Alano: An Efficient Neighbor Discovery Algorithm In An Energy-Restricted Large-Scale Network

Tong Shen, Yuexuan Wang, Zhaoquan Gu, Dongda Li, Zhen Cao, Heming Cui, and Francis C.M. Lau

***Abstract*—Neighbor discovery is a fundamental step in constructing wireless sensor networks and many algorithms have been proposed aiming to minimize its latency. Recent developments of intelligent devices call for new algorithms, which are subject to energy restrictions. In energy-restricted large-scale networks, a node has limited power supply and can only discover other nodes that are within its range. Additionally, the discovery process may fail if excessive communications take place in a wireless channel. These factors make neighbor discovery a very challenging task and only few of the proposed neighbor discovery algorithms can be applied to energy-restricted large-scale networks. In this paper, we propose Alano, a nearly optimal algorithm for a large-scale network, which uses the nodes' distribution as a key input. When nodes have the same energy constraint, we modify Alano by the Relaxed Difference Set (RDS), and present a Traversing Pointer (TP) based Alano when the nodes' energy constraints are different. We compare Alano with the state-of-the-art algorithms through extensive evaluations, and the results show that Alano achieves at least 31.35% lower discovery latency and has higher performance regarding quality (discovery rate) and scalability.**

I. Introduction

Wireless sensor networks are deployed in a wide range of real-life applications today, such as volcanic investigation [1], seismic detection [2], agriculture monitoring [3], etc. The popularity of Internet of Things (IoT) will accelerate this trend and make wireless sensor networks even more pervasive.

In constructing a wireless sensor network, neighbor discovery is an important fundamental step. In most situations, a node participating in a task needs to discover its nearest neighbors in order to carry out operations like broadcasting and peer-to-peer communication. In this paper, we study neighbor discovery under the constraint of limited energy. We assume a large-scale scenario, where nodes are aware of their energy consumption and are connected via a multi-hop network.

Despite extensive research, neighbor discovery in a large-scale network remains a problem that is not fully solved. Existing algorithms can be classified into two categories: deterministic, and probabilistic. The algorithms presented in Refs. [4]–[9] are based on deterministic sequences. Most deterministic algorithms however were designed for two nodes only, despite being then applied to a multi-node scenario. On the other hand, probabilistic algorithms handle neighbor discovery in a clique of n nodes [10]–[13], i.e. every two nodes are neighbors, and utilize the global number n to compute an optimal probability for action decisions. However, a large-scale network is generally not a clique and node to node communication could go through multiple hops. In addition, prominent existing algorithms such as Birthday [10] and Aloha-like [11] do not consider energy consumption of the neighbor discovery process, which could have a negative impact on the energy state of the sensors.

To address the problems identified above, we looked into the existing neighbor discovery algorithms and found that the key issue lies in the way collisions are dealt with in large-scale networks. There are three aspects to this issue. First, transmission signals fade with distance and simultaneous transmissions will cause collisions. Deterministic algorithms aiming at two nodes [5], [8] fail to reduce such collisions. Some beacon-based algorithms [4], [6], [7], [9] do not have this problem but their time slot is 40 times larger [5] and therefore resulting in long latency. Second, a large-scale network is not one-hop network and a node can practically only discover those neighbors that are within its range. Probabilistic algorithms [11]–[13] assuming a small clique network fail to estimate the number of neighbors, and thus cannot reduce the collisions effectively since the number of neighbors provides an important hint on how many collisions will occur at the same time. Third, nodes have limited energy and they only have a small time window to find their neighbors. Indeed, energy conservation and neighbor discovery are two conflicting goals in existing algorithms. ineffective.

We have conducted experiments to confirm the issue in real actions. We deployed 16 EZ240 sensors [14] and found existing algorithms to be either insufficient or excessive in the way they deal with collisions, and both would result in long latency. As the number of neighbors increased, collisions of Hedis [8] happened as frequently as 10.1% to 19.96%, evoking the CSMA [15] function in the MAC layer to wait for a random time. Hello [7] utilized a beacon mechanism to avoid collisions but the time slot was 40 times larger and it resulted in 10 times longer latency. An Aloha-like method [12] showed a high idle rate (when no neighbors are transmitting) of 18.92%, which reduced the collisions, but excessively. All these algorithms could not achieve low latency and energy conservation at the same time simply because they failed to deal with collisions effectively.

Our key hypothesis is that, using an estimate of the expected number of neighbors of a node and synchronizing the times the nodes turn on the radio, both low-latency and energy-efficiency in neighbor discovery can be achieved. We take the distribution of nodes into consideration. As studied in [16], nodes in a wireless sensor network are likely to follow a uniform or a Gaussian distribution. According to the local density, a node can estimate the number of neighbors and calculate an optimal probability for action decisions. Based on this, we propose Alano, [1] a nearly optimal probability based algorithm for a large-scale network. We play on the duty cycle mechanism [17] which is the fraction of time the radio is on (i.e., the sensor is woken up), and deterministically synchronize the wake-up times between neighbors in Alano. Specifically, if all nodes have the same (symmetric) duty cycle, such as a fleet of sensors having a default duty cycle setting, we propose the Relaxed Difference Set based algorithm (called RDS-Alano); if nodes have different (asymmetric) duty cycles, such as when sensors would adjust the duty cycle according to the remaining energy, we propose the Traversing Pointer based algorithm (called TP-Alano).

In the simulations, we have found that Alano achieves 31.35% to 85.25% lower latency, higher discovery rate, and better scalability in large scale networks. In comparison with the state-of-the-art algorithms [6]–[8], [12] Alano reaches nearly 100% discovery in twice as fast a speed. When the number of nodes increases from 1000 to 9000, Alano shows 4.68 times to 6.51 times lower latency for neighbor discovery.

The contributions of the paper are summarized as follows:

1) We utilize the distribution of nodes and propose Alano, a nearly optimal algorithm that can achieve low-latency neighbor discovery for a large-scale network.
2) In an energy-restricted large-scale network, we propose RDS-Alano for symmetric nodes and TP-Alano for asymmetric nodes. Both algorithms achieve low latency for discovering neighbors and can prolong the nodes' lifetime.
3) We conduct experiments for and extensive simulations for large-scale networks, in which Alano achieves lower latency, higher discovery rate, and better scalability. The results show Alano's promises for deployment in IoT in the future.

