. Therefore, it is necessary to consider the problem in duty-cycle scenario, and the analysis provides us with the basic understanding of the duty-cycle neighbor discovery in low-duty-cycle WSNs.

In this paper we study the problem of neighbor discovery when nodes use ALOHA-like slotted discovery algorithms in low-duty-cycle WSNs. This problem is non-trivial because a node may have to transmit many times without collisions to make all its neighbors discover it, while one time is enough for all-node-active networks.

By reducing the analysis to *K Coupon Collector's Problem*, we show that the expected time to discover all its $n-1$ neighbors for each node is upper bounded by $ne(\log_2 n + (3\log_2 n - 1)\log_2\log_2 n + c)$ for some constant c with high probability, and it is lower bounded by $ne\ln n + cn$, where c is a positive constant and e is the base of natural logarithm. Moreover, we show that the discovery time is around the expectation. Also, we extend the ALOHA-like algorithm to deal with the case when nodes do not know the number of neighbors. It only leads to at most a factor of two slowdown for upper bound compared with known number of neighbors. Then we validate our theoretical results by extensive simulations, and explore the different algorithm performance in low-duty-cycle and non-duty-cycle WSNs. Moreover, we apply our approach to analyze new scenario of unreliable links in low-duty-cycle WSNs. To the best of our knowledge, ours is the first work to analyze the performance of ALOHA-like neighbor discovery algorithms in low-duty-cycle WSNs.

The rest of the paper is organized as follows. Section II introduces the model and assumption. Section III describes the ALOHA-like discovery algorithm and its analysis. Section IV considers the case when nodes do not know the number of neighbors. Then we show our simulation results in Section V.

Section VI extends our discussion to a new scenario. Finally, we conclude in Section VII and give the future work.

II. Network Model and Assumptions

In this section, we present the neighbor discovery model in low-duty-cycle WSNs, and give some critical assumptions as follows:

- Time model: Time is divided into equal-length slots, and nodes are synchronized on slot boundaries.
- Duty-cycle model: Nodes work in duty-cycle, and the duty cycle is defined as the ratio of the active interval to a complete cycle of active and dormant interval.
- Radio model: Nodes in transmission range of each other can communicate, and the errors caused by fading is negligible. In other words, once packets are transmitted without a collision, they must be received correctly.

Furthermore, each node has a locally unique identifier, which can be used for neighbor discovery. A collision happens at the receiver, if two or more nodes transmit at the same slot, and there is no collision resolution mechanism. The feedback from receiver to transmitter is also not considered in this paper.

We notice that the time and radio model considered in this paper are idealized. However, they can help us to gain the fundamental understanding of neighbor discovery algorithms in low-duty-cycle WSNs. Some discussion of unrealistic radio model is in Section VI. Some techniques can be utilized to relax our idealized model, e.g. realistic radio model [5], asynchronous time model [6], which is left for our future work.

III. ALOHA-like Neighbor Discovery

In this section, we analyze the performance of ALOHA-like discovery algorithm in a clique of size n, and n is known to all nodes in the clique. Although the analysis is based on a clique, we show that it can be applied to a network in Section V. Moreover, the assumption of known number of neighbors is relaxed in Section IV. The notations used in this section are summarized in Table I.

TABLE I
TERMINOLOGY

Symbol	Definition
n	Number of nodes in a clique.
p_w	Probability that node is active in each time slot.
p_t	Probability that active node transmits in each time slot.
p_s	Probability that a given node transmits successfully in a clique of n nodes in each time slot. $p_s = p_w p_t(1 - p_w p_t)^{n-1}$.
p_c	Probability that some node transmits successfully in a clique of n nodes in each time slot. $p_c = np_s$
ξ	Event that some node transmits successfully in a clique of n nodes in each time slot. $P[\xi] = p_c = np_s$.
K	Time slots of successful transmission needed for a given node to be discovered by all $n-1$ neighbors in a clique with high probability, and only the given node transmits successfully in those time slots.
T	Time slots of successful transmission needed for the end of neighbor discovery in a clique of n nodes with high probability.
W	Random variable of the neighbor discovery time for a clique of n nodes.

