Ad Hoc Networks 77 (2018) 54–68

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

On heterogeneous duty cycles for neighbor discovery in wireless sensor networks

Lin Chen[a,b], Ruolin Fan[c], Yangbin Zhang[a], Shuyu Shi[d], Kaigui Bian[a,*], Lin Chen[e], Pan Zhou[f], Mario Gerla[c], Tao Wang[a], Xiaoming Li[a]

[a] Peking University, China
[b] Yale University, USA
[c] University of California, Los Angeles, USA
[d] National Institute of Informatics, Japan
[e] University of Paris-Sud, France
[f] Huazhong University of Science and Technology, China

ARTICLE INFO

Article history:
Received 10 November 2017
Revised 5 April 2018
Accepted 16 April 2018
Available online 27 April 2018

Keywords:
Neighbor discovery
Heterogeneous duty cycles
Wireless sensor networks

ABSTRACT

Neighbor discovery plays a crucial role in the formation of wireless sensor networks and mobile networks where the power of sensors (or mobile devices) is constrained. Due to the difficulty of clock synchronization, many asynchronous protocols based on wake-up scheduling have been developed over the years in order to enable timely neighbor discovery between neighboring sensors while saving energy. However, existing protocols are not fine-grained enough to support all *heterogeneous* battery duty cycles, which can lead to a more rapid deterioration of long-term battery health for those without support. Existing research can be broadly divided into two categories according to their neighbor-discovery techniques—the quorum-based protocols and the co-primality based protocols. In this paper, we propose two neighbor discovery protocols, called *Hedis* and *Todis*, that control the duty cycle granularity of quorum and co-primality based protocols respectively, by enabling the finest-grained control of heterogeneous duty cycles. We compare the two optimal protocols via analytical and simulation results, which show that the optimal co-primality based protocol (Todis) is not only simpler in its design, but also has a better performance.

1. Introduction

As human technology continues to advance at an unprecedented rate, there are more mobile wireless devices in operation than ever before. Many have taken advantage of the ubiquity of these devices to create mobile social network applications that use mobile sensing as an important feature [1,2]. These applications rely on their devices' capability to opportunistically form decentralized networks as needed. For this to happen, it is important for these devices to be able to discover one another to establish a communication link. In order to save energy, each of the devices alternates between active and sleeping states by keeping its radio "ON" for only some of the time [3]. This is challenging to achieve because two neighboring nodes have the opportunities of discovering each other only when both of their radios are "ON" at the same time; and with clock drifts, having set times for all the nodes to wake up at the same time is not trivial. Since clock synchronization is difficult in a distributed system, neighbor discovery must be done asynchronously. Over the years, the asynchronous neighbor discovery problem has been widely studied [4–13], and existing research mainly focused on satisfying the following three design requirements:

1. Guarantee neighbor discovery within a reasonable time frame;
2. Minimize the number of time slots for which the node is awake to save energy;
3. Match the nodes' wake-up schedules with their heterogeneous battery duty cycles[1] as closely as possible (i.e. finer duty cycle granularity).

Most existing solutions to this problem use patterned wake-up schedules to satisfy the first two requirements. We classify these solutions into two broad categories: (1) *quorum* based protocols that arrange the radio's time slots into a matrix and pick wake-

---

* Corresponding author.
E-mail address: bkg@pku.edu.cn (K. Bian).

[1] Duty cycle is the percentage of one period in which a sensor/radio is active.

https://doi.org/10.1016/j.adhoc.2018.04.007

up times according to quorums in the matrix; and (2) *co-primality* based protocols that use number theory to choose numbered time slots as the radio's wake-up times.

In a quorum-based protocol, a node populates time slots into a matrix, where the elements in the matrix represent time slots the node takes to run a period of the wake-up schedule [14]. The specific arrangements of rows and columns depend upon the protocol scheme, which typically assign slots as "active" or "sleeping", such that it will ensure these chosen active time slots in the matrix of one node will overlap with those active ones of a neighboring node. Especially, when nodes have the same duty cycles, two nodes choosing active times from a row and a column respectively in the matrix will be ensured to achieve neighbor discovery regardless of clock drifts.

A co-primality based protocol directly takes advantage of properties of the Chinese remainder theorem (CRT) [15] to ensure that any two nodes would both be active in the same time slot [6]. Under these protocols, nodes wake up at time slots in multiples of chosen numbers (a.k.a. protocol parameters) that are co-prime to one another. Such a neighbor discovery protocol fails when nodes choose the same number that would compromise the co-primality. Thus, every node is allowed to choose several numbers and wake up at multiples of all of those chosen numbers, which guarantees that nodes discover one another within a bounded time/delay.

Up to now, all of the protocols incepted, be it quorum-based or co-primality based, fail to meet the third design requirement, as their requirements for duty cycles are too specific. As a quorum-based protocol, Searchlight [4] requires that the duty cycles be in the form $\frac{2}{n^i}$, where n is a fixed integer and $i = 1, 2, 3, \ldots$ (it only supports duty cycles of $1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \ldots$ if $n = 2$). Therefore, it greatly restricts the choices of supported duty cycles due to the requirement for duty cycles to be in the form $\frac{2}{n^i}$. For a co-primality based protocol like Disco [6], it restricts duty cycles to be in the form $\frac{1}{p_1} + \frac{1}{p_2}$, where p_1 and p_2 are prime numbers. Such stringent requirements on duty cycles force devices to operate at duty cycles that they are not designed to operate at, thus shortening their battery longevity.