The remainder of the paper is organized as follows. The next section highlights the related works and their unsolved problems. Section III presents the system model and basic definitions. We introduce Alano and show the method to combine the nodes' distribution in the design in Section IV. We propose two modified algorithms (RDS-Alano, TP-Alano) for an energy-restricted large-scale network for both symmetric and asymmetric nodes in Section V. The simulation results are given and discussed in Section VI, and we conclude the paper in Section VII.

II. Related Work

Existing neighbor discovery algorithms can be classified into two categories: deterministic and probabilistic.

Deterministic algorithms adopt certain mathematical tools to ensure discovery between every two neighbors. The first tool is the quorum system [18]: for any two intersected quorums, two neighboring nodes could choose any quorum in the system to design the discovery schedule. Hedis [8] is a typical one. Another important tool is co-primality where two co-prime numbers are chosen by the neighbors to design the discovery schedule, and they can discover each other within a bounded latency by the Chinese Remainder Theorem [19]. Some representative algorithms are Disco [4], U-Connect [5], and Todis [8]. These algorithms have the obvious advantage that they can guarantee fast discovery for two nodes within a bounded latency.

However, there are weaknesses in these deterministic algorithms when they are applied to large-scale networks. Hedis [8] only assumes that only when two nodes turn on the radio at the same time can they they then find each other. However, in reality, neighbors play asymmetric roles and one is transmitting while the other one is receiving. U-Connect [5] considers the transmitting and receiving roles. Nevertheless, different from the two-node scenario, collisions will happen when many nodes are transmitting simultaneously. Some deterministic algorithms propose a beacon-based transmission protocol. For example, Disco [4] assumes that a node has a capability to send a beacon (comprising one or a few bits) at both the beginning and the end of an active time slot, and the assumption is adopted in SearchLight [6], Hello [7], and Nihao [9]. Under this assumption, the listening time slot is much larger than the transmitting time slot and collisions can thus be avoided. Nevertheless, this mechanism makes a complete slot 40 times larger than that of U-Connect [5] and thus cannot promise a reasonable discovery latency.

Another category is probabilistic algorithms [10]–[13] which utilize probability techniques to inject randomness into the discovery process. Birthday protocol [10] is one of the earliest algorithms that works on the birthday paradox, i.e. the probability that two people have the same birthday exceeds $\frac{1}{2}$ among 23 people. Following that, smarter probabilistic algorithms are proposed, such as an Aloha-like algorithm [11], [12], PND [13]. Particularly, the Aloha-like algorithm [11] does not consider energy consumption, which was later extended to an energy-restricted network by [12].

Unlike deterministic algorithms, probabilistic algorithms show significant strength for a network consisting of multiple nodes. However, probabilistic algorithms can only give an expectation for the discovery latency and cannot guarantee a good latency bound. In addition, most of the existing algorithms assume the network is a clique, which implies any two nodes are neighbors. This assumption can hardly match the situation of a large-scale network. Some work uses the received signal strength to decide how far a node can transmit [20], and in the protocol model, this is simplified to be that two nodes can communicate if their distance is no larger than a threshold.

III. Preliminaries

In this section, we first describe the system model of an energy-restricted large-scale network. Then we give a formulation of the Neighbor Discovery problem. The notations are listed in Table I.

[1]Alano is the god of luck in Greek mythology.

TABLE I
NOTATIONS FOR NEIGHBOR DISCOVERY

Notation	Description
N	The number of nodes in the network
u_i	Node u_i with ID i
$\widetilde{n_i}$	Node u_i's expected number of neighbors
S_i	The set of u_i's neighbors
t_0	The length of a time slot
δ_{ij}	The transmission time drift between u_i and u_j
t_i^s	The start time of node u_i
$L(i,j)$	The discovery latency that node u_i discovers node u_j
$L(i)$	The discovery latency that node u_i discovers all neighbors
θ	The pre-defined global duty cycle
θ_i	Node u_i's local duty cycle
W	The time slot spent by a node discovering all neighbors
M	Neighboring matrix, $M_{ij} = 1$ means u_i and u_j are neighbors

A. System Model

We introduce three important aspects in an energy-restricted large-scale network.

Communication: When multiple nodes communicate simultaneously, transmission may fail due to interference. We adopt the protocol model (also called the graph model or unit-disk model [21], [22]) to describe the process, which assumes a node u_i can receive u_j's message successfully if u_j is the only transmitter that is within u_i's communication range. The protocol model is a popular one that enables the development of efficient algorithms for crucial networking problems. Some other models, such as the signal to interference plus noise ratio (*SINR*) model, are more complicated and lack good algorithmic features. In addition, it is shown that these models can be transformed to the protocol model by particular means in [23].

Network connectivity: A node can only discover nodes that are within its radio range. In a large-scale network, two nodes may not be connected directly and we call two nodes that are connected by one-hop communication as *neighbors*.

Energy-restricted: A node in the network has limited energy, and can turn on or off its radio in order to save energy [17], [24]. When a node turns on the radio, it can transmit a message including its identifier (ID), or listen on the channel to receive a neighbor's message.

Technically speaking, we assume an energy-restricted large-scale network to be consisting of N nodes in a set, $U = \{u_1, u_2, \ldots, u_N\}$. Nodes are distributed in a large area and they communicate through a fixed wireless channel. We assume the locations of the nodes obey some distribution, such as the uniform distribution or the Gaussian distribution [16]. Suppose a node has a fixed radio range Δ and two nodes u_i, u_j are neighbors if their distance suits $d(u_i, u_j) \leq \Delta$.

Suppose time is divided into slots of equal length t_0 [25], which is assumed to be sufficient to finish a complete communication process (one node transmits a message including its ID and a neighbor receives the message). A node that has its radio on can choose to be in the transmitting state or the listening state:

- **Transmitting state:** a node transmits (broadcasts) a package containing its ID on the channel;
- **Listening state:** a node listens on the channel to receive messages from neighbors.

In the protocol model, a node u_i can discover its neighbor u_j in time slot t if and only if u_j is the only neighbor of u_i that transmits and u_i is listening in the slot.