A. Algorithm Description

Assume in each time slot, each node independently chooses to be active with identical probability p_w, and be dormant with probability $1 - p_w$. The duty cycle p_w is a constant, which is set by specific application before deployment. Here, we take a probabilistic approach to define the duty cycle in accord with the ALOHA-like algorithm. For those active nodes, they independently and identically transmit with probability p_t, and listen with probability $1-p_t$. Note that the dormant nodes can not transmit or listen, and p_t is the parameter to be determined. We refer to this algorithm as Algorithm 0.

Let's consider a clique of n nodes numbered $\{1, ..., n\}$. When two or more nodes transmit simultaneously in the same slot, there is a collision. In order to maximize the expected number of neighbors discovered in a single slot, we must determine the product of two parameters $p_w p_t$, which is the probability to transmit for each node. The optimal choice of $p_w p_t$ can be shown to be $1/n$, where n is the clique size. The detailed proof can be found in [3]. Intuitively, we can treat $p_w p_t$ as the transmission probability in all-node-active networks, and the optimization process is almost the same with all-node-active networks.

In the reminder of this section, we set $p_w p_t = 1/n$, where n is the clique size. Furthermore, we assume $p_w = 1/2$ in the following analysis for simplicity. As a matter of fact, a positive constant is also workable. Given the current setting, we study the time required to discover all neighbors for the clique. Before doing so, we give a definition:

Definition 1. *Neighbor discovery time for a clique of n nodes is the number of slots to discover all $n(n-1)$ links in a clique.*

Let p_s be the probability that a given node transmits successfully in each time slot. Given the fact which is proven in [6]:

$$(1 - \frac{1}{k})^{k-1} \geq \frac{1}{e}, \forall k = 1, 2, \cdots \quad (1)$$

where e is the base of natural logarithm. Then we have:

$$p_s = p_w p_t(1 - p_w p_t)^{n-1} = \frac{1}{n}(1 - \frac{1}{n})^{n-1} \geq \frac{1}{ne} \quad (2)$$

Since nodes are symmetric in a clique, the probability that some node transmits successfully in a given time slot is np_s.

B. Expected Discovery Time

If nodes are always active, we can easily use the *Coupon Collector's problem* to analyze the expected neighbor discovery time, which is shown in [6]. However, the duty-cycle operation complicates the problem, as a node may have to transmit successfully many times to make all its neighbors discover it. In this section, we show that the upper bound analysis can be reduced to the *K Coupon Collector's Problem.*

1) Upper Bound: Given a clique of n nodes numbered $\{1, \cdots, n\}$, it must be in one of three states in each time slot: *successful transmission* (S), *idle* (I), or *collision* (C). During the *successful transmission* state, some node transmits successfully. During the *idle* state, the channel is clean, which might be caused by no active nodes or no transmissions among active nodes. During the *collision* state, two or more active nodes transmit simultaneously. Let event ξ denote that the clique is in S state, and it is easy to see that $P[\xi] = np_s$.

Firstly, we give two lemmas:

Lemma 1. *Given a clique of n nodes numbered $\{1, ..., n\}$, if node j, $1 \leq j \leq n$, transmits $K = 3\log_2 n$ slots without a collision, all its $n-1$ neighbors discover node j with high probability*[1].

