Alano: May You Discover Your Neighbors In An Energy-Restricted Large-Scale Network

Abstract—**Neighbor discovery is a fundamental step in constructing wireless sensoir networks and many elegant algorithms have been proposed to minimize discovery latency. However, few of them can be applied to an energy-restricted large-scale network, which is more appealing and promising due to the development of intelligent devices. In an energy-restricted large-scale network, a node has limited power supply and it can only discover a part of nodes that are within distance range; in addition, the discovery process may fail if much communication exists on the wireless channel. These factors make neighbor discovery a challenging task in establishing the networks.**

In this paper, we propose Alano, a nearly optimal algorithm for a large-scale network on the basis of nodes' distributions. When nodes have same energy constraints, we modify Alano by Relaxed Difference Set (RDS) (denote as RDS-Alano); while we present a Traversing Pointer (TP) based Alano (denote as TP-Alano) when the energy constraints are different. We compare Alano with the state-of-the-art protocols through extensive evaluations, and the results show that Alano achieves at least 31.35% lower discovery latency and it has higher performance regarding quality (discovery rate), scalability and robustness.

I. Introduction

The popularity of Internet of Things has turned people's attention back to wireless sensor networks [1], with a wide range of applications such as volcanic investigation [2], seismic detection [3], agriculture monitoring [4], etc.

Neighbor discovery, in which a node tends to discover its nearby neighboring nodes before further applications like broadcasting and peer-to-peer communication, is a fundamental step to construct a wireless sensor network. In this paper, we study it in an energy-restricted large-scale scenario, where nodes are aware of energy consumption and multi-hop connected.

Unfortunately, despite great efforts, neighbor discovery in a large-scale network still remains an open problem. The existing algorithms can be classified into two categories: deterministic, and probabilistic. In the deterministic algorithms [5]–[11], sensors take actions based on deterministic sequences. Nevertheless, most deterministic algorithms are designed only for two nodes and directly applied to multi-node scenario. Probabilistic algorithms handle neighbor discovery in a clique of n nodes [12]–[14] , i.e. every two nodes are neighbors, and utilize the global number n to compute an optimal probability for action decisions. However in a large-scale network, a node is only able to sense nodes within a distance range. In addition, some existing algorithms [12], [13] do not consider energy consumption of the neighbor discovery process. In wireless sensor networks, sensors have very limited energy and have to be recharged frequently, attempting for neighbor discovery consistently is energy-consuming.

Therefore, we look into the existing neighbor discovery algorithms and find that the key issue lies in collisions in the large-scale networks. This issue owns to three reasons. First, transmission signals fade with distance and simultaneous transmissions would incur collisions among various nodes. Deterministic algorithms aiming at two nodes all fail to reduce such collisions. Second, a large-scale network is not one-hop connected and a node can only discover its neighbors within a distant range. Probabilistic algorithms assuming a samll clique network fail in estimating the number of neighbors, and thus can not reduce the collision effectively since the number of neighbors plays a vital role in how many collisions will occur at the same time. Third, nodes are powered by limited energy and they typically try to find neighbors only for a fraction of time. Energy consumption and neighbor discovery quality are a paradox in existing algorithms, since collisions cause great energy consumption and if taken energy consumption into account, neighbor discovery may become even ineffective.

We conduct experiments on the seven existing algorithms [5]–[11] and confirm the collision is the key issue. By experiment, we find collisions caused by simultaneous transmissions result in the waste of time and energy. Collisions of searchlight [7], the optimal deterministic algorithm, happen as frequently as 99.8% of the time; collisions of PND [15], a typical probabilistic algorithm, happen as frequently as 43.4% of the time. Aloha-like [14], the optimal probabilistic algorithm, shows a high idle rate (none of the neighbors are transmitting) as 94.6% of the time, which reduces the collision excessively and still results in a high latency. This is the same with CSMA [16], a general collision avoidance technique in networks, the idle rate of which is XX% by simulations. That means, because of collisions, existing works waste time and energy, and cannot achieve low latency and energy efficiency for large-scale networks.