A node has limited energy and it has to turn off the radio to save energy for most of the time. We assume a *duty schedule* for a node u_i which is a pre-define sequence $S_i = \{s_i(t)\}_{0 \leq t < T}$ of period T, in which

$$s_i^t = \begin{cases} S & u_i \; turns \; off \; the \; radio \; in \; slot \; t \\ T & u_i \; is \; in \; transmitting \; state \; in \; slot \; t \\ L & u_i \; is \; in \; listening \; state \; in \; slot \; t \end{cases}$$

duty cycle is defined as the fraction of time a node turns its radio on, which is formulated as:

$$\theta_i = \frac{|\{t : 0 \leq t < T, s_i(t) \in \{T, L\}\}|}{T}.$$

If all nodes have the same duty cycle all the time, i.e. $\theta_i = \theta_j$ for any nodes u_i, u_j, we call them *symmetric nodes*. Otherwise, they are *asymmetric nodes*. Denote the start time of node u_i as t_i^s and denote δ_{ij} as the time drift between a pair of neighbors u_i, u_j, i.e. $\delta_{ij} = t_i^s - t_j^s$.

B. Problem Definition

A node u_i executes operations (sleep, transmitting, or listening) according to the pre-defined duty schedule S_i. When u_i starts the neighbor discovery process, denote $L(i,j)$ as the slot cost to find the neighbor u_j and we define **discovery latency** of node u_i as the time to discover all neighbors:

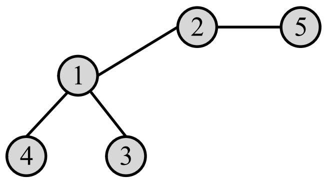
$$L(i) = \max_{j: M_{i,j}=1} L(i,j).$$

It is important to note that neighbor discovery is not bidirectional and any pair of neighbors have to discover each other separately. We formulate the neighbor discovery problem for node u_i as:

Problem 1: For a node u_i and the set of its neighbors $S_i = \{u_j | d(u_i, u_j) \leq \Delta\}$, design duty schedules for all nodes such that: $\forall u_j \in S_i$:

$$\exists t \; s.t. :$$
$$s_i(t) = L, s_j(t) = T, and \; \forall u_k \in S_i, u_k \neq j : s_k(t) \in \{L, S\}.$$

Fig. 1 shows an example of 5 nodes; the topology is depicted in Fig. 1(a), and Fig. 1(b) describes the neighbor discovery process. The duty cycle is set as 0.25 and nodes start in different time slots. The designed duty schedule for node u_1 is $S_1 = \{T, S, S, S, S, S, T, S, S, T, S, S, S, T, ...\}$. Suppose time drift between node u_1 and u_4 is $\delta_{14} = 1$; u_1 u_2 start in the same time slot. In slot 12, node u_5 discovers neighbor u_2, but node u_1 cannot discover u_2 since another neighbor u_3 is in transmitting state simultaneously.



(a) The topology of a simple wireless network

Time	…	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	…
Node 1			T	S	S	S	S	S	L	S	S	L	S	S	S	T	…
Node 2			S	S	L	S	S	S	T	S	L	S	S	S	S	L	…
Node 3	…	S	S	S	T	S	S	S	T	S	L	S	S	S	L	S	…
Node 4				S	S	S	S	S	S	S	S	S	T	T	T	T	…
Node 5							S	S	L	S	S	S	T	S	S	S	…

(b) Neighbor discovery process

Fig. 1. An example of neighbor discovering process. S, T and L represents Sleep pattern, Transmitting state and Listening state in wake-up pattern respectively.

IV. Alano Algorithm for a large-scale Network

In large-scale networks, nodes are not fully-connected and thus communications may fail due to concurrent transmissions. When not considering the energy constraints of nodes, we propose a nearly optimal algorithm for a large-scale network, which implies $\theta_i = 1$ for all node u_i. Supposing the locations of nodes obey some distribution, we propose the Alano algorithm and analyze its performance for two commonly used distribution (uniform distribution and Gaussian distribution).

A. Alano Algorithm

Suppose the locations of nodes obey some distribution and each node u_i is aware of its position (x_i, y_i). Then u_i could compute its local density by the following general function:

$$f(x,y) = \begin{cases} \varphi(x,y) & (x,y) \in D \\ 0 & (x,y) \notin D \end{cases}$$

where (x, y) is the position's coordinates, and D is the network covering area. Denote the range of u_i's neighbors' positions as R_i, and any neighbor with coordinates $(x, y) \in R_i$ suits:

$$(x - x_i)^2 + (y - y_i)^2 \le \Delta^2.$$

where Δ is the communication range. Then, node u_i's expected number of neighbors (denote as $\widetilde{n_i}$) is:

$$\widetilde{n_i} = N \iint_{R_i} f(x,y)\,dx\,dy - 1.$$

In a large-scale network, we ignore the boundary area of the network. Note that, when the network covering area D is much larger than the area R_i of node u_i, we have:

$$\widetilde{n_i} \simeq N\pi\Delta^2\varphi(x_i, y_i). \tag{1}$$

We present **Alano**, a randomized algorithm for node u_i in Alg. 1. By computing the expected number of nodes, u_i enters transmitting state or listening state according to the generated probability in Line 1.

Algorithm 1 Alano Algorithm

1: Set transmit probability $p_t^i := \frac{1}{\widetilde{n_i}+1}$, $t := 0$;
2: **while** $t \le T$ **do**
3: Generate a random number $\epsilon \in (0, 1)$;
4: **if** $\epsilon < p_t$ **then**
5: Transmit a message containing u_i's information including its ID through the channel;
6: **else**
7: Listen on the channel. If receive a message successfully, decode the message and record the sender's ID;
8: **end if**
9: $t := t + 1$;
10: **end while**

In the following, we first consider a general situation that nodes in the network are uniformly distributed. We derive a proof that the probability chosen in Alano is the optimal one and show that the discovery latency will not be much larger than its expectation. T in Alg. 1 Line 2 is the time threshold and can be set as the latency bound. Then we analyze a more common situation where nodes obey the Gaussian Distribution; we present an approximation analysis showing that the discovery latency is not much larger than that of the uniform distribution.

B. Analysis for Uniform Distribution

Uniform distribution is used in most deployments of wireless networks. For instance, to monitor an unknown area, many sensors are deployed uniformly to collect information, such as temperature and humidity [26]. The nodes are evenly deployed and the density function is:

$$f(x,y) = \begin{cases} \frac{1}{A} & (x,y) \in D \\ 0 & (x,y) \notin D \end{cases}$$

where A is the area of D. By Eqn. (1), u_i's expected number of neighbors is $\widetilde{n_i} = \frac{N\pi\Delta^2}{A}$ and the probability in Line 2 is set as $p_t^i = \frac{1}{\widetilde{n_i}+1} = \frac{A}{N\pi\Delta^2+A}$.

Lemma 1: Alg. 1 achieves optimal discovery latency for uniform distribution by setting $p_t^i = \frac{1}{\widetilde{n_i}+1}$.