Proof: We assume that the transmission from node j is always successful in all K slots. The K slots may distribute over all the time slots, and does not need to be continuous. We define event ε as follows: after transmitting K slots, there exist some nodes not receiving any discovery messages from node j, and event x_i, $1 \leq i \leq n$ and $i \neq j$: node i does not receive any discovery messages sent from node j within K slots. We observe that $P[x_i] = (1-p_w)^K$ since node i must be in the dormant state in all K slots. Then

$$P[\varepsilon] = P[x_1 \vee ... \vee x_n] \leq nP[x_i] = n(1-p_w)^K \quad (3)$$

The inequality follows from *Union Bound*. Substituting $K = 3\log_2 n$ and $p_w = 1/2$ into equation (3), we get $P[\varepsilon] \leq 1/n^2$. In other words, after node j transmits K slots successfully, all nodes in the clique discover node j with high probability. So the proof is complete. ■

Lemma 2. *Given a clique of n nodes and*

$$T = n(\log_2 n + (3\log_2 n - 1)\log_2\log_2 n + c)$$

for some constant c, if event ξ happens T times, the neighbor discovery process for the clique is finished with high probability.

Proof: We can regard a successful transmission of node i as collecting coupon i, $1 \leq i \leq n$. By Lemma 1, only when we collect each coupon for at least $K = 3\log_2 n$ times, the discovery process is finished w.h.p., so the problem can be reduced to *K Coupon Collector's Problem*. Let random variable T_K denote the time slots needed to collect K copies of each coupon for the first time. [9] showed that:

$$P(T_K < n(\log_2 n + (K-1)\log_2\log_2 n + c)) = e^{-\frac{e^{-c}}{(K-1)!}}$$

When all nodes transmit successfully K times, the probability that the neighbor discovery process is finished is given by $(1-1/n^2)^n$. So if event ξ happens T times, the probability that the discovery process is finished is $p' = (1-\frac{1}{n^2})^n e^{-\frac{e^{-c}}{(K-1)!}}$. Observe that $e^{-\frac{e^{-c}}{(K-1)!}}$ and $(1-1/n^2)^n$ is close to 1 for large positive c when n goes to infinity. It is easy to show $1 - p' = o(1)$, so the proof is complete. ■

Based on two lemmas, we give a theorem as follows:

[1]In this paper an event is said to happen with high probability (*w.h.p.*), if the probability that event happens is given by $f(n)$ for any random network of size n, and f(n)=1-o(1), where g(n)=o(1) means $\lim_{n\to\infty}\frac{g(n)}{c} = 0$ for a positive constant c.

Theorem 1. *Let W be the neighbor discovery time for a clique of n nodes, and the expectation of W is upper bounded by*

$$E[W] \leq ne(\log_2 n + (3\log_2 n - 1)\log_2\log_2 n + c) \quad (4)$$

with high probability, where c is a positive constant.

Proof: Consider the state of a clique in each time slot, event ξ happens with probability $p_c = np_s$. W is the needed time slots when ξ happens $T = n\log_2 n + (3\log_2 n - 1)n\log_2\log_2 n + cn$ times, so W is a Pascal random variable with parameter np_s and T. Note that *Pascal Distribution* equals to *Negative Binomial Distribution* after some simple manipulations. For *Pascal distribution*, we have $E[W] = T/np_s$. Given equation (2), we have $np_s = (1-\frac{1}{n})^{n-1} \geq 1/e$, then

$$E[W] \leq eT = ne(\log_2 n + (3\log_2 n - 1)\log_2\log_2 n + c)$$

w.h.p.. So the proof is complete. ■

Note that the result in low-duty-cycle WSNs is almost a factor of $3\ln\ln n$ slowdown over all-node-active WSNs [6].

2) Lower Bound: Consider a fictitious node C which is regarded as active all the time. Let the random variable L denote the number of slots needed for the fictitious node C to discover all n nodes in the clique. It is obvious that $P[L > t] \leq P[W > t]$. In other words, $E[L] \leq E[W]$, so $E[L]$ is the lower bound for $E[W]$. By the similar analysis of *Coupon Collector's Problem* [6], we have

$$E[W] \geq E[L] = ne(\ln n + \Theta(1)) = ne\ln n + O(n) \quad (5)$$

C. Bounds on Deviation From Expectation

Until now, we know the upper bound of expected discovery time for a clique of n nodes is $O(n\log_2 n\log_2\log_2 n)$, and the lower bound is $O(n\ln n)$. We next show that the deviation of W from expectation can be bounded.