Our insight idea is that, by estimating the expected number of neighbors of a node and aligning the time it turns on the radio with its neighbors, we can achieve both low-latency and energy-efficiency for neighbor discovery. We first take the distributions of nodes into consideration. As studied in [17], nodes in a wireless sensor network are likely to follow a uniform or a Gaussian distribution for detecting aims. According to the local density, a node can estimate the number of neighbors and calculate an optimal probability for action decisions. Then based on this, we propose Alano[1], the first nearly optimal probability based algorithm for a large-scale network. Finally, we involve the duty cycle mechanism, i.e. the fraction of time the radio is on (also called the sensor wake up), and modify Alano by deterministic techniques to align the wake-up time between neighbors. Specifically, if all nodes have the same (symmetric) duty cycle, such as a batch of sensors have a default duty cycle setting, we propose a Relaxed Difference Set based algorithm (called RDS-Alano); if nodes have different (asymmetric) duty cycles, such as a sensor adjusts the duty cycle by the remaining energy, we propose a Traversing Pointer based algorithm (called TP-Alano).

Our simulation shows the proposed Alano has lower latency, higher discovery rate, better scalability, and robustness, compared to the state-of-the-art algorithm [7]–[9], [14] in a large-scale network. Alano achieves 31.35% to 85.25% lower latency and reaches nearly 100% discovery rate within half time. When the number of nodes increases from 1000 to 9000, Alano shows 4.68 times to 6.51 times lower latency for neighbor discovery. In addition, Alano still keeps low latency robustly when 5% to 30% nodes are off duty.

The contributions of the paper are summarized as follows:

1) We utilize nodes' distribution and propose Alano, a nearly optimal algorithm that achieves 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 node's lifetime;
3) We conduct experiments for fundamental observation and extensive simulations for large-scale networks. Alano achieves lower latency, higher discovery rate, better scalability, and robustness compared with the state-of-the-art algorithms.

The remainder of the paper is organized as follows. The coming section highlights some related works and puts forward vital problems that the existing algorithms remain. The system model and basic definitions are introduced in Section III. We present Alano and show the method to combine nodes' distribution in Section IV. We propose two modified algorithms (RDS-Alano, TP-Alano) for a energy-restricted large-scale network for both symmetric and asymmetric nodes in Section V. The extensive simulation results are shown in Section VI and we conclude the paper in Section VII.

II. Related Work

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

Deterministic algorithms adopt some mathematic tools to ensure discovery between every two neighbors. The first tool is called quorum system [18], [19] : for any two intersected quorums, two neighboring nodes could choose any quorum in the system to design the discovery schedule. Hedis [9] 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 [20]. Some representative algorithms are Disco [5], U-Connect [6], and Todis [9]. These algorithms hold an obvious advantage that they can guarantee fast discovery for two nodes within a bounded latency.

However, there exists some weak points in the deterministic algorithms when applying to large-scale networks. U-Connect [6] only assumes two nodes turn on the radio at the same time and thus find each other. But in reality neighbor discovery is the process that a node receives a handshaking package from its neighbor successfully. Hedis and Todis [9] consider the transmitting and receiving roles. However different from two-node scenario, collisions will happen when many nodes are transmitting simultaneously. Some deterministic algorithms propose a different transmission protocol. For example, Disco [5] assumes that a node has a capability to send a beacon (one or a few bits) at both beginning and end of an active time slot, and the assumption is adopted in SearchLight [7], Hello [8], and Nihao [11]. Under the assumption, these algorithms could guarantee discovery between two nodes within an ideal latency. But sending a beacon may not work well in a large scale network since a node need to distinguish its neighbors by the received messages, which implies a transmitted package should contain the sender node's information [21].

Another category is probabilistic algorithms [12]–[15] which utilize probability techniques to promote the randomness of discovering the neighbors. Birthday protocol [12] 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 Aloha-like [13], [14], PND [15]. Particularly, Aloha-like [13] does not consider the energy consumption, which is later extended to an energy-restricted network by [14].

Different from deterministic algorithms, probabilistic algorithms show an significant strength for a network consisting of multiple nodes. However, probabilistic algorithms only present an expectation discovery latency and they can not 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 depict a large-scale network due to the limited communication range of a node. Some works adopt the received signal strength to decide how far a node can transmit [22], and in the protocol model, it is simplified that two nodes can communicate if their distance is no larger than a threshold.

