Alano: May You Discover Your Neighbors

***Abstract*—The abstract goes here.**

..

..

..

..

..

..

..

..

..

..

..

..

..

..

..

..

..

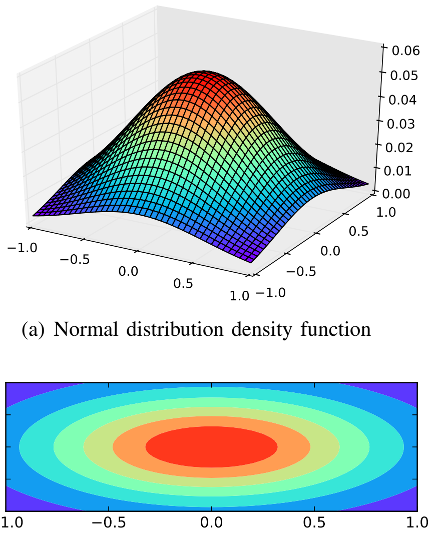
..

..

..

..

..



(a) Normal distribution density function

(b) Network obeys normal distribution

Fig. 1. An example of a network obeying normal distribution.

I. Introduction

Information and Communication Technology (ICT) equipment has exploded on the scene in the last twenty years [1]. Varying from their specific applications, these devices consists of a wide variety of networks.

Neighbor discovery is a fundamental means for the devices to participate in network communication. A popular way to deal with this issue is to construct a deterministic sequence of transmission state in each time slot [2]–[8]. It holds an obvious advantage that the time to discovery a neighbor node can be bounded within a limited time. Nevertheless, most deterministic approaches only consider two nodes to discover each other, without further deliberating on the details (e.g. collisions from neighbors) when applying for multi-nodes. Instead, there are some probability based algorithms showing an ideal expected time bound for multi-nodes to find their neighbors. However, the weakness part of this kind of methods lies in the poor performance of discovery latency in the worst case.

A more crucial issue is that neither deterministic or probabilistic approaches consider a partially-connected networks in reality, the devices of which can only possess a fraction neighbors among all the nodes. Deploying a fully-connected network in a large-scale area, such as campus networks, wireless sensor networks, mobile gaming community, etc., is technically impractical due to the limited sensing range of devices communication. How far the other nodes can be detected as a neighbor for a mobile equipment depends on criterion such as the received signal strength. Thus a practical situation is that in a network , the nodes are partially connected with its detectable neighbors.

*********To be dealt with details later********

How we deal with partially-connected networks:

1. Consider distribution of network devices: uniform distribution normal distribution

2. propose a distribution based Alano alogrithm

*

*

*

*

***Motivation of energy-efficient networks:**

Among the partially-connected networks, there is a special one named energy-efficent network. Nodes in this type of networks have to maintain strict power budgets to attain years of lifetime [9]. Duty circle mechanism, the , is utilized to power-awareness ought to be fully taken into consideration. Correspondingly, the neighbor discovery process needs adjustment to deal with the dilemma between a balance of energy-efficiency and low- latency.

Contribution conclusion

In this paper, we first focus on the mathematical analysis of the distribution of the nodes in the networks. We give a expectation of neighbor number for each node in a network which obey**** propose Alano[1], a RDS-Alano algorithm for partially-connected networks as well as a TP-Alano algorithm for the energy-effienct networks.

The contributions of this paper are as follows:

1) We model the distribution of nodes in their networks and analyse the expectation number of neighbors of a node in uniform distribution and normal distribution and then propose Alano, a strategy
2) We propose a Relaxed DifferenceSet based Alano algorithm (RDS-Alano) for the global duty circle scenario.
3) We propose a Traversing Pointer based Alano algorithm (TP-Alano) for the local duty circle scenario.

***Rest part structure**

The remainder of the paper is organized as follows. The next section highlights some related work and puts forward some serious problems. Some notion definitions and the system model are given in Section III. We analyse the node's expectation number of neighbors and propose Alano algorithm in IV as a foundation. Section V describes the RDS-Alano algorithm for global duty circle scenario and TP-Alano algorithm for local duty circle scenario respectively in energy-efficient networks. We have conducted extensive simulations, and the results are shown in Section VI. Finally, we conclude the paper in Section VII.

II. Related Work

Introduce the representative existing algorithms.

1. Birthday Alg.
2. BlindDate Alg.
3. Disco Alg.
4. Hello Alg.
5. Searchlight Alg.
6. Talk More Listen Less Alg.
7. Todis&Hedis Alg.
8. U-Connect Alg.
9. ALOHA-like Alg.