Remember W is a Pascal random variable with parameter np_s and T, and $E[W] = T/np_s$, $Var[W] = T(1-np_s)/(np_s)^2$. Using *Chebshev's Inequality*, we have

$$P[|W - E[W]| \geq E[W]] \leq \frac{Var[W]}{(E[W])^2} \leq \frac{1-np_s}{T}$$

Since the successful time slots T must be at least n to receive all n nodes, $T \geq n$, and $np_s = (1-\frac{1}{n})^{n-1} \geq 1/e$ by equation (2). The former inequality can be rewritten as:

$$P[W \geq 2E[W]] \leq P[|W - E[W]| \geq E[W]] \leq \frac{1-\frac{1}{e}}{n}$$

It is easy to show $\frac{1-\frac{1}{e}}{n} = o(1)$, so W is less than $2E[W]$ with high probability, i.e.

$$W \leq 2ne(\log_2 n + (3\log_2 n - 1)\log_2\log_2 n + c) \quad (6)$$

IV. UNKNOWN NUMBER OF NEIGHBORS

In this section, we will relax the assumption that each node knows the number of neighbors $n-1$. Note that with the neighbor knowledge, it is easy to set the product of active and transmitting probability $p_w p_t = 1/n$, and terminate the discovery process within our derived upper bound. However, it is hard to get the neighbor knowledge in practical environment. So we consider the case that nodes have no knowledge of the average number of neighbors, and even have no estimate of neighbors.

A. Algorithm Description

The new algorithm is named Algorithm 1. The algorithm behaves in the unit of phases, while a phase includes some time slots. The operation is like this: in phase i, each node is active and transmits with probability $1/2^i$, i.e. $p_w p_t = 1/2^i$. Phase i lasts for $2^{i+1}e((3i-1)\log_2 i + i + c)$ slots, where c is a positive constant.

The key idea is that: each node geometrically decreases the probability $p_w p_t$ until entering the desired $\lceil \log_2 n \rceil$-th phase. Note that in that phase, $p_w p_t \approx 1/n$ and the total slots are $2ne((3\log_2 n - 1)\log_2 \log_2 n + \log_2 n + c)$. According to the analysis in Section III, each node in the clique can find all its neighbors in $\lceil \log n \rceil$-th phase with high probability.

B. Time Analysis

Now we calculate the total time slots denoted by W. W equals to the sum of the total number of slots in the past $\lceil \log n \rceil$ epoches, and is given by:

$$W = \sum_{i=1}^{\lceil \log_2 n \rceil} 2^{i+1} e((3i-1)\log_2 i + i + c)$$

$$W \leq 2e \sum_{i=1}^{\lceil \log_2 n \rceil} 2^i (3i \log_2 \lceil \log_2 n \rceil + i + c)$$

$$W \leq 2e((3\log_2 \lceil \log_2 n \rceil + 1) \sum_{i=1}^{\lceil \log_2 n \rceil} 2^i i + c \sum_{i=1}^{\lceil \log_2 n \rceil} 2^i)$$

Given the result from [6]: $\sum_{i=1}^{\lceil \log n \rceil} i2^i = 2n(\log n - 1) + 2$, $\sum_{i=1}^{\lceil \log n \rceil} 2^i = 2n - 2$. Assume $k' = \log_2 \lceil \log_2 n \rceil$, we have:

$$W \leq 2e((3k'+1)(2n(\log_2 n - 1) + 2) + c(2n-2))$$

$$W \leq 4e((3k'+1)(n(\log_2 n - 1) + 1) + c(n-1))$$

Given $n > 3k'$ for large n and $c > 1$, we have

$$\begin{aligned} W &\leq 4ne((3k'(\log_2 n - 1) + (\log_2 n - 1) + (c+1)) \\ &\leq 4ne((3k'(\log_2 n - 1) + \log_2 n + c) \\ &\leq 4ne((3\log_2 n - 1)k' + \log_2 n + c) \end{aligned}$$

Suppose $k' = \log_2 \log_2 n$ for simplicity, then we have

$$W \leq 4ne((3\log_2 n - 1)\log_2 \log_2 n + \log_2 n + c) \quad (7)$$

Comparing equation (7) with (6), we conclude that without the knowledge of the number of neighbors only leads to no more than a factor of two slowdown for the upper bound.