III. Preliminaries

In this section, we describe system model of an energy-restricted large-scale network, and we formulate the Neighbor Discovery problem formally. 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
u_{ij}	the j^{th} neighbor of node u_i
δ_{ij}	The transmission time drift between u_i and u_j
$n*$	The average number of neighbors is n, $n = p_n N$
t_0	The length of a time slot
$L(i,j)$	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 factors of an energy-restricted large-scale network.

Communication: when multiple nodes communicate simultaneously, a transmission may fail due to the communication interference. We adopt the protocol model (also called graph model, unit-disk model [23], [24]) 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 model that enables the development of efficient algorithms 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 [25].

Network connectivity: a node can only discover nodes that are within its communication range under the protocol model. In a large-scale network, two nodes may not be connected directly and we call two nodes connected by one-hop communication as *neighbors*. Thus, a large-scale network is always *partially-connected*, contrary to a fully-connected network in which any two nodes are neighbors, i.e. they are connected by one-hop communication.

Energy-restricted: a node in the network has very limited energy and we assume a node can turn on or off its radio to save energy. 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 consisting of N nodes as set $U = \{u_1, u_2, \ldots, u_N\}$. Nodes are distributed in a large place and they can communicate through a fixed wireless channel. We assume the locations of the nodes obey some distributions, such as uniform distributions and Gaussian distribution [17]. Suppose a node has a fixed communication range Δ and two nodes u_i, u_j are neighbors if their distance suits $d(u_i, u_j) \leq \Delta$. We can use a symmetric matrix $M_{N\times N}$ to represent nodes' neighboring relations as:

$$M_{i,j} = \begin{cases} 1 & u_i \ and \ u_j \ are \ neighbors \\ 0 & otherwise \end{cases}$$

Suppose time is divided into slots of equal length t_0 [26], which is sufficient to finish a complete communication process (one node transmits a message including its ID and a neighbor receives the message). A node who turns its radio on can choose to be transmitting state or 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 message 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 listens 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 is a pre-define sequence $S_i = \{s_i(t)\}_{0\leq t<T}$ of period T, in which

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

We define *duty cycle* 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 (turning off radio, 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 slots cost to find the neighbor u_j and we define **discovery latency** of node u_i as the time to discover all neighbors:

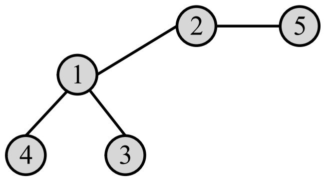
$$L(i) = \max_{j:M_{i,j}=1} L(i,j).$$

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 set of its neighbors $N_i = \{u_j | d(u_i, u_j) \leq \Delta\}$, design duty schedules for all nodes such that: $\forall u_j \in N_i$:

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

Fig.1 shows an example of 5 nodes, the topology is depicted in Fig.1(a), and Fig.1(b) describes the neighbor discovery



(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.

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.

IV. Alano Algorithm for a Large Scale Network

In a large scale network, nodes are not fully connected and communications may fail due to concurrent transmissions. When we do not consider 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. Suppose the locations of nodes obey some distribution, we propose Alano algorithm and analyze its performance for two common used distribution (uniform distribution and Gaussian distribution).

A. Alano Algorithm

Suppose the locations of nodes obey some distribution and node u_i is aware of its position coordinates (x_i, y_i). Then u_i could compute its local density as follows [24], [27].

Denote the general density function as :

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

where (x, y) is a position coordinate, and D is the network covering area.

For each node u_i with position (x_i, y_i), the range of its neighbors' positions (denote as R_i) is formulated as:

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

where Δ is the communication range. Then, node u_i's expected number of neighbors (denote as $NB(u_i)$ is:

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

In a large scale network, we ignore the boundary area of the network and assume nodes in the network are of an enormous quantity. Then, the $NB(u_i)$ can be rewritten as:

$$NB(u_i) = N \iint_{R_i} \varphi(x,y)\, dx\, dy.$$

When the network covering area D is much larger than the area R_i of node u_i,

$$NB(u_i) \simeq N\pi r^2 \varphi(x,y). \tag{1}$$

We present **Alano**, a randomized algorithm for node u_i in Alg. 1. By computing the expected number of nodes, u_i choose to be in transmitting state or listening state according to the generated probability on Line 2.

