Never Live Without Neighbors: From Single- to Multi-Channel Neighbor Discovery for Mobile Sensing Applications

Lin Chen, *Member, IEEE*, Kaigui Bian, and Meng Zheng

***Abstract*—Neighbor discovery is of paramount importance in mobile sensing applications and is particularly challenging if the operating frequencies of mobile devices span multiple channels. In this paper, we formulate the *multi-channel neighbor discovery* problem and establish a theoretical framework of it, under which we derive the performance bound of any neighbor discovery protocol guaranteeing discovery. We then develop a multi-channel discovery protocol that achieves guaranteed discovery with order-minimum worst-case discovery delay and fine-grained control of energy conservation levels.**

***Index Terms*—Neighbor discovery, multi-channel, energy efficiency.**

I. INTRODUCTION

THE EVER-GROWING deployment of millions of personal mobile devices, e.g., smart-phones and tablets, generates numerous mobile sensing applications ranging from mobile social networking, intelligent transportation, proximity-based gaming, environment monitoring to participatory and crowd sensing. The success of such applications, where mobile devices equipped with different types of sensors interact with each other upon encounters, relies heavily on data timely collected and shared among the nearby users in an opportunistic fashion.

The supporting primitive that discovers all the neighbors in a mobile device's communication range is referred to as *neighbor discovery* protocol, which is one of the bootstrapping primitives supporting many basic network functionalities, such as topology control, clustering, medium access control, etc. Ideally, nodes should discover their neighbors as quickly as possible for other protocols to quickly start their execution.

Designing efficient neighbor discovery protocols for decentralized mobile sensing applications is particularly challenging due to the energy constraint of personal mobile devices. Specifically, the technique of duty cycling is used to reduce the energy consumption when these devices are in the idle state. Under duty cycling, each device alternates between active and sleeping modes by turning their radios on only periodically to achieve synchronization and save energy. *Duty cycle* refers to the fraction of time a device is in the active mode [2], [10]. For example, a device whose duty cycle is 1% activates during one time slot every 100 slots. Despite its effectiveness in energy conservation, the duty cycling technique significantly challenges the neighbor discovery protocol design in the quest of limiting discovery latency with low power consumption. Specifically, the two important design objectives, saving energy through a duty-cycle based scheduling and limiting the neighbor discovery latency, are at odd with each other.

Moreover, the operating frequencies of mobile devices typically span a swath of spectrum subdivided into multiple orthogonal channels. Such multi-channel characteristic brings an additional dimension to the neighbor discovery problem, as each pair of neighbors not only need to wake up at the same time slot, but also should switch to the same channel in order to discover each other. Wireless channels are notoriously unstable with channel conditions varying in both time and space domains. Any two nodes may have different channel perceptions due to their locations, traffic patterns, interference, noises, etc. Consequently, to achieve maximum discovery robustness, an effective neighbor discovery protocol needs to ensure discovery between any pair of neighbors on every common channel they can access.

The multi-channel paradigm, combined with the duty cycle based operation mode of wireless nodes, poses three major challenges for devising neighbor discovery protocols.

- *Lack of clock synchronization*: Due to resource constraint, it is extremely difficult to maintain tight synchronization among local clocks of wireless nodes, and thus the clocks of any two nodes may drift away from each other by an arbitrary amount of time, which may lead to discovery failure.
- *Asymmetrical duty cycle*: The duty cycles of two network nodes are typically asymmetrical, depending on their independent energy constraint and the applications running on them. Neighbor discovery protocols should ensure that

Manuscript received September 12, 2014; revised April 10, 2015; August 13, 2015; and October 15, 2015; accepted November 30, 2015; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor Y. Bejerano. The work of L. Chen was supported by the ANR grant Green-Dyspan (ANR-12-IS03). The work of K. Bian was supported by the NSFC under Grant 61572051. The work of M. Zheng was supported by the NSFC under Grant 61304263.

L. Chen is with the Laboratoire de Recherche en Informatique (LRI-CNRS UMR 8623), Université Paris-Sud, 91405 Orsay, France (e-mail: chen@lri.fr).

K. Bian is with the Institute of Network Computing and Information Systems, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: bkg@pku.edu.cn).

M. Zheng is with the Key Laboratory for Networked Control Systems, Shengyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China (e-mail: zhengmeng_6@sia.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2015.2505170

any two nodes can wake up at a same slot regardless of their duty cycles.
- *Asymmetrical channel perceptions*: Wireless channels are notoriously unstable with channel conditions varying in both time and space domains. Consequently, any two nodes may have different channel perceptions due to their locations, traffic patterns, interference, noises, etc.