C. Termination Condition

Let X_j be the number of neighbors discovered for each node in phase j. For Algorithm 1, the termination condition for each node is as follows.

TC Stop at the end of phase $j+1$ if $X_j \geq 2^{j-1} \wedge X_{j+1} \leq 2^j$.

With termination condition in Algorithm 1, each node can independently terminate the neighbor discovery process. The key idea is that in each phase, it may have discovered enough number of neighbors corresponding to that phase w.h.p.. Let m be the largest integer which makes $n = 2^m + k$, $0 < k \leq 2^m$. The proof is almost the same with [6]. We show that each node stops the process at the $m+2$-th phase w.h.p..

Consider the phase $m+1$, $2^{m+1} \geq n$, $p_w p_t = 1/2^{m+1}$,

$$p_s^{m+1} = \frac{1}{2^{m+1}}(1 - \frac{1}{2^{m+1}})^{2^m + k - 1}$$

$$p_s^{m+1} \geq \frac{1}{2^{m+1}}(1 - \frac{1}{2^{m+1}})^{2^{m+1}-1} \geq \frac{1}{2^{m+1}e}$$

Applying the analysis in Section III, we have $X_i^{m+1} = n$ with probability $(1 - \frac{1}{n^2})^n e^{-\frac{e^{-c}}{(K-1)!}}$, where $K = 3\log_2 n$. For phase $m+2$, we have the similar result $X_i^{m+2} = n$ with probability $(1 - \frac{1}{n^2})^n e^{-\frac{e^{-c}}{(K-1)!}}$. Since the events in different phase are independent, the event $\{X_i^{m+1} \geq 2^m \wedge X_i^{m+2} \leq 2^{m+1}\}$ happens w.h.p., given the fact $1 - (1 - \frac{1}{n^2})^{2n} e^{-\frac{2e^{-c}}{(K-1)!}} = o(1)$ for large positive c. So our algorithm stops by the end of $m+2$ phase w.h.p.. We also investigate the case when our algorithm terminates before phase $m+2$. We simulate cliques of size from 3 to 100, and run 100 times for each size. Each node always terminates in $m+2$ phase except some cases when the size is 3. The extension of termination condition in a multi-hop network is to double the phase length and announce the termination in the second half of the phase, the same approach proposed in [6].

V. PERFORMANCE EVALUATION

A. Experiments Setup

Our simulation under Matlab includes two sets of setting: network and clique. Network setting is a 2D plane of a 3km×3km area, and nodes are generated by a uniform distribution, giving a transmission range of 150m. The average number of neighbors per node is set from 3 to 25 so as to look into the algorithm performance in different number of neighbors. The number of nodes in a network is determined by the density, for example, with the average number of neighbors 16, we have a total of 2000 nodes. For each density, we run 20 times in order to validate the result in different node placement. The point is given as average value of all runs.

Firstly we compare the derived upper and lower bound of Algorithm 0 with simulation results in network setting to valid of our clique assumption in analysis, and then validate Algorithm 1 in clique setting. Furthermore, we explore the difference of performance between duty-cycle and non-duty-cycle algorithm to gain deep understanding of our duty-cycle algorithm in clique setting. Finally, we investigate the impact of different duty-cycle in clique setting.