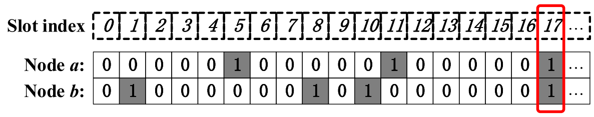
In this paper, we present two fine-grained neighbor discovery protocols, called *Hedis* (*heterogeneous discovery* as a quorum-based protocol) and *Todis* (*triple-odd based discovery* as a co-primality based protocol), that guarantee asynchronous neighbor discovery in a heterogeneous environment, meaning that each node could operate at a different duty cycle. We analytically compare these two protocols with existing state-of-the-art protocols to confirm their fine granularity in the support of duty cycles, and also compare them against each other as a comparison between the two general categories of neighbor discovery protocols (quorum vs. co-primality based protocols).

The rest of this paper is organized as follows. We formally define the problem as well as any necessary terms in Section 2, and give a taxonomy of current research efforts in this area in Section 3. In Sections 4 and 5, we present our optimizations for the quorum-based and co-primality based protocols respectively, and we evaluate them with simulations in Section 6. Finally, we conclude with Section 8.

2. Problem formulation

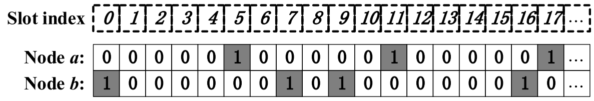
Here we define the terms and variables used to formally describe the neighbor discovery problem and its solution; and meanwhile we state the assumptions used in devising our protocols.

Wake-up schedule: We consider a time-slotted wireless sensor network where each node is energy-constrained. The nodes follow a *neighbor discovery wake-up schedule* that defines the time pattern of when they need to wake up (or sleep), so that they can discover their respective neighbors in an energy-efficient manner.



Slot index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	…
Node *a*:	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	…
Node *b*:	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	…

(a) Without clock drift



Slot index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	…
Node *a*:	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	…
Node *b*:	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	…

(b) Node b drifts by 1 time slot to the left

Fig. 1. An example of neighbor discovery: two neighbor discovery schedules are $\mathbf{s}_a = \{0, 0, 0, 0, 0, 1\}$ and $\mathbf{s}_b = \{0, 1, 0, 0, 0, 0, 0, 0, 1\}$. Without clock drift (a), the two nodes can discover each other every 18 time slots since $\text{lcm}(T_a, T_b) = 18$. With clock drift (b), neighbor discovery fails.

Definition 1. The neighbor discovery *schedule* (or simply schedule) of a node a is a sequence $\mathbf{s}_a \triangleq \{s_a^t\}_{0 \le t < T_a}$ of period T_a and

$$s_a^t = \begin{cases} 0 & a \text{ sleeps in slot } t \\ 1 & a \text{ wakes up in slot } t \end{cases}$$

We do not assume clock synchronization among nodes, therefore any two given nodes may have random clock drifts. We use the cyclic rotation of a neighbor discovery schedule to describe this phenomenon. For example, a clock drift by k slots of node a's schedule s_a is

$$rotate(\mathbf{s}_a, k) = \{r_a^t\}_{0 \le t < T_a},$$

where $r_a^t = s_a^{(t+k) \bmod T_a}$.

Definition 2. The *duty cycle* δ_a of node a is the percentage of time slots in one period of the wake-up schedule where node a is active (node a wakes up), defined as

$$\delta_a = \frac{|\{0 \le t < T_a : s_a^t = 1\}|}{T_a}.$$

For example, a node that wakes up on average in one slot for every 2 time slots has a duty cycle of 50%.

The importance of duty cycle matching: In a wireless mobile sensor network, each sensor node may have a different duty cycle due to various factors. By adjusting the duty cycles of a sensor, one is able to exploit the tradeoff between conserving battery power and packet forwarding capacity. A smaller duty cycle consumes less power because the radio is powered on for less of the time; however, because the radio is off for so long, the node cannot spend as much time transmitting packets, causing high end-to-end delays. On the other hand, as the duty cycle increases, the radio is powered on more frequently, thus mitigating delays while using up more battery power. Due to the ever-changing network conditions (periods of high and low traffic rates) and each sensor node's power status, the notion of having dynamic duty cycles is now an area of active research [16–18]. Thus, a neighbor discovery protocol must support duty cycles at a fine granularity in order for these new dynamic duty cycled schemes to come into fruition.

Neighbor discovery: Suppose two nodes a and b have schedules $\mathbf{s}_a$ and $\mathbf{s}_b$ of periods T_a and T_b, respectively. If $\exists t \in [0, \text{lcm}(T_a, T_b))$ such that $s_a^t = s_b^t = 1$ where $\text{lcm}(T_a, T_b)$ is the least common multiple of T_a and T_b, we say that:

- Nodes a and b can discover each other in slot t.
- Slot t is called a *discovery slot* between a and b.

Fig. 1 shows an example of two sensor nodes with neighbor discovery schedules $\mathbf{s}_a = \{0, 0, 0, 0, 0, 1\}$ and $\mathbf{s}_b =$