We use the term *multi-channel neighbor discovery* problem to denote the following problem in the context described above: *How can neighbor nodes with heterogeneous duty cycles, operating on different channels, without clock synchronization, discover each other over every common channel within a bounded delay?* Particularly, the following requirements should be satisfied:
- Maximum discovery diversity/robustness with bounded (and minimum) worst-case discovery delay;
- Support for heterogenous and arbitrary duty cycles to provide fine-grained control of energy conservation levels.

We emphasize that it is the combination of the above design requirements that makes the neighbor discovery protocol design far from trivial and should be handled holistically. As reviewed in Section II, no existing work, to the best of our knowledge, can satisfy both requirements simultaneously. To address the multi-channel neighbor discovery problem, we present the design and evaluation of a multi-channel neighbor discovery protocol, termed as MCD (Multi-channel Discovery), that can guarantee discovery between any pair of neighbor nodes over *every* common channel within *bounded* delay with fine-grained control over duty cycles. The main contributions of the paper are articulated as follows.
- We establish a theoretical framework on the multi-channel neighbor discovery problem, under which we derive the performance bound of any neighbor discovery protocol achieving guaranteed discovery. For any protocol achieving full discovery diversity, we show that the lower-bound of the discovery delay L scales squarely in the number of channels N, and linearly in the duty cycle reciprocal of any of the two neighbor nodes.
- We design MCD, a multi-channel discovery protocol that achieves guaranteed discovery with order-minimal worst-case discovery delay and fine-grained control over duty cycles. We present analytical and simulation results, demonstrating the capability of MCD of ensuring discovery between any two neighbors on every common channel within bounded delay, even they have arbitrary clock drifts and asymmetrical channel perceptions.

Specifically, the major technicalities we devise to address the multi-channel neighbor discovery problem are (1) the use of consecutive odd numbers to approximate duty cycle with fine granularity and guaranteed discovery, (2) the design of channel hopping sequences to ensure discovery on every commonly accessible channels. The proposed neighbor discovery protocol, MCD, has the following properties making it especially suitable for mobile sensing applications.
- *Fine-grained control of duty cycle*: In contrast to existing solutions using prime numbers or power-multiples, MCD can support more than 95% of duty cycles in practical settings, thus providing much more fine-grained control of energy conservation levels.
- *Bounded worst-case discovery delay*: MCD achieves bounded discovery delay even between nodes with heterogeneous duty cycles.
- *Full discovery diversity*: MCD guarantees discovery over *each* channel, thus minimizing the probability of discovery failures caused by various channel problems.
- *Robustness against asymmetrical channel perceptions*: MCD achieves the same discovery performance even if nodes have asymmetrical channel perceptions, either on the accessible channel set or on the channel index.
- *Robustness against clock drift*: MCD achieves the same performance even if clocks of any two nodes drift away from each other by an *arbitrary* amount of time.

The rest of the paper is organized as follows. Section II summarizes the related work on neighbor discovery. Section III describes the system model and formulates the optimal multi-channel neighbor discovery problem. Section IV establishes the theoretical performance bound. Section V presents the design of MCD in the single-channel case and performs a theoretical analysis on its performance. Section VI further presents the design of MCD in the mult-channel case and investigates its performance there. Section VII presents the simulation results. Section VIII concludes the paper.

II. Related Work

The neighbor discovery protocols for duty-cycled networks in the literature, usually in OSI layers 1 and 2, can be categorized into *probabilistic* and *deterministic* protocols. In this section, we give a high-level overview of these two types of approaches and briefly analyze the pros and cons of each.

Probabilistic protocols (cf. major work in this category [15], [18], [22], [27], [31]) adopt probabilistic strategies at each node. Specifically, each node remains active or asleep with different probabilities. A representative one is the birthday protocol [18] where nodes transmit/receive or sleep with different probabilities. The work in [31] further addresses the case with multi-packet reception and directional antennas. Probabilistic protocols have the advantages of being memoryless and stationary and thus are especially robust and suitable for in decentralized environments where no prior knowledge or coordination is available. Moreover, they usually perform well in the average case by limiting the expected discovery delay. The main drawback of them is the lack of performance guarantee in terms of discovery delay. This problem is referred to as the long-tail discovery latency problem in which two neighbor nodes may experience extremely long delay before discovering each other.

Deterministic protocols, on the other hand, are proposed to provide strict upper-bound on the worst-case discovery delay (cf. major work in this category [4], [8], [13], [14], [16], [19], [26], [28], [32]). In deterministic protocols, each node wakes up according to its neighbor discovery schedule carefully tuned to ensure that each pair of two wake-up schedules overlap in at least one active slot. The key element in the deterministic protocol design is how to devise the neighbor discovery schedule to ensure discovery and minimize the worst-case discovery delay, regardless of the duty cycle asymmetry and the relative clock drift. Compared to probabilistic approaches that work well in the average case while fail to bound the worst-case discovery delay, deterministic protocols have good worst-case performance while usually have longer expected discovery delay.