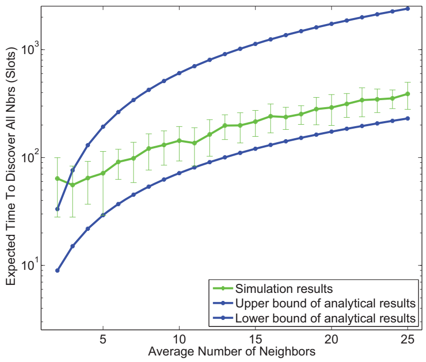

Fig. 1. Validation of Algorithm 0

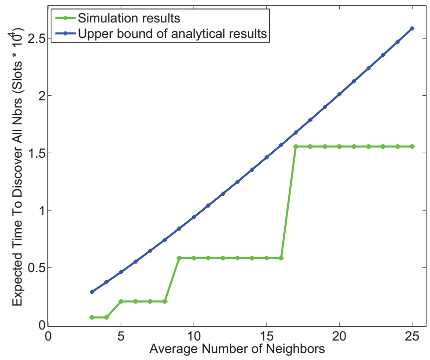

Fig. 2. Validation of Algorithm 1

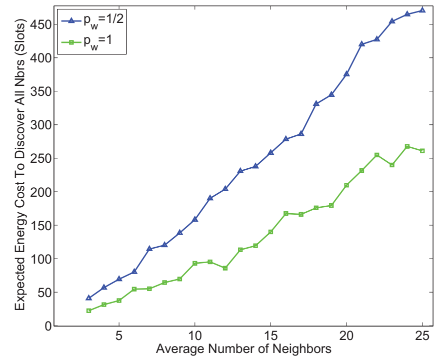

Fig. 3. Expected Energy Cost in Clique

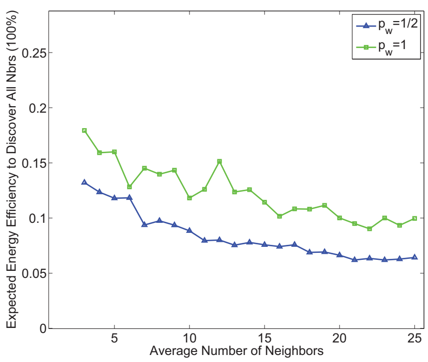

Fig. 4. Expected Energy Efficiency in Clique

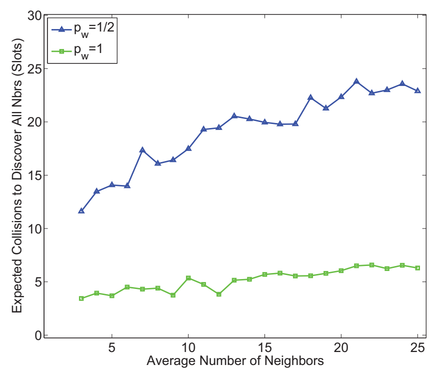

Fig. 5. Expected Number of Collisions in Clique

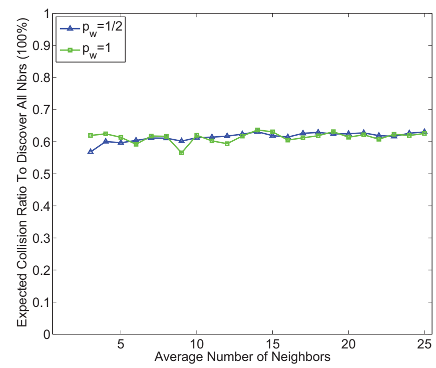

Fig. 6. Expected Ratio of Collisions in Clique

B. Simulation Results

1) Validation of Clique Assumption: Figure 1 depicts the comparison of the simulation and analytical results including the upper bound and lower bound defined in eqs. (5) and (6) with $c = 0$, and the 95% confidence intervals are also given. Note that $p_w = 1/2$ and $p_w p_t = 1/n$ since our derived results are based on it. We observe a close match between lower bound and simulation results, while the upper bound is a little looser. The reason why clique analysis fits a network is that we calculate the expected discovery time, which is averaged over all nodes in a network. For a network, central nodes have more neighbors than average density, while nodes in boundary have less neighbors than average density. Thus, central nodes may have larger discovery time, and boundary nodes have smaller discovery time. As a result, the average discovery time can be represented by average density in the network. The mismatch when n is small, e.g. $n = 3$, is due to the large transmission probability $p_w = 2/3$ leading to more collisions. The result is in accord with the conclusion in [6], further validating the assumption that analysis in a clique can represent the analysis in a network. Figure 2 shows the performance of Algorithm 1. The upper bound is given by equation (7) with $c = 40$, and simulation result matches theoretical result. In particular, for some density the result is the same, that is because the execution of Algorithm 1 is based on phase, and nodes stop at the same phase. Note that for $n = 3$ there is a small deviation (too small to be visible in the graph), because some nodes stop too early. However, it rarely happens, which verifies the effectiveness of our terminal condition.