Proof: For any two nodes u_i abd u_j in the uniform distribution, we get $\widetilde{n_i} = \widetilde{n_j} = \widetilde{n}$ and $p_t^j = p_t^i = p_t = \frac{1}{\widetilde{n}+1}$.

From Alg. 1, the probability that node u_i discovers a specific neighbor (such as u_j) successfully in a time slot (denote as p_s) is:

$$p_s = p_t(1 - p_t)^{\widetilde{n}-1}(1 - p_t).$$

In order to compute the maximum probability to discover a node, we compute the differential function of p_s as:

$$\frac{d(p_s)}{d(p_t)} = (1 - p_t)^{\widetilde{n}} - \widetilde{n}p_t(1 - p_t)^{\widetilde{n}-1}.$$

When $p_t = \frac{1}{\widetilde{n}+1}$, p_s gets the maximum value:

$$p_s = \frac{1}{\widetilde{n}+1}(1-\frac{1}{\widetilde{n}+1})^{\widetilde{n}} \approx \frac{1}{(\widetilde{n}+1)e}.$$

Therefore, the probability chosen in Alano algorithm for transmitting is optimal. ■

Theorem 1: Alg. 1 discovers all neighbors for node u_i within $T = \Theta(n \ln n)$ time slots with high probability.

Proof: Let the random variable X denote the time a node spends discovering all neighbors. Let X_j denote the cost of time slots for u_i to discover a new neighbor after $j-1$ neighbors have been discovered. Obviously, $X = \sum_{j=1}^{\widetilde{n}}$ and X_j follows Geometric distribution $G(p(j))$, where $p(j) = (\widetilde{n}-j+1)p_s$. We get

$$E[X_j] = \frac{1}{p_j}, Var[X_j] = \frac{1-p_j}{p_j^2}.$$

Therefore,

$$E[X] = \sum_{j=1}^{\widetilde{n}} = \frac{C_n}{p_s} \approx (\widetilde{n}+1)e(\ln(\widetilde{n}+1)+\Theta(1)) = \Theta(n \ln n). \tag{2}$$

where $C_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$. Also we get

$$Var[X] = \sum_{j=1}^{\widetilde{n}} Var[X_j] \leq \frac{\pi^2}{6p_s^2} - \frac{C_n}{p_s}.$$

By the *Chebyshev's inequality*, we get

$$P[X \geq 2E[X]] \leq \frac{Var[X]}{E[X]^2} \leq \frac{\pi^2}{6C_n^2} - \frac{p_s}{C_n}.$$

Therefore,

$$\lim_{n \to +\infty} P[X \geq 2E[X]] = 0. \tag{3}$$

According to equation (2) and (3), we get $X = O(n \ln n)$. Thus X is bounded by $O(n \ln n)$ with high probability. ■

C. Analysis of Gaussian Distribution

The Gaussian distribution is getting more commonly adopted in wireless networks, such as in an intrusion detection application which needs a larger detection probability around important entities [16]. We assume the positions of nodes obey a 2D Gaussian distribution, and we present a theoretical proof that the discovery latency is not much larger than that of the uniform distribution. Without loss of generality, we consider $(x, y) \sim N(0, 1, 0, 1, 0)$.

Theorem 2: Alg. 1 discovers all neighbors for node u_i within $T = \Theta(n \ln n)$ time slots with high probability.

Proof: Denote the approximate neighbors of node u_i as set $S_i = \{u_j | d(u_i, u_j) \leq \Delta\}$. When nodes obey the Gaussian distribution, the probability that node u_i discovers a certain neighbor node u_j successfully in a time slot (denote as p_s) can be formulated as:

$$p_s^i = (1-p_t^i) \cdot p_t^j \cdot \prod_{u_k \in S_i, u_k \neq u_j} (1-p_t^k).$$

Denote $p_t^{max} = \max_{u_j \in S_i}\{p_t^j\}, p_t^{min} = \min_{u_j \in S_i}\{p_t^j\}$, for every $u_j \in S_i$, we have:

$$(1-p_t^i)p_t^{min}(1-p_t^{max})^{\widetilde{n_i}-1} \leq p_s^i,$$
$$p_s^i \leq (1-p_t^i)p_t^{max}(1-p_t^{min})^{\widetilde{n_i}-1}.$$

Denote:

$$H = (1-p_t^i)p_t^{min}(1-p_t^{max})^{\widetilde{n_i}-1}.$$
$$Q = (1-p_t^i)p_t^{max}(1-p_t^{min})^{\widetilde{n_i}-1}.$$

We derive the expectation of X_j for $1 \leq j \leq n_i$ as:

$$\frac{1}{(\widetilde{n_i}-j+1)Q} \quad \leq \quad E[X_j] \quad \leq \quad \frac{1}{(\widetilde{n_i}-j+1)H}.$$

Combine the equations to derive:

$$\frac{1}{Q}C_n \quad \leq \quad E[\sum_{j=1}^{\widetilde{n_i}} W_j] \quad \leq \quad \frac{1}{H}C_n.$$

Since: $p_t^{min} = \frac{1}{\widetilde{n}_{max}+1}$, $p_t^{max} = \frac{1}{\widetilde{n}_{min}+1}$. where $\widetilde{n}_{max} = \max\{\widetilde{n}_j | u_j \in S_i\}$, $\widetilde{n}_{min} = \min\{\widetilde{n}_j | u_j \in S_i\}$.

And $\widetilde{n}_{max}$, $\widetilde{n}_{min}$ can be computed as follows:

$$\widetilde{n_i} \simeq N\pi\Delta^2 \frac{1}{2\pi} \mathrm{e}^{-\frac{x_i^2+y_i^2}{2}}.$$

$$\widetilde{n}_{max} \simeq N\pi\Delta^2 \frac{1}{2\pi} \mathrm{e}^{-\frac{(\sqrt{x_i^2+y_i^2}-\Delta)^2}{2}} = \alpha_{(x_i,y_i)}\widetilde{n_i}.$$

$$\widetilde{n}_{min} \simeq N\pi\Delta^2 \frac{1}{2\pi} \mathrm{e}^{-\frac{(\sqrt{x_i^2+y_i^2}+\Delta)^2}{2}} = \beta_{(x_i,y_i)}\widetilde{n_i}.$$

where $\alpha_{(x_i,y_i)} = \mathrm{e}^{\frac{2\Delta\sqrt{x_i^2+y_i^2}-\Delta^2}{2}}$, $\beta_{(x_i,y_i)} = \mathrm{e}^{-\frac{2\Delta\sqrt{x_i^2+y_i^2}+\Delta^2}{2}}$, both of which can be seen as constants. Then we get:

$$\beta_{(x_i,y_i)}\mathrm{e}^{\frac{1}{\alpha_{(x_i,y_i)}}} n \ln n \leq E[X] \leq \alpha_{(x_i,y_i)}\mathrm{e}^{\frac{1}{\beta_{(x_i,y_i)}}} n \ln n.$$