Algorithm 1 Alano Algorithm

1: $\hat{n_i} = N \iint_{R_i} \varphi(x,y)\, dx\, dy$;
2: $p_t^i = \frac{1}{\hat{n_i}}$;
3: **while** $True$ **do**
4: A random float $\epsilon \in (0, 1)$;
5: **if** $\epsilon < p_t$ **then**
6: Transmit a message containing node information of u_i;
7: **else**
8: Listen on the channel and decode the node information if receive a message successfully;
9: **end if**
10: **end while**

In the following parts, we first consider a general situation that nodes in the network are uniform 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. Then we analyze a more common situation that nodes obey Gaussian Distribution, we present an approximation analysis that the discovery latency is not be much larger than that of uniform distribution.

B. Analysis for Uniform Distribution

Uniform distribution is a basic one for deployment of wireless networks. For instance, to monitor an unknown area, many sensors are deployed uniformly to sense information, such as temperature and humidity [**?**]. The nodes are evenly deployed and the density function is:

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

where A is the area of D.

By Eqn. (1), u_i's expected number is $\hat{n} = \frac{N\pi r^2}{A}$ and the probability in Line 2 is set as $p_t = \frac{1}{\hat{n}} = \frac{A}{N\pi r^2}$.

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:

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

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

$$p_s' = (1-p_t)^{\hat{n}-1} - (\hat{n}-1)p_t(1-p_t)^{\hat{n}-2}.$$

It is easy to verify that when $p_t = \frac{1}{\hat{n}}$, p_s gets the maximum value:

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

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

We analyze the expected discovery latency for a node u_i to discover all neighbors. Denote W_j as a random variable representing the cost number of the time slots to discover a new neighbor after $j-1$ neighbors have been discovered. It is easy to check that W_j follows Geometric distribution with parameter $p(j)$ where $p(j) = (\hat{n}-j+1)p_s$ **[?]**. The expectation of W_j can be computed as:

$$E[W_j] = \frac{1}{p(j)} = \frac{1}{(\hat{n}-j+1)p_s}.$$

Then, the expectation discovery latency of node u_i is:

$$E[W_j] = \sum_{j=1}^{\hat{n}} \frac{1}{p_s} H_n \approx ne(lnn + \Theta(1)) = \Theta(nlnn).$$

where H_n is the n-th Harmonic number $H_n = lnn + \Theta(1)$.

The expected discovery latency is bounded by $O(n \ln n)$, and we show that the discovery latency can not be much larger than the expectation.

If W_i is already computed, the value of W_j will not be affected for $i < j$. That is, for $i \neq j$, W_i and W_j are independent and they satisfy $P(W_j = w_j | W_i = w_i) = P(W_j = w_j)$. Since W_j follows Geometric distribution, $Var[W_j] = \frac{1-p_j}{p_j^2}$, and the variance of discovery latency (denoted as W???what is W) is computed as:

$$Var[W] = \sum_{j=1}^{n} Var[W_j] \leq \frac{\pi^2}{6p_{suc}^2} - \frac{H_n}{p_{suc}}.$$

By *Chebyshev's inequality*, the probability that the discovery latency is 2 times larger than the expectation is

$$P[W \geq 2E[W]] \leq \frac{Var[W]}{E[W]^2} \leq \frac{\pi^2}{6H_n^2} - \frac{p_{suc}}{H_n}.$$

For a large n, $P[W \geq 2E[W]]$ is close to 0. That is, the time for a node to discover all neighbors is very likely to be smaller than 2 times of the expectation. Therefore, W is also bounded by $O(n \ln n)$.

C. Analysis of Gaussian Distribution

Gaussian distribution is common adopted in wireless network. For instance, an intrusion detection application needs larger detection probability around important entities [17]. We assume the positions of nodes obey 2D Gaussian distribution, and we present a theoretical proof that the discovery latency is not much larger than that of uniform distribution.