2) Comparison between Duty-cycle and Non-duty-cycle: In order to compare the setting of duty-cycle and non-duty-cycle, we define two metrics: energy cost and energy efficiency. Energy cost is defined as the sum of transmission cost and reception cost, while the cost is calculated in individual node and averaged among all nodes. Transmission cost means the times slots spent on transmission, while reception cost means the time slots spent on reception. Furthermore, the transmission and reception are effective if the node transmits without a collision or discovers a new neighbor. Therefore, the energy efficiency is defined as the ratio of effective time slots to total time slots. In current setting, we suppose the energy cost for transmission and reception is 1:1, which can be adjusted by more accurate power assumption model. Both metrics provide insightful investigation about the difference between two scenarios.

For both algorithms, $p_w p_t = 1/n$. The only difference of these two algorithms is whether nodes choose to be dormant. For duty-cycle algorithm, duty cycle $p_w < 1$, while $p_w = 1$ for non-duty-cycle algorithm. For simplicity, we only consider the scenario of known number of neighbors. In the simulation, we set $p_w = 1/2$ for duty-cycle algorithm since our analysis is based on it.

Figure 3 shows the energy cost, and figure 4 depicts the energy efficiency. The cost is obviously larger for duty-cycle

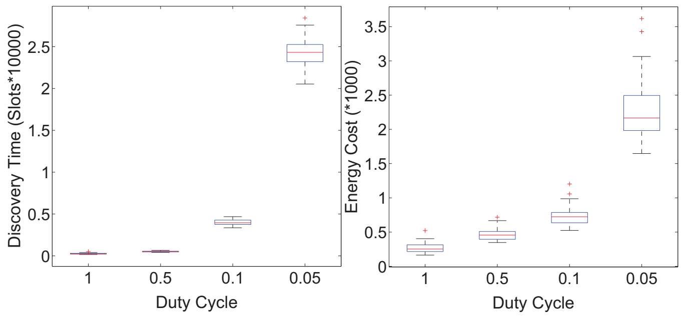

Fig. 7. We compare the discovery time and energy cost when a clique of 25 nodes is in different duty cycle. The smaller duty cycle, the larger the time and the cost. There is a distinct increase when the duty cycle is low (5%).

algorithm from figure 3, especially for large n, because the duty-cycle algorithm increases the time to finish neighbor discovery. However, considering a time slot, active nodes in duty-cycle setting are fewer than non-duty-cycle algorithm, leading to fewer collisions, so the energy efficiency is closer.

In order to further explain the close energy efficiency, we measure the number of collisions in both algorithms, and calculate the ratio of collision time slots to transmissions time slots. Figure 5 shows that collision also increases in duty-cycle setting, due to the same reason of longer discovery time. However, figure 6 indicates the ratio remains to be very close, further explaining the closeness of energy efficiency. The comparison between these two algorithms tell us duty-cycle indeed costs more energy, but the efficiency is close.

3) Impact of Duty Cycle: Figure 7 shows the energy cost and discovery time when nodes wake up with different duty cycle (100%, 50%, 10%, 5%) for 25 nodes. The box plots depict that, when duty cycle decreases, the cost and discovery time increase. If the duty cycle becomes low, e.g. 5%, the cost and discovery time rise obviously. Intuitively, the duty cycle is too low, and the fewer nodes are active, thus increasing the transmission times. This graph shows that there is a trade-off between duty cycle and the cost of neighbor discovery.