More specifically, based on the design of wake-up schedule, major existing deterministic protocols can further be divided into three classes as briefly reviewed in the following.

- The first class of them, based on Quorum [16], [26], constructs the wake-up schedule by assigning a column and a row of an $m \times m$ array to each node such that no matter which row and column are selected, any two nodes have at least two overlapping awaken slots. The main drawback of the Quorum-based approaches is the support of only symmetrical duty cycles [26]. Although enhanced solutions have been proposed to support asymmetric duty cycles, only two different duty cycles can be supported [16].
- The second class of deterministic protocols overcome this limitation by using prime numbers to guarantee bounded discovery delay even for asymmetrical duty cycles. A typical one in this class is Disco [8], in which each node selects two prime numbers, based on which its wake-up schedule is configured. A more recent solution, U-Connect [14], uses a single prime number per node and has a shorter discovery delay, given the same duty cycle.
- The third class, Searchlight, proposed in [4] and a number of follow-up schemes [23], [28], employs two kinds of wake-up slots, termed as anchor and probe slots, to achieve both lower worst-case and average discovery delay.

One drawback of existing deterministic protocols is the failure to support all duty cycles due to their limited choices on either primes or power-multiples, and consequently only a limited choices of energy conservation levels can be supported. Recently, some probabilistic (cf. [17]) and deterministic (cf. [11] based on quorum systems) neighbor discovery protocols for multi-channel networks have been developed, but they either fail to provide bounded discovery delay ([17]) or only support symmetrical duty cycles ([11]).

Despite extensive research efforts devoted to neighbor discovery, none of them can solve the multi-channel neighbor discovery problem by achieving bounded discovery delay for nodes operating on heterogenous duty cycles, i.e., addressing the three design challenges posed in Section I simultaneously. In contrast, the solution we develop in this paper can achieve bounded discovery delay for nodes operating on heterogeneous duty cycles in multi-channel environments.

III. Multi-Channel Neighbor Discovery

A. System Model

We consider a time-slotted (but not necessarily synchronized) energy-constraint wireless network operating on a set $\mathcal{N}$ of $N \triangleq |\mathcal{N}|$ frequency channels. To discover its neighbors in the multi-channel environment, each node wakes up periodically and switches across different channels. The main design challenges we need to address are summarized as follows:

- *Lack of clock synchronization*: Due to the resource constraint, it is extremely difficult to maintain tight synchronization among the local clocks of different nodes, and thus the clocks of any two nodes may drift away from each other by an arbitrary amount of time, which may lead to the discovery failure.
- *Asymmetrical duty cycles*: The duty cycles of two network nodes are typically asymmetrical, depending on their independent energy constraint and the applications running on them. Neighbor discovery protocols should ensure that any two nodes can wake up in a same slot on the same channel regardless of their duty cycles.
- *Asymmetrical channel perceptions*: Wireless channels are notoriously unstable with channel conditions varying in both time and space domains. Consequently, any two nodes may have different channel perceptions due to their locations, traffic patterns, interference, noises, etc. Formally, each node u has its own perception on $\mathcal{N}$, denoted as $\mathcal{N}_u$.

Slot index	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	...
Node *a* :	0	0	1	0	0	1	0	0	1	0	0	1	...
Node *b* :	0	1	0	2	0	1	0	2	0	1	0	2	...

Fig. 1. Neighbor discovery schedule example.

In the following, we formally define the *neighbor discovery schedule* that characterizes the wake-up and channel hopping pattern of a node.

Definition 1 (Neighbor Discovery Schedule): The neighbor discovery schedule of a node u is defined as a sequence $\mathbf{x}_u \triangleq \{x_u^t\}_{1\leq t\leq T_u}$, where T_u is the period of the sequence,[1] and

$$x_u^t = \begin{cases} 0 & u \text{ sleeps in slot } t \\ h \in \mathcal{N}_u & u\text{, operating on channel } h\text{, wakes up.} \end{cases}$$

Definition 2 (Duty Cycle): The duty cycle of a node u, denoted by δ_u, is defined as the percentage of slots per period of the neighbor discovery schedule $\mathbf{x}_u$ where u is active. Formally, δ_u is defined as

$$\delta_u \triangleq \frac{|t \in [0, T_u - 1] : x_u^t = 1|}{T_u}.$$

The reciprocal of δ_u is denoted by d_u.

Consider two nodes a and b with their neighbor discovery schedules being $\mathbf{x}_a$ and $\mathbf{x}_b$ whose periods are T_a and T_b. Given the periodicity of $\mathbf{x}_a$ and $\mathbf{x}_b$, it suffices to consider consecutive T_aT_b slots, i.e, $1 \leq t \leq T_aT_b$. If $\exists t \in [1, T_aT_b]$ and $h \in \mathcal{N}$ such that $x_a^t = x_b^t = h$, we say that a and b can discover each other in slot t on channel h. Slot t is called the discovery slot and channel h is called the discovery channel between a and b. Example 1 illustrates the above definition.

