Neighbor Discovery in Wireless Networks

(Note : the paper title and the name of the proposed algorithm will be considered carefully and altered later.)

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

I. Introduction

Main components in this section:

1.The significance of this research and the background:

2.The weakness of the existing methods and the challenges:

The existing methods can be divided into two categories : deterministic or probabilistic.

The deterministic approaches assume there are no collision or are only suitable for two nodes. Some assume all the nodes are the neighbor of each other.

Some methods assume nodes have an estimate of N, which is the total amount of sensors in the network.

The probabilistic approaches have a poor performance and can not guarantee a worst case bound.

Challenge of my work: We consider the most demanding problem model in this work, which is also closest to the practical situation.

3.The main contribution and the key idea of my work:

Consider the partially-connected network, which is the most demanding problem model.

Assume nodes do not have an estimate of N.

Combine the advantages of both two category approaches.

The Proposed algorithm guarantees achieving neighbor discovery within $O(nf(\theta))$, which holds a better performance than the state-of-the-art works.

The core is to propose a deterministic approach to align the sensors' wake-up time and then achieve neighbor discovery with a probabilistic approach.

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.

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 $S = \{s^t\}_{0\leq t<T}$ of period T and

$$s^t = \begin{cases} 0 & sleep \\ 1 & wake-up \end{cases}$$

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.

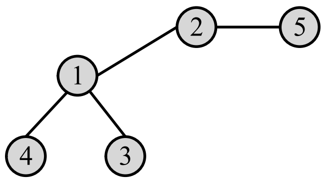
Definition 2: **Duty circle** represents the fraction of one period T where a node turns its radio on. It can be formulated as:

$$\theta = \frac{|\{0 \leq t < T : s^t = 1\}}{T}.$$

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.



(a) Topology

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.

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 in [5, 6]. In this paper, we study two neighbor discovery algorithms, first when nodes do not have a collision detection mechanism and next when nodes do and study the impact of collision detection on algorithm performance.

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:

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: Given a duty circle θ, design a duty schedule and transiting strategy which optimizes $L(i)$ to the most extent.

An example of neighbor dicovery process is given in Fig.1. Fig.1(a) shows the topology of a partially-connected wireless sensor networks, which consists of 5 sensor nodes. Fig.1(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. A

In this section, we propose a synchronous duty-time alignment (SDA) algorithm, a probabilistic approach to construct duty schedule in the synchronous scenario, as well as an asynchronous duty-time alignment (ADA) algorithm in the asynchronous scenario. Then we give the performance analysis and derive the upper bound.

The intuitive idea of constructing

A. Duty schedule construction based on SDA

B. Duty schedule construction based on ADA

C. Performance Analysis

V. B

In this section, we first describe a simple SDA based neighbor discovery algorithm for the synchronous scenario. Then we propose a ADA based neighbor discovery algorithm for the asynchronous scenario without and with collision detection mechanism respectively. Then we give the performance analysis and derive the upper bound.

A. A simple SDA based neighbor discovery algorithm

B. An ADA based neighbor discovery algorithm

without collision detection

with collision detection

VI. Evaluation

(Note : Evaluation is expected to be less than 2 pages)

VII. Conclusion

The conclusion goes here.

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