Denote the approximate neighbors of node u_i as set $N_i = \{u_{i1}, u_{i2}, ..., u_{i\hat{n}_i}\}$. When nodes obey Gaussian distribution, the probability that node u_i discovers a certain neighbor node u_{ij} successfully in a time slot (denote as P_{suc}) can be formulated as:

$$p_{suc} = (1-p_t^i)p_t^{ij} \prod_{k=1}^{\hat{n}_i, k\neq j} (1-p_t^{ik})$$

Denote $p_t^{imax} = \max_{1\leq j\leq \hat{n}_i}\{p_t^{ij}\}$, $p_t^{imin} = \min_{1\leq j\leq \hat{n}_i}\{p_t^{ij}\}$, for every $1 \leq j \leq n_i$, we have:

$$\begin{aligned}
&(1-p_t^i)p_t^{imin}(1-p_t^{imax})^{\hat{n}_i-1}\\
\leq&(1-p_t^i)p_t^{ij} \prod_{k=1}^{\hat{n}_i, k\neq j} (1-p_t^{ik})\\
\leq&(1-p_t^i)p_t^{imax}(1-p_t^{imin})^{\hat{n}_i-1}
\end{aligned}$$

Denote:

$$P = (1-p_t^i)p_t^{imin}(1-p_t^{imax})^{\hat{n}_i-1}$$
$$Q = (1-p_t^i)p_t^{imax}(1-p_t^{imin})^{\hat{n}_i-1}$$

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

$$\begin{array}{ccccc}
\frac{1}{\hat{n}_i Q} & \leq & E[W_1] & \leq & \frac{1}{\hat{n}_i P}\\
\frac{1}{(\hat{n}_i-1)Q} & \leq & E[W_2] & \leq & \frac{1}{(\hat{n}_i-1)P}\\
 & & .... & & \\
\frac{1}{Q} & \leq & E[W_{\hat{n}_i}] & \leq & \frac{1}{P}
\end{array}$$

Combine the above equations to derive:

$$\frac{1}{Q}H_n \quad \leq \quad E[\sum_{j=1}^{\hat{n}_i} W_j] \quad \leq \quad \frac{1}{P}H_n$$

(???)Since the sensible neighbors are within a close distance of the node compared to the total network area, which implies the density function values are within the same order of magnitude. Thus we can conclude that the expectation of the time latency in the normal distributed networks are still $E[W] = O(nlnn)$. Similarly the bounded latency can be proved to be $W = O(nlnn)$ in the same way as uniform distribution.

V. Modified Alano for An Energy-Restricted Network

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

In the preceding section, energy constraint is not a crucial factor in Alano; and we are to modify Alano for both symmetric nodes and asymmetric nodes. Our initiative idea is to align the slots that the radio is on with the neighbors in a bounded time; 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 Relax Difference Set (RDS) to align time slots that nodes are in transmitting or listening state.

RDS is an efficient tool to construct cyclic quorum systems [18], [19]. The definition is:

Definition 1: A set $R = \{a_1, a_2, ..., a_k\} \subseteq Z_n$ (the set of all nonnegative integers less than n) is called a RDS if for every $d \neq 0$ (mod n), there exists at least one ordered pair (a_i, a_j) such that $a_i - a_j \equiv d$ (mod n), where $a_i, a_j \in D$.

It has been proved that any RDS must have cardinality $|R| \geq \sqrt{N}$ [19]. 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: $R := \emptyset$;
2: $\lambda := \lceil \sqrt{N} \rceil$,$\mu := \lceil \frac{\lceil \sqrt{N} \rceil}{2} \rceil$;
3: **for** $i = 1 : \lambda$ **do**
4:     $R := R \cup i$;
5: **end for**
6: **for** $j = 1 : \mu$ **do**
7:     $R := R \cup (1 + j * \lambda)$;
8: **end for**

The intuitive idea of Alg. 2 can be described as Fig. 2. The framed elements are selected as Line. 4 and Line. 7. We show the correctness of the construction formally.