Therefore, we get the non-uniform bound:

$$E[X] = O(\psi(x_i, y_i) n \ln n). \tag{4}$$

where $\psi(x_i, y_i) = \alpha_{(x_i,y_i)}\mathrm{e}^{\frac{1}{\beta_{(x_i,y_i)}}}$. We derive the upper bound of $Var[X]$:

$$Var[X] = \sum_{j=1}^{\widetilde{n_i}} Var[X_j] = \sum_{j=1}^{\widetilde{n_i}} \frac{1-p_j}{p_j^2}$$
$$\leq \sum_{j=1}^{\widetilde{n_i}} \frac{1-(\widetilde{n_i}-j+1)H}{(\widetilde{n_i}-j+1)^2H^2} \leq \frac{\pi^2}{6H^2} - \frac{p_s}{H}.$$

By the *Chebyshev's inequality*, we get

$$P[X \geq 2E[X]] \leq \frac{Var[X]}{E[X]^2} \leq \frac{\pi^2\alpha^2\mathrm{e}^{\frac{2}{\beta}-\frac{2}{\alpha}}}{6\beta^2 \ln \widetilde{n_i}^2} - \frac{\alpha\mathrm{e}^{\frac{1}{\beta}}}{\beta^2\mathrm{e}^{\frac{2}{\alpha}}\widetilde{n_i}\ln\widetilde{n_i}}.$$

Therefore,

$$\lim_{n \to +\infty} P[X \geq 2E[X]] = 0. \tag{5}$$

According to equation (4) and (5), we get the non-uniform bound $X = O(\psi(x_i, y_i) n \ln n)$. Thus X is bounded by $O(\psi(x_i, y_i) n \ln n)$ with high probability. ■

V. Modified Alano for An Energy-Restricted Network

In an energy-restricted network, nodes have limited energy and designing duty schedule for a node needs to take its duty cycle into account. Obviously, a lower duty cycle implies a larger discovery latency since the node turns its radio off much of time during the schedule.

In the preceding section, energy constraint is not considered a crucial factor in the design of Alano. Here, we would modify Alano for both symmetric nodes and asymmetric nodes as determined by their duty cycles. Our key idea is to synchronize the slots such that the radio is on for neighboring nodes in a bounded time, and then invoke Alano algorithm to achieve low-latency neighbor discovery.

A. RDS-Alano for Symmetric Nodes

Symmetric nodes have the same duty cycle $\theta_i = \theta_j = \theta$, $\forall u_i, u_j$. We utilize Relaxed Difference Set (RDS) to synchronize time slots that nodes are in transmitting or listening state.

RDS is an efficient tool to construct cyclic quorum systems [18]. The definition is:

Definition 1: A set $R = \{a_1, a_2, ..., a_k\} \subseteq Z_M$ (the set of all non-negative integers less than M) is called a relaxed difference set (RDS) if for every $d \not\equiv 0$ (mod M) there exists at least one ordered pair (a_i, a_j), $a_i, a_j \in R$ such that $a_i - a_j \equiv d$ (mod M).

It has been proved that any RDS must have cardinality $|R| \geq \sqrt{N}$ [18]. We present a linear algorithm to construct a RDS with cardinality $\lceil \frac{3\sqrt{N}}{2} \rceil$ under Z_N in Alg. 2.

Algorithm 2 RDS construction under Z_N

1: $R := \emptyset$; $\lambda := \lceil \sqrt{N} \rceil$,$\mu := \lceil \frac{\lceil \sqrt{N} \rceil}{2} \rceil$;
2: **for** $i = 1 : \lambda$ **do**
3:     $R := R \cup i$;
4: **end for**
5: **for** $j = 1 : \mu$ **do**
6:     $R := R \cup (1 + j * \lambda)$;
7: **end for**

We show the correctness of the construction formally.

Lemma 2: Set $R = \{r_0, r_1, ..., r_{\lambda+\mu-1}\}$ constructed in Alg. 2 is a RDS, where $|R| = \lambda + \mu = \lceil \sqrt{N} \rceil + \lceil \frac{\lceil \sqrt{N} \rceil}{2} \rceil \approx \lceil \frac{3\sqrt{N}}{2} \rceil$.

Proof: Obviously, if there exists one ordered pair (a_i, a_j) satisfying $a_i - a_j \equiv d$ (mod N), an opposing pair (a_j, a_i) exists such that $a_j - a_i \equiv (N - d)$ (mod N). Thus we only need to find at least one ordered pair (a_i, a_j) for each $d \in [1, \lfloor N/2 \rfloor]$.

In the construction, λ in Line 1 is the smallest integer satisfying $\lambda^2 \geq N$. Every d within range $[1, \lfloor N/2 \rfloor]$ can be represented as: $d = 1 + j \times \lambda - i$, where $1 \leq j \leq \mu, 1 \leq i \leq \lambda$. Thus, there exists $a_j = 1 + j \times \lambda$ from Line. 3 and $a_i = i$ from Line. 6 satisfying $a_j - a_i \equiv d$. Then, the lemma can be derived. ■

For symmetric nodes with duty cycle θ, we present a RDS based Alano (RDS-Alano) algorithm as Alg. 3.

In Alg. 3, RDS is used to construct a deterministic schedule for a node to turn on its radio in every period of length T, and Alano is utilized as a probabilistic strategy to determine whether it is in transmitting state or listening state.

Algorithm 3 RDS Based Alano Algorithm

$T := \lceil \frac{9}{4\theta^2} \rceil$; $t := 0$;
Invoke Alg. 2 to construct the RDS $R = r_0, r_1, ..., r_{\lceil \frac{3\sqrt{T}}{2} \rceil}$ under Z_T;
while $True$ **do**
    if $(t + 1) \in R$ **then**
        Invoke Alg. 1 to determine transmission state;
    else
        Sleep;
    end if
    t := (t + 1) % T;
end while

We derive discovery latency bound for RDS-Alano:

Theorem 3: RDS-Alano guarantees that the discovery latency of a node is bounded within $O(\frac{n \log n}{\theta^2})$ with high probability.