VI. DISCUSSION

After presenting our analysis results and evaluation, we now apply our model to analyze the new scenario of unreliable links in low-duty-cycle WSNs. To simplify the discussion, the probability that all the links in a clique are lossy is independent and identical. Furthermore, the time slot is synchronized.

Under this assumption, each node can sleep with probability p_d, or transmit with probability p_t, otherwise listen with probability $1 - p_d - p_t$. However, since the link is lossy, we can further divide the listen state into valid and invalid states. The receivers can decode the discovery packets successfully in valid listening state, while not in invalid state. The node is in valid listening state with probability p_v and invalid with probability p_l. Hence $p_t + p_d + p_v + p_l = 1$. Note that if $p_d = 0$, we are in all-node-active networks. Moreover, we observe that this scenario is similar to our discussion in this paper, since the link can be lossy with some probability, leading to many transmissions to make it known to all its neighbors. By a similar analysis in former sections, we can derive that the upper bound of expected neighbor discovery time is $ne(n \log_2 n + (3 \log_{\frac{1}{p_d+p_l}} n - 1) \log_2 \log_2 n + c)$ for some constant c with high probability.

VII. CONCLUSION AND FUTURE WORK

In this paper, we revisit the problem of neighbor discovery in low-duty-cycle WSNs, and derive the time bound required to discovery all the neighbors. For ALOHA-like discovery algorithm with known number of neighbors, the upper bound is $ne(\log_2 n + (3 \log_2 n - 1) \log_2 \log_2 n + c)$ for a constant c with high probability, and the lower bound is $ne \ln n + cn$ for a constant c, where e is the base of natural logarithm. Also, the time is around the expectation with high probability. By considering the scenario of unknown number of neighbors, we show that the upper bound is only a factor of two slowdown compared with known number of neighbors. The analysis result is further verified by extensive simulations. Furthermore, we show that our analysis can be applied to the scenario of unreliable links in low-duty-cycle WSNs.

In the future, we would like to extend our work by considering more practical model, e.g. asynchronous time model, more practical radio model and heterogenous duty cycle model, and thus provide further understanding of the neighbor discovery in practical low-duty-cycle WSNs.

ACKNOWLEDGMENT

We would like to thank Dr. Yitong Yin for the helpful discussion. The work is supported in part by China NSF grants No. (60721002, 60825205, 61073152, 61003277), National Basic Research Program of China (973 Program) under grant No. 2009CB3020402, and the NSF of Jiangsu Province under grant No. BK2010102.

REFERENCES

[1] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X.-Y. Li, and G. Dai, Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest. In ACM SenSys, 2009.

[2] G. Tolle, J. Polastre, R. Szewczyk, N. Turner, K. Tu, S. Burgess, D. Gay, P. Buonadonna, W. Hong, T. Dawson, and D. Culler. A Macroscope in the Redwoods. In ACM SenSys, 2005.

[3] M. McGlynn and S. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In ACM MobiHoc, 2001.

[4] S. A. Borbash, A. Ephremides, and M. J. McGlynn. An asynchronous neighbor discovery algorithm for wireless sensor networks. Elsevier Ad Hoc Networks, 5(7):998-1016, 2007.

[5] R. Khalili, D. Goeckel, D. Towsley, and A. Swami. Neighbor discovery with reception status feedback to transmitters. In IEEE Infocom, 2010.

[6] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili. Neighbor discovery in wireless networks and the coupon collectors problem. In ACM MobiCom, 2009.

[7] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. In IEEE Transactions on Networking, Apr. 2004.

[8] T. Moscibroda, P. von Rickenbach, and R. Wattenhofer, Analyzing the energy-latency trade-off during the deployment of sensor networks, In Infocom, Apr. 2006.

[9] P. Erdös and A. Rényi, On a classical problem of probability theory, Magyar Tud. Akad. Mat. Kutato Int. Kozl, 1961.