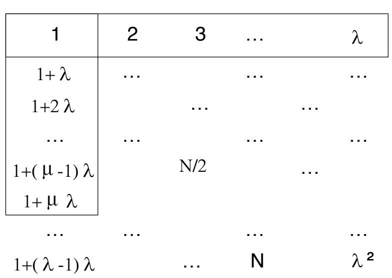


Fig. 2. An Sketch of RDS construction in Alg. 2

Lemma 1: 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 opposite 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 2 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. 4 and $a_i = i$ from Line. 7 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

$T := \lceil \frac{9}{4\theta^2} \rceil$;
Invoke Alg. 2 to construct the RDS $R = r_0, r_1, ..., r_{\lceil \frac{3\sqrt{T}}{2} \rceil}$ under Z_T;
$t := 0$;
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 1: 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. Regarding each period of T time slots as a 'super' slot of Alano algorithm, we can derive that the discovery latency is bounded within $O(\frac{n \log n}{\theta^2})$ slots with high probability by combining the analysis of Alano. ■

(??? not clear) There is a detail to be noticed when regarding the whole period T as a time slot in Alg. 1. Due to the periodicity, a node will rendezvous the same neighbors in the identical wake-up slot in each period. Since for a certain node there are more than one wake-up slots in each period, the rendezvoused neighbors in each wake-up time slot may be equal or less than the total number of its neighbors, resulting less collisions and lower latency compared to when all the neighbors wake up in the same time 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 align 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.

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. 5).

Algorithm 4 Traversing Pointer Based Alano Algorithm

```
1:  T := Find the smallest prime ≥ 2/θ_i;
2:  t := 0;
3:  while True do
4:      t_1 := t%T;
5:      t_2 := ⌊t/T⌋%(T − 1) + 1;
6:      if t_1 = 0||t_1 = t_2 then
7:          Invoke Alg. 1 to determine transmission state;
8:      else
9:          Sleep;
10:     end if
11:     t := t + 1;
12: end while
```

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. 3.

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 expected duty cycle.

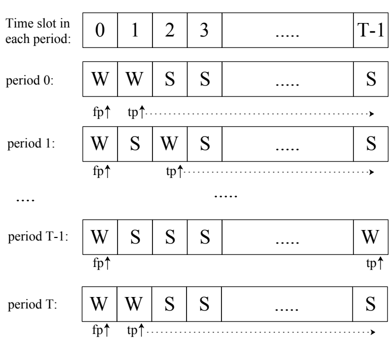


Fig. 3. A sketch of TP construction in Alg. 4

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

Theorem 2: 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 [20], there exists a time slot $t_\tau \in [0, T_iT_j)$ satisfying:

$$0 = t_\tau \quad mod \quad T_i. \tag{2}$$

$$\delta_{ij} = t_\tau \quad mod \quad T_j. \tag{3}$$

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. Regarding the whole period T_iT_j as a 'super' slot of Alano, we derive that the discovery latency is bounded within $O(\frac{nlogn}{\theta_i\theta_j})$ with high probability. ■

VI. EVALUATION

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.

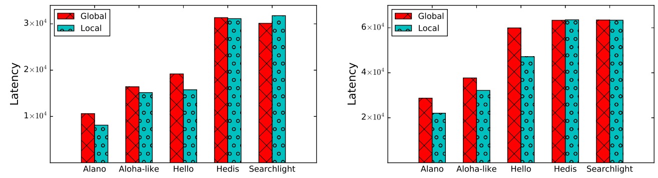
We simulated a network that follows uniform distribution and Gaussian distribution respectively. For uniform distribution, we suppose 500 nodes are distributed in an area of $100m \times 100m$ and each node's communication range is $\Delta = 10m$. For Gaussian distribution, we suppose 1000 nodes are distributed in the same area, but each node's communication range is $\Delta = 5m$. We set the Gaussian distribution as $N(50, 15^2)$ in our evaluation. We set time slot length to be $20ms$ and 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 and realistic than that in [6]–[15].

We evaluated discovery latency of Alano, Aloha-like [14], Hello [8], Hedis [9], and Searchlight [7] in the generated partially-connected networks. As the deterministic algorithms, Hello, Hedis and Searchlight, only have two states $\{ON, OFF\}$ (representing the radio is on or off). To make the comparison more fair, we assume that when a node is in $\{ON\}$ state, it transmits or listens with equal probability 0.5. In the following parts, we show that Alano has lower latency, higher discovery rate, better scalability, and better robustness.