Proof: First, we verity that the duty cycle in RDS-Alano (denote as $\widetilde{\theta}$) is

$$\widetilde{\theta} = \frac{|RDS|}{|T|} = \frac{\lceil \frac{3\sqrt{T}}{2} \rceil}{T} = \theta.$$

For any pair of neighbor nodes (u_i, u_j), we can find an ordered pair (r_i, r_j) from their respective RDS such that $r_i - r_j \equiv \delta_{ij}$ (mode T), where δ_{ij} is the time drift. This implies any neighbor nodes can turn on their radios in the same time slot for at least once during every period of length T. Considering each period of T time slots to be a 'super' slot of Alano algorithm, we can derive that the discovery latency is bounded within $O(\frac{n \ln n}{\theta^2})$ slots with high probability by combining the analysis of Alano. ■

Remark 1: In a RDS, a node can discovery its neighbors in different time slots. When treating a period of T as a 'super' slot of Alano, there may be less than the total neighbors in each wake-up sub-slot, resulting less collisions and lower latency compared to when all the neighbors wake up in the same sub-slot. Thus the latency bound can not be larger.

B. TP-Alano for Asymmetric Nodes

Considering a more practical network where nodes can adjust their duty cycles, we present a traversing pointer method to synchronize time slots that nodes are in transmitting or listening state for asymmetric nodes.

For a more practical scenario, the nodes in a wireless sensor networks for instance, are assigned to diverse tasks such as temperature measurement, sunshine collection, etc., and thus ought to have asymmetric capability of battery-management with local duty cycle θ_i.

Algorithm 4 Traversing Pointer Based Alano Algorithm

1: $T :=$ the smallest prime no less than $\frac{2}{\theta_i}$; $t := 0$;
2: **while** $True$ **do**
3: $\quad t_1 := t\%T$;
4: $\quad t_2 := \lfloor t/T \rfloor \%(T-1)+1$;
5: $\quad$ **if** $t_1 = 0$ *or* $t_1 = t_2$ **then**
6: $\quad\quad$ Invoke Alg. 1 to determine transmission state;
7: $\quad$ **else**
8: $\quad\quad$ Sleep;
9: $\quad$ **end if**
10: $\quad t := t+1$;
11: **end while**

Suppose the duty cycle of node u_i is θ_i, we present a traversing pointer based Alano (TP-Alano) algorithm as Alg. 4. In each period of T slots, a node turns on its radio in two different time slots, one of which is the first slot of each period and the other one is a traversing slot that changes from period to period (as described in Line. 4).

We call the first time slot of each period as *fixed pointer* and the traversing slot as *traversing point*. The pointers are designed to guarantee that nodes u_i, u_j could turn on their radio simultaneously in very period of length T_iT_j. A sketch of the pointers is described in Fig. 2.

Note that, a period of T slots is constructed as Line 1 where we try to *find the smallest prime* $\geq \frac{2}{\theta_i}$, then it is likely to make the duty cycle of each period smaller than the expected one. This can be easily solved by selecting some random slots to turn on the radio for listening in each period T, to conform to the expected duty cycle.

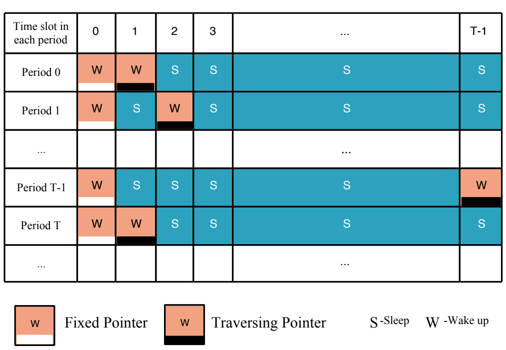


Fig. 2. A sketch of TP construction in Alg. 4

We show the discovery latency of TP-Alano algorithm as:

Theorem 4: TP-Alano guarantees that the discovery latency $L(i,j)$ is bounded within $O(\frac{nlogn}{\theta_i\theta_j})$ with high probability., where θ_i and θ_j are the duty cycles of a pair of neighbors (u_i, u_j) respectively.

Proof: We first prove that any pair of nodes (u_i, u_j) turn on their radios (for transmitting or listening) simultaneously for at least once in every period of length T_iT_j.

Case 1: $T_i \neq T_j$. Since T_i and T_j are different primes, according to Chinese Remainder Theorem [19], there exists a time slot $t_\tau \in [0, T_iT_j)$ satisfying:

$$0 = t_\tau \quad mod \quad T_i. \tag{6}$$

$$\delta_{ij} = t_\tau \quad mod \quad T_j. \tag{7}$$

Thus, there exists a fixed pointer of node u_i and a fixed pointer of node u_j in which both nodes turn on the radios in every period of length T_iT_j .

Case 2: $T_i = T_j$. Since $T_i = T_j = T$, if the time drift between u_i and u_j is $\delta_{ij} = 0$, the fixed pointers of u_i and u_j will be the same in every period of length T. Otherwise, since the traversing point will traverse all the time slots once during period of length $(T-1)T$, there exists a traversing point of u_i and a fixed pointer of u_j satisfying that both nodes turn on the radios simultaneously in every period of length $(T-1)T$; similarly a traversing point of u_j and a fixed pointer of i satisfying that they both turn on the radios.

Thus for any pair of neighbor nodes (u_i, u_j), they turn on their radios for transmitting or listening for at least once in every period of length T_iT_j. Considering the whole period T_iT_j to be a 'super' slot of Alano, we derive that the discovery latency is bounded within $O(\frac{n \ln n}{\theta_i\theta_j})$ with high probability. ∎

VI. Evaluation

We conduct experiments to verify our analytical results. and evaluate the algorithms in simulated large-scale networks.

A. Experiments for Fundamental Observation

In our experiments, we used 16 EZ240 sensors [14], each of which has an 8MHz MSP430F1611 microprocessor, 10K RAM, 48K ROM and a 1M flash. CC2420 is used as the communication module that runs the IEEE 802.15.4 protocol. RTIMER_ARCH_CECOND is the clock frequency and a time slot is set as 500/RTIMER_ARCH_CECOND, which is around 0.5ms in the real world. As discussed in section II, the time slot of beacon-based algorithms, e.g. searchlight [6], is set as 20ms. From the code in the MAC layer we can see that the CSMA mechanism makes the process to wait for a random time when a collision happens, which implies that collisions would lead to long latency.

TABLE II
Experimental results of discovery latency. Alano has 30.81% to 87.31% lower latency on average.

Algorithms	Alano	Hello	Searchlight	ALOHA
Average latency	1.278	1.847	8.58	10.07
Maximum latency	1.52	3.11	10.11	16.13
Minimum latency	1.10	0.73	6.31	5.11

We compared Alano with Searchlight [6], Hello [7] and an Aloha-like algorithm [12] in the experiments. Searchlight and Hello are two deterministic algorithms and Aloha-like is a typical probabilistic algorithm.

The result in the Table II shows that Alano outperforms the other algorithms. Hello shows low latency which results from its ideal latency bound for two nodes, and its weakness in handling collisions is not obvious in a small-scale network.

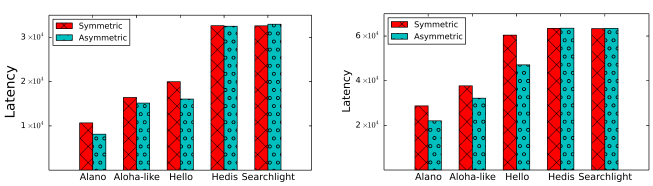
Searchlight is a typical beacon-based algorithm with larger time slots, which results in long latency. Aloha-like is designed for a clique network and the probability adopted is not optimal.

B. Simulations for energy-restricted Large-scale Networks

To simulate a large-scale network, we implemented Alano in C++ and evaluated the algorithms in a cluster of 9 servers, each equipped with an Intel Xeon 2.6GHz CPU with 24 hyper-threading cores, 64GB memory and 1T SSD.

We simulated a network that follows the uniform distribution and the Gaussian distribution respectively. For the uniform distribution, we suppose 500 nodes are distributed in an area of $100m \times 100m$ and each node's communication range is $\Delta = 10m$. For the Gaussian distribution, we suppose 1000 nodes are distributed in the same area, but each node's communication range is $\Delta = 5m$. We set the Gaussian distribution to $N(50, 15^2)$ in our evaluation. We set the duty cycle of a node to be 0.1 for symmetric nodes; for asymmetric nodes, we set their duty cycle randomly from 0.05 to 0.15 with step 0.02. These settings make the network more complicated but more realistic than those in [5]–[8], [10]–[13], [27].

We evaluated Alano, Aloha-like [12], Hello [7], Hedis [8], and Searchlight [6] in the generated networks. Since the neighbor discovery process will be stopped by CSMA after several failures, we set a latency threshold. Hello and Searchlight have a beacon transmission of $0.54ms$ at the beginning and the end of each slot, and a beacon makes up about $1/40$ of a slot; so we divide each $\{ON\}$ slot into 40 mini-slots, and the node transmits at the first and last mini-slot, and listens in other mini-slots. In the following, we show that Alano has lower latency, higher discovery rate, and better scalability.

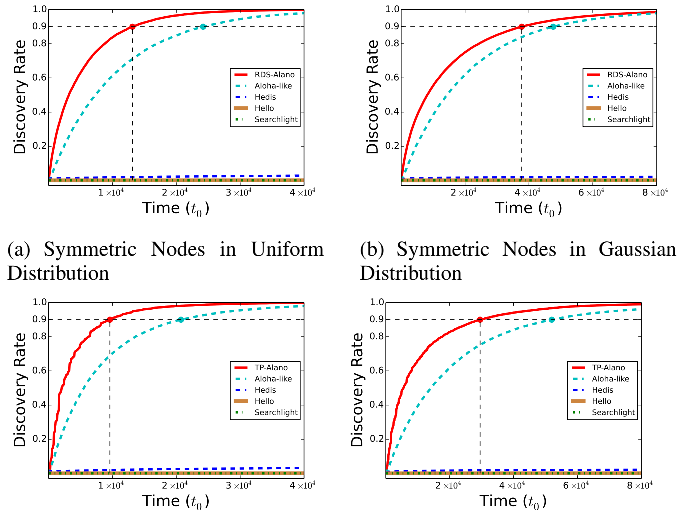


(a) Uniform Distribution (b) Gaussian Distribution

Fig. 3. Alano achieves 31.35% to 2.91 times lower latency.

1) Speed-Discovery Latency: When nodes follow the uniform distribution, we show the discovery latency comparison for both symmetric nodes and asymmetric nodes in Fig. 3(a). From the figure, Alano achieves 54.64% to 1.95 times lower discovery latency for symmetric nodes, and 85.25% to 2.91 times lower discovery latency for asymmetric nodes. When nodes follow the Gaussian distribution, as depicted in Fig. 3(b), Alano achieves 31.35% to 1.21 times lower discovery latency for symmetric nodes, and 45.94% to 1.88 times lower discovery latency for asymmetric nodes. The deterministic algorithms, Hello, Hedis and Searchlight, have high latency due to either collisions or larger time slots.

2) Quality-Discovery Rate: We use the discovery rate to evaluate Alano's quality. The discovery rate of a node u_i is defined as the percentage of discovered neighbors over u_i's



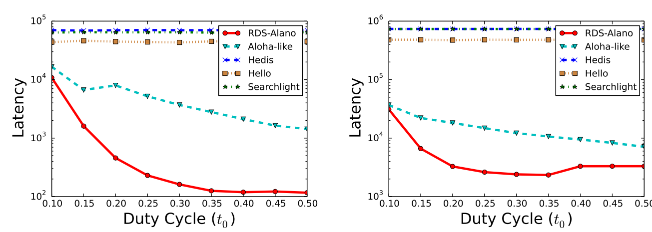
(a) Symmetric Nodes in Uniform Distribution (b) Symmetric Nodes in Gaussian Distribution

(c) Asymmetric Nodes in Uniform Distribution (d) Asymmetric Nodes in Gaussian Distribution

Fig. 4. Alano has higher discovery rate during the whole process for both uniform and Gaussian distributions.

all neighbors. In Fig. 4, we increase the number of nodes from 500 to 2000 for the uniform distribution, and from 1000 to 3000 for the Gaussian distribution; the results show Alano has higher discovery rate during the whole process for both the uniform and Gaussian distributions. In the uniform distribution, Alano achieves 90% discovery twice as fast as Aloha-like. It is remarkable that the performance of Aloha-like is close to Alano for the Gaussian distribution. This is because a number of nodes gather around the center area making the network similar to a clique. Thus Aloha-like shows its strength.