(Note : Introduction and Related Work are expected to be 2 pages)

III. Preliminaries

In this section, we first give some notion definitions and introduce the collision detection mechanism. Then we formulate the Neighbor Discovery problem formally.

A. Sensor Node Model

The wireless sensor network consists of a number of sensors distributed separately in a target area. The deployed sensor nodes keep their most time in sleep pattern to avoid quick energy consumption and wake up timely to work on duty.

When a sensor wake up on a time slot, it can turn to either the transmitting state or listening state.

- **Transmitting state**. A node turn to transmitting state will broadcast a package containing its own identify information to all neighbors.
- **Listening state**. A node turn to listening state will monitor the frequency channel to collect its neighbors' packages. However collision will occur when two or more neighbor nodes transmit concurrently and thus no valid information will be gathered

Transiting between the states only costs little time, compared to one complete time slot.

In our model, we assume that each node has a unique identifier ID_i which is aware by theirselves, while the total amount of sensors N is not necessary to be known. Time is divided into slots of equal length t_0, which is sufficient to finish one communication process(transmit or receive a piece of package). In each time slot, a node transform its pattern according to a pre-defined duty schedule.

Definition 1: **Duty schedule** is a pre-defined sequence $S = \{s^t\}_{0\le t<T}$ of period T and

$$s^t = \begin{cases} S & Sleep \\ T & Transmit \\ L & Listen \end{cases}$$

Each node construct its own duty schedule according to a specific strategy and repeats it until finding all the neighbors. Since the waking-up duration has a significant affect on the battery's lifetime, duty circle is defined to restrict the energy consumption.

Definition 2: **Duty circle** represents the fraction of one period T where a node turns its radio on. It can be formulated as:

$$\theta = \frac{|\{0 \le t < T : s^t = T||s^t = L\}}{T}.$$

B. Collision Detection Mechanism

A collision detection mechanism allows a node to distinguish between the case where two or more nodes are transmitting and one where no node is transmitting. Indeed, practical solutions for collision detection have been proposed.

C. Problem Definition

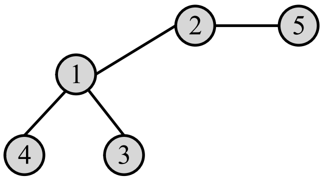
We consider a partially-connected sensor network, where two nodes are neighbors if they locate within the radio range of each other. A symmetric matrix $M_{N\times N}$ is used to record the neighboring relations as:

$$M_{i,j} = \begin{cases} 1 & connected \\ 0 & disconnected \end{cases}$$

Each sensor follows its duty schedule to achieve neighbor discovery. In a synchronous scenario, sensors start their neighbor discovery process at the same time, while in a asynchronous scenario all nodes start at different time slots.

Notice that the neighbor discovery process is not bidirectional, which means any pair of neighbors need to find each other separately. The time slots within which a sensor node u_i find one of its neighbors u_j can be formulated as $L(i, j)$. Then we define the discovery latency that node u_i discovers all neighbors as:

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



(a) The topology of a wireless sensor networks

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

Definition 3: **Discovery latency** of node u_i is the time to discover all neighbors:

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

Thus the neighbor discovery problem can be formulated as:

Problem 1: For a sensor node with identifier $ID\ i$, design a strategy of constructing a duty schedule with duty circle θ, which satisfies : for $\forall$ node $j_{M_{i,j}=1}$:

$\exists$ t s.t. :

$$S_i(t) = L$$

$$S_j(t) = T$$

$\forall$ node $k_{M_{i,k}=1\&\&k\neq j}$:

$$S_k(t) = L || S_k(t) = S$$

An example of neighbor dicovery process is given in Fig.2. Fig.2(a) shows the topology of a partially-connected wireless sensor networks, which consists of 5 sensor nodes. Fig.2(b) describes the neighbor discovery process in the asynchronous scenario, as we can see the nodes start their process at different time slot. The duty schedule of node 1, for example, is $S_1 = \{1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, ...\}$. At time slot 12, node 5 find its neighbor node 2 while node 1 could not find node 2 due to a collision from its another neighbor node 3.

IV. Partially-Connected Networks

A. Expectation Number Of Neighbor Nodes

B. Uniform Distribution

C. Normal Distribution

D. Alano:

V. Energy-Efficient Networks

In an energy-efficient network (i.e., wireless sensor networks), the battery-consumption is a crucial factor to be taken into account. Duty circle is a key technique to deal with the dilemma between a balance of energy-efficiency and low-latency.