A. Speed: Discovery Latency

When nodes follow uniform distribution, we show the discovery latency comparison for both symmetric nodes and asymmetric nodes in Fig. 4(a). From the figure, Alano achieves 54.64% to 1.95 times lower discovery latency for symmetric

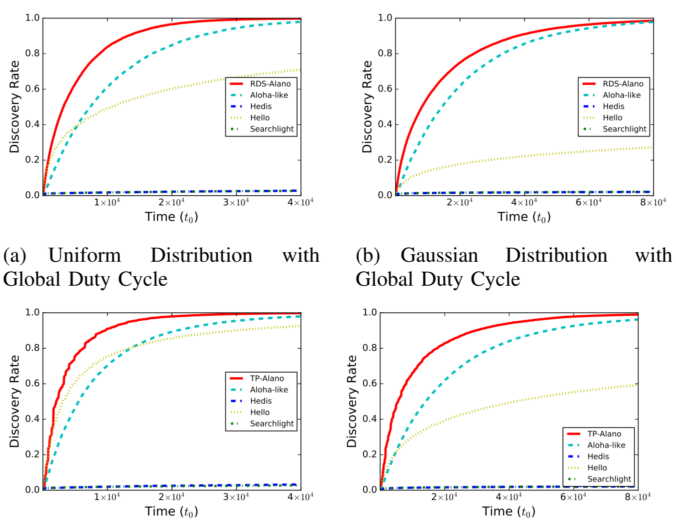


(a) Uniform Distribution (b) Gaussian Distribution

Fig. 4. Alano achieves lower latency.

nodes, and 85.25% to 2.91 times lower discovery latency for asymmetric nodes. When nodes follow Gaussian distribution, as depicted in Fig. 4(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 because these protocols only considered the bounded latency between two nodes. When the network becomes denser and a node has many neighbors, they cannot discover rapidly due to communication collisions.

B. Quality: Discovery Rate



(a) Uniform Distribution with Global Duty Cycle

(b) Gaussian Distribution with Global Duty Cycle

(c) Uniform Distribution with Local Duty Cycle

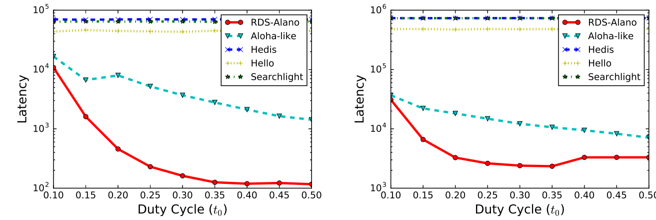
(d) Gaussian Distribution with Local Duty Cycle

Fig. 5. Alano achieves higher discovery rate in larger networks.

We use discovery rate to evaluate Alano's quality. Discovery rate of a node u_i is defined as the percentage of discovered neighbors over u_i's all neighbors. In Fig. 5, we increase the number of nodes from 500 to 1000 for the uniform distribution, and increase the number of nodes from 1000 to 2000 for the Gaussian distribution, the results show that Alano has higher discovery rate during the whole process for both uniform and Gaussian distributions. The deterministic algorithms, Hello, Hedis and Searchlight, cannot discover all nodes due to the occurrence of collisions.

C. Scalability: Duty Cycle and Network Density

We evaluated Alano's scalability regarding duty cycle and network density.



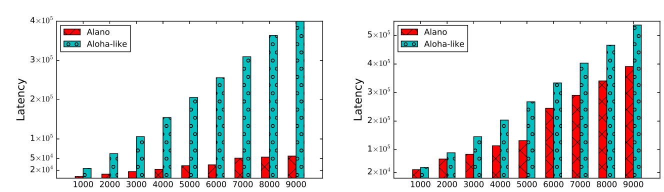
(a) Uniform Distribution (b) Gaussian Distribution

Fig. 6. Alano achieves lower latency in different duty cycle.

1. Duty Cycle When symmetric nodes with different duty cycles, Fig. 6 shows that Alano has lower latency. Compared with Aloha, Alano has from 53.66% to 11.23 times lower latency. The latency of Alano and Aloha generally decreases as the duty cycle increases, while Hello, Hedis and Searchlight have high latency due to the collision. In 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.