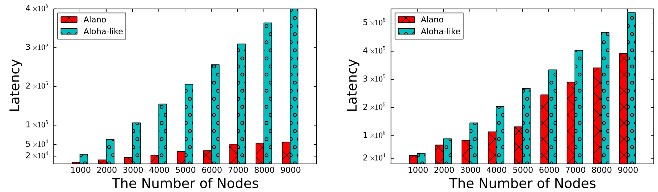
3) Scalability-Duty Cycle and Network Density: We evaluated scalability regarding duty cycle and network density.



(a) Uniform Distribution (b) Gaussian Distribution

Fig. 5. Alano achieves 53.66% to 11.23 times lower latency in different duty cycle.

Duty Cycle. When symmetric nodes have different duty cycles, Fig. 5 shows that Alano has lower latency. Compared with Aloha-like, Alano has from 53.66% to 11.23 times lower latency. The latency of Alano and Aloha-like generally decreases as the duty cycle increases, while Hello, Hedis and Searchlight have high latency due to collisions. In the Gaussian distribution, Alano has a small twist with duty cycle 0.35, because when the duty cycle increases, nodes are more likely to transmit and therefore collide.



(a) Uniform Distribution (b) Gaussian Distribution

Fig. 6. Alano achieves 25.23% times to 6.51 times lower latency when the network becomes denser.

Network Density. When the number of nodes increases, the network becomes denser. We choose the Aloha-like algorithm for comparison because Hello, Hedis and Searchlight already have higher latency than Aloha-like when there are 500 nodes in uniform distribution and 1000 nodes in Gaussian distribution. As shown in Fig. 6(a), Alano achieves 4.68 times to 6.51 times lower discovery latency than the Aloha-like algorithm for the uniform distribution, when the number of nodes increases from 1000 to 9000. When the number of nodes increases from 1000 to 9000 for the Gaussian distribution, Alano achieves 25.23% to 1.03 times lower discovery latency as shown in Fig. 6(b).

VII. Conclusion

In this paper, we systematically study the process of neighbor discovery in an energy-restricted large-scale network. To begin with, we propose Alano for a large-scale network where the nodes' distribution is utilized to decide a node's transmitting probability. For different distributions, such as the uniform distribution and the normal distribution, we show that Alano achieves nearly optimal discovery latency. Then, we propose two modified methods for an energy-restricted network on the basis of different duty cycle mechanisms: Relaxed Different Set based Alano (RDS-Alano) for symmetric nodes and Traversing Pointer based Alano (TP-Alano) for asymmetric nodes. We conduct extensive simulations to compare Alano with the state-of-the-art algorithms, and the results show that Alano achieves better performance regarding discovery latency, discovery rate, and scalability. In our future work, we will study the diversity of the nodes' distribution and the characteristics of a dynamic network, to achieve a more practical and efficient deployment in IoT.

References

[1] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *IEEE internet computing*, 2006.

[2] M. Suzuki, S. Saruwatari, N. Kurata, and H. Morikawa, "A high-density earthquake monitoring system using wireless sensor networks," in *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, 2007.

[3] Y. Wang, Y. Wang, X. Qi, L. Xu, J. Chen, and G. Wang, "L3sn: A level-based, large-scale, longevous sensor network system for agriculture information monitoring," *Wireless Sensor Network*, 2010.

[4] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008.

[5] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proceedings of the 9th ACM/IEEE international conference on information processing in sensor networks*. ACM, 2010.

[6] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: Won't you be my neighbor?" in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012.

[7] W. Sun, Z. Yang, K. Wang, and Y. Liu, "Hello: A generic flexible protocol for neighbor discovery," in *INFOCOM, 2014 Proceedings IEEE*.

[8] L. Chen, R. Fan, K. Bian, M. Gerla, T. Wang, and X. Li, "On heterogeneous neighbor discovery in wireless sensor networks," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*.

[9] Y. Qiu, S. Li, X. Xu, and Z. Li, "Talk more listen less: Energy-efficient neighbor discovery in wireless sensor networks," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*.

[10] 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*, 2001.

[11] S. Vasudevan, D. Towsley, D. Goeckel, and R. 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.

[12] L. You, Z. Yuan, P. Yang, and G. Chen, "Aloha-like neighbor discovery in low-duty-cycle wireless sensor networks," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*.

[13] T. Song, H. Park, and S. Pack, "A probabilistic neighbor discovery algorithm in wireless ad hoc networks," in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th*.

[14] T. Huang, H. Chen, Z. Zhang, and L. Cui, "Easipled: Discriminating the causes of packet losses and errors in indoor wsns," in *Global Communications Conference (GLOBECOM), 2012 IEEE*.

[15] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans," in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*.

[16] Y. Wang, W. Fu, and D. P. Agrawal, "Gaussian versus uniform distribution for intrusion detection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed systems*, 2013.

[17] W. Zhang, Y. Zhou, M. A. Suresh, and R. Stoleru, "Performance analysis and tuning of coexisting duty cycling wifi and wireless sensor networks," in *Sensing, Communication, and Networking (SECON), 2017 14th Annual IEEE International Conference on*.

[18] W.-S. Luk and T.-T. Wong, "Two new quorum based algorithms for distributed mutual exclusion," in *Distributed Computing Systems, 1997., Proceedings of the 17th International Conference on*. IEEE.

[19] C. Ding, D. Pei, and A. Salomaa, *Chinese remainder theorem: applications in computing, coding, cryptography*. World Scientific, 1996.

[20] V. Daiya, J. Ebenezer, S. S. Murty, and B. Raj, "Experimental analysis of rssi for distance and position estimation," in *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*. IEEE.

[21] T. Moscibroda and R. Wattenhofer, "The complexity of connectivity in wireless networks." in *Infocom*, 2006.

[22] L. Wang, P.-J. Wan, and W. Washington, "Connectivity of multihop wireless networks with log-normal shadowing," *Wireless Networks*, 2015.

[23] M. M. Halldórsson and T. Tonoyan, "How well can graphs represent wireless interference?" in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 2015.

[24] A. Dunkels, "The contikimac radio duty cycling protocol," 2011.

[25] L. F. Van Hoesel and P. J. Havinga, "A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches," 2004.

[26] A. Flammini, D. Marioli, E. Sisinni, and A. Taroni, "A real-time wireless sensor network for temperature monitoring," in *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*.

[27] K. Wang, X. Mao, and Y. Liu, "Blinddate: A neighbor discovery protocol," *IEEE Transactions on Parallel and Distributed Systems*, 2015.