We first consider the homogenous energy-arrangement situation that all the nodes share a global duty circle θ, and propose a RDS based Alano algorithm. Then we propose a traversing pointer based Alano algorithm for a more general scenario, where nodes have heterogeneous battery-scheduling capability with local duty circle θ_i.

Our initiative idea is to align the wake-up slots of the neighbor nodes within a bounded time, and then invoke the Alano algorithm to achieve neighbor discovery process w.h.p. More specifically, we utilize the property of RDS and traversing pointer to guarantee a wake-up slot rendezvous in each period T.

A. Global θ: A RDS Based Alano Algorithm

When a global duty circle θ is shared by all the nodes in the network, we utilize relaxed difference set (RDS) to align the wake-up time slots.

Relaxed difference set (RDS) is an efficient tool to construct cyclic quorum systems [10], [11]. The definition can be described as:

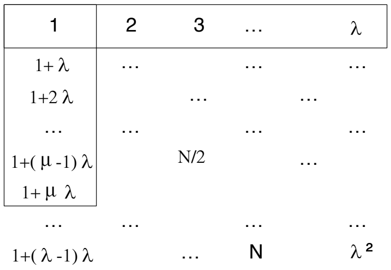
Definition 4: A set $R = \{a_1, a_2, ..., a_k\} \subseteq Z_n$ (the set of all nonnegative integers less than n) is called a Relaxed Difference Set (RDS) if for every $d \neq 0$ (mod n), there exists at least one ordered pair (a_i, a_j) such that $a_i - a_j \equiv d$ (mod n), where $a_i, a_j \in D$.

It has been proved that any RDS must have cardinality $|R| \geq \sqrt{N}$ [11]. We present a simple linear algorithm for RDS construction under Z_N with$\lceil \frac{3\sqrt{N}}{2} \rceil$ cardinality in Alg. 1.

Algorithm 1 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 initiative idea of Alg. 1 can be described as Fig. 3. The framed elements are selected as in Alg. 1 Line. 4 and Line. 7.



1	2	3	…		λ
$1+\lambda$	…		…		…
$1+2\lambda$		…		…	
…	…		…		…
$1+(\mu-1)\lambda$		N/2		…	
$1+\mu\lambda$					
…	…		…		…
$1+(\lambda-1)\lambda$		…	N		λ^2

Fig. 3. An Sketch of RDS construction in Alg. 1

We give a formal correctness proof of the construction as following:

Theorem 1: The set $R = \{r_0, r_1, ..., r_{\lambda+\mu-1}\}$ constructed in Alg. 1 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: We first reach a consensus that if there exists one ordered pair (a_i, a_j) satisfying $a_i - a_j \equiv d$ (mod N), then we can get an opposite pair (a_j, a_i) 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 from 1 to $\lfloor N/2 \rfloor$.

The λ in Line 2 is the smallest integer satisfying $\lambda^2 \geq N$. Then every d from 1 to $\lfloor N/2 \rfloor$ can be represented as: $d = 1 + j \times \lambda - i$, where $1 \leq j \leq \mu, 1 \leq i \leq \lambda$. Thus there exists $a_j = 1 + j \times \lambda$ added in Line. 4 and $a_i = i$ added in Line. 7 satisfying $a_j - a_i \equiv d$. ■

Next, we present a RDS based Alano algorithm (RDS-Alano) in Alg. 2, to achieve neighbor discovery process in a partially-connected and energy-efficient network with global duty circle θ.

In Alg. 2, RDS is used to construct a deterministic schedule for the node to wake up in every period T, and Alano is utilized as a probabilistic strategy to determine the transmission state (transmit or listen) in each wake-up slot.

Algorithm 2 RDS Based Alano Algorithm

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

We show a proof of time latency bound for Alg. 2 as following:

Theorem 2: Alg. 2 guarantees the discovery latency to be bounded within $O(\frac{nlog}{\theta^2})$ w.h.p.

Proof: It is easy to confirm that the duty circle $\widetilde{\theta}$ in Alg. 2 corresponds to θ as:

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

For any pair of neighbors (node i, node j), we can find an ordered pair (r_i, r_j) from their respective RDS such that $r_i - r_j \equiv \delta_t$ (mode T), which indicates any neighbor pair can wake up in the same time slot at least once in every period T. Regarding the whole period T as a time slot in Alg. **??**, we obtain the latency bound as $O(\frac{nlogn}{\theta^2})$. ■

B. Local θ: A Traversing Pointer Based Alano Algorithm

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 heterogenous capability of battery-management with local duty circle θ_i.

We propose a traversing pointer based Alano algorithm(TP-Alano) in Alg. 3. In each period T, every node wakes up in two different time slots, one of which is the first slot of each period and another is a traversing slot different from period to period, as Alg. 3 Line . 5 indicates.

Algorithm 3 Traversing Pointer Based Alano Algorithm

```
 1: $T$ :=Find the smallest prime $\geq \frac{2}{\theta_i}$;
 2: $t := 0$;
 3: while $True$ do
 4:     $t_1 := t\%T$;
 5:     $t_2 := (t/T)\%(T - 1) + 1$;
 6:     if $t_1 = 0||t = t_2$ then
 7:         Invoke Alg. ?? to determine transmission state;
 8:     else
 9:         Sleep;
10:     end if
11:     $t := t + 1$;
12: end while
```

We call the first time slot in each period T as a *fixed pointer* and the traversing slots as a *traversing point*. These pointers are used to guarantee a wake-up time rendezvous in every period T_iT_j. A sketch of the pointers is described in Fig. 4.

Note that, since the period of T is selected as: *find the smallest prime* $\geq \frac{2}{\theta_i}$, which is likely to result in the consequence that the duty circle $\widetilde{\theta_i}$ in Alg. 3 is smaller than the expected θ. This can be easily solved by selecting some random wake-up time slots in each period T to conform to duty circle θ.

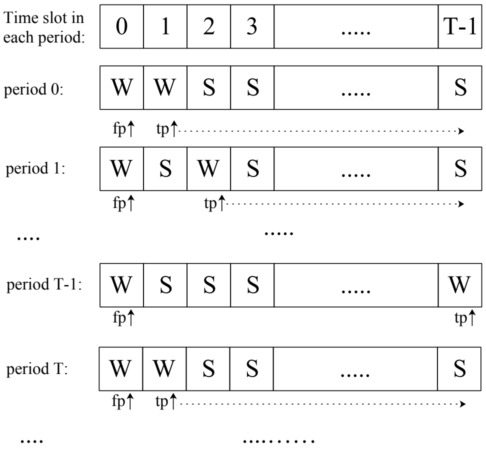

Time slot in each period:
0 | 1 | 2 | 3 | ..... | T-1
period 0:
W | W | S | S | ..... | S
fp
tp
period 1:
W | S | W | S | ..... | S
fp
tp
....
.....
period T-1:
W | S | S | S | ..... | W
fp
tp
period T:
W | W | S | S | ..... | S
fp
tp
....
....

Fig. 4. An Sketch of TP construction in Alg. 3

We give a correctness proof of the time bound to achieve neighbor discovery process as following:

Theorem 3: Alg. 3 guarantees the discovery latency to be bounded within $O(\frac{nlogn}{\theta_i\theta_j})$ w.h.p., where θ_i and θ_j are the duty circles of a pair of neighbors (node i,node$j$) respectively.

Proof: We first prove that any pair of nodes (node i,node$j$) will wake up at the same time slot every period T_iT_j.

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

$$0 = t_\tau \quad mod \quad T_i. \tag{1}$$

$$\delta = t_\tau \quad mod \quad T_j. \tag{2}$$

where δ is the asynchronous drift between node i and node j.

Equation1 and 2 implies there exists a fixed pointer of node i and a fixed pointer of node j will rendezvous in every T_iT_j .

Case 2: $T_i = T_j$. Since $T_i = T_j = T$, if the asynchronous drift between node i and node j $\delta = 0$, the fixed pointers of node i and node j will rendezvous with each other in every period T. Otherwise since the traversing point will traverse all the time slot once in each period $(T-1)T$, the traversing point of node i will consequentially rendezvous with the fixed pointer of node j once every period $(T-1)T$, and the traversing point of node j will consequentially rendezvous with the fixed pointer of node i once every period $(T-1)T$ as well.

Thus for any pair of neighbor (node i, node j), they can wake up at the same time slot at least once in every period T_iT_j. Regarding the whole period T_iT_j as a time slot in Alg. **??**, we obtain the latency bound as $O(\frac{nlogn}{\theta_i\theta_j})$. ■

VI. Evaluation

(Note : Evaluation is expected to be around 1 page)

VII. Conclusion

The conclusion goes here.

References

[1] S. Zeadally, S. U. Khan, and N. Chilamkurti, "Energy-efficient networking: past, present, and future," *The Journal of Supercomputing*, pp. 1–26, 2012.

[2] 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.

[3] 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.

[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*. ACM, 2008, pp. 71–84.

[5] 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.

[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, pp. 185–196.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

(Note : Conclusion and Reference are expected to be less than 1 page)