2. Network Density

When the number of nodes increases, the network becomes denser. We choose 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. 7(a), Alano achieves 4.68 times to 6.51 times lower discovery latency than Aloha-like algorithm for uniform distribution, when the number of nodes increases from 1000 to 9000. When the number of nodes increases from 1000 to 9000 for Gaussian distribution, Alano achieves 25.23% to 1.03 times lower discovery latency as shown in Fig. 7(b).

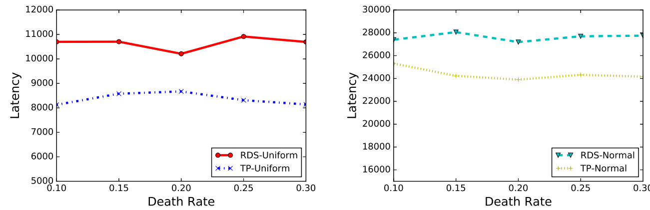


(a) Uniform Distribution (b) Gaussian Distribution

Fig. 7. Alano achieves lower latency with different number of nodes.

D. Robustness

In reality, a node may leave the network now and then (such as the node runs out of energy or the node is offline for other usage). We evaluate the robustness of the neighbor discovery algorithms in Fig. 8. When 5% to 30% nodes die (see x-axis of the figure), Alano still reaches low discovery latency for both uniform and Gaussian distributions.



(a) Uniform Distribution (b) Gaussian Distribution

Fig. 8. Alano still reaches low latency when nodes die.

VII. Conclusion

In this paper, we initiate the study of neighbor discovery in an energy-restricted large-scale network. To begin with, we propose Alano for a large-scale network where nodes' distribution are utilized to trigger the design of a node's transmitting probability. For different distributions, such as uniform distribution and normal distribution, we show that Alano achieves nearly optimal discovery latency. Then, we propose two modification methods for a 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, the results show that Alano achieves better performance regarding discovery latency, discovery rate (quality), scalability and robustness.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] 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*, vol. 10, no. 2, pp. 18–25, 2006.

[3] 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, pp. 373–374.

[4] 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*, vol. 2, no. 09, p. 655, 2010.

[5] 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*. ACM, 2008, pp. 71–84.

[6] 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, pp. 350–361.

[7] 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, pp. 185–196.

[8] W. Sun, Z. Yang, K. Wang, and Y. Liu, "Hello: A generic flexible protocol for neighbor discovery," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 540–548.

[9] 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*. IEEE, 2015, pp. 693–701.

[10] K. Wang, X. Mao, and Y. Liu, "Blinddate: A neighbor discovery protocol," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 949–959, 2015.

[11] 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*. IEEE, 2016, pp. 1–9.

[12] 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*. ACM, 2001, pp. 137–145.

[13] 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, pp. 181–192.

[14] 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*. IEEE, 2011, pp. 749–754.

[15] 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*. IEEE, 2014, pp. 1–5.

[16] 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*, vol. 2. IEEE, 1996, pp. 392–396.

[17] 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*, vol. 24, no. 2, pp. 342–355, 2013.

[18] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks," *Mobile Networks and Applications*, vol. 10, no. 1-2, pp. 169–181, 2005.

[19] 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, 1997, pp. 100–106.

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

[21] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*. ACM, 2004, pp. 125–138.

[22] 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, 2011, pp. 1093–1098.

[23] T. Moscibroda and R. Wattenhofer, "The complexity of connectivity in wireless networks." in *Infocom*, 2006, pp. 1–13.

[24] L. Wang, P.-J. Wan, and W. Washington, "Connectivity of multihop wireless networks with log-normal shadowing," *Wireless Networks*, vol. 21, no. 7, pp. 2279–2292, 2015.

[25] E. Lebhar and Z. Lotker, "Unit disk graph and physical interference model: Putting pieces together," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*. IEEE, 2009, pp. 1–8.

[26] 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.

[27] R. Meester and R. Roy, *Continuum percolation*. Cambridge University Press, 1996, vol. 119.