Example 1: Consider a network of two channels and two nodes a, b whose neighbor discovery schedules are $\mathbf{x}_a = \{0, 0, 1\}$ and $\mathbf{x}_b = \{0, 1, 0, 2\}$ with $T_a = 3$ and $T_b = 4$. The duty cycles of a and b are $\delta_a = \frac{1}{3}$ and $\delta_b = \frac{1}{2}$, or $d_a = 3$, $d_b = 2$. The neighbor discovery schedules of a and b are repeated each 12 slots, as illustrated in Fig. 1 for one period. We can observe that a and b can discover each other on slots 6 on channel 1.

[1]A random neighbor discovery schedule is a special case where $T_u \to \infty$.

To model the situation where the clocks of different nodes are not synchronized,[2] we apply the concept of *cyclic rotation* to neighbor discovery schedules. Specifically, given a neighbor discovery schedule $\mathbf{x}_a$, we denote $\mathbf{x}_a(k)$ a cyclic rotation of $\mathbf{x}_a$ by k slots where k is called the cyclic rotation phase. In Example 1, we have $\mathbf{x}_a(1) \triangleq \{1,0,0\}$ and $\mathbf{x}_b(2) = \{0,2,0,1\}$.

B. Optimal Multi-Channel Neighbor Discovery

Performance Metric 1: Maximum Time to Discovery: In neighbor discovery, the primary performance metric is the maximum time to discovery (MTTD), i.e., the worst-case discovery delay. Given two nodes a and b, the MTTD between them is defined as the upper-bound of the latency (in number of slots) before successful mutual discovery for all possible clock drift between them. Reconsider Example 1, we can observe that the MTTD is 11, achieved between $\mathbf{x}_a(6)$ and $\mathbf{x}_b(6)$.

Performance Metric 2: Discovery Diversity: The second metric, particularly pertinent for the multi-channel environment, is the *discovery diversity*, which characterizes the capability of a neighbor discovery protocol of discovering a neighbor regardless of its operational channel. We say that a neighbor discovery protocol achieves *full* discovery diversity if the discovery of any pair of nodes is guaranteed on *every* common channel they can access. The neighbor discover schedule in Example 1 cannot achieve full discovery diversity as a and b can never discover each other on channel 2.

Performance Metric 3: Maximum Time to Discovery With Full Diversity: When full discovery diversity can be achieved, we further define the third metric *maximum time to discovery with full diversity* (MTTD-FD) as the worst-case delay to achieve full discovery diversity. The MTTD-FD can be regarded as a generalization of the MTTD in multi-channel networks. The MTTD-FD degenerates to the MTTD in single-channel networks. Throughout the paper, we analyze the MTTD in single-channel case and the MTTD-FD in multi-channel case.

We conclude this section by formulating the optimal multi-channel neighbor discovery problem.

Problem 1: The optimal multi-channel neighbor discovery problem is defined as follows:

$$\begin{aligned} &\textit{minimize} \quad T, \\ &\textit{subject to} \quad \forall t_a^0 \in [1, T_a], t_b^0 \in [1, T_b], \forall \delta_a, \delta_b, \exists t \le T \\ &\qquad\qquad \textit{such that } x_a^t(t_a^0) = x_b^t(t_b^0) = h, \forall h \in \mathcal{N}_a \bigcap \mathcal{N}_b. \end{aligned}$$

That is, devising neighbor discover schedules to minimize T, the worst-case discovery delay while achieving full discovery diversity between any pair of nodes a and b for any duty cycle pair (δ_a, δ_b), any initial time offset t_a^0 and t_b^0, and any channel perception $\mathcal{N}_a$ and $\mathcal{N}_b$.

In what follows, we first establish a theoretical performance bound of any neighbor discovery protocol. We then present the baseline design and optimization of MCD in the single-channel case, before proceeding to the multi-channel case with symmetrical channel perception (i.e., $\mathcal{N}_a = \mathcal{N}_b$). We complete our analysis by addressing the generic case with asymmetrical channel perceptions and arbitrary clock drift to iron out a version of MCD that works in practice.

[2]Here we assume that clocks of different nodes are asynchronous but their slot boundaries are aligned. The situation of unaligned slot boundaries is analyzed in Section VI-D.

IV. Protocol-Independent Discovery Delay Bound

Armed with the theoretical framework established previously, this section derives the performance bound of any multi-channel neighbor discovery protocol achieving full discovery diversity. The result derived in this section establishes the lower-bound of the solution of Problem 1.

Theorem 1 (Protocol-independent Bound of MTTD-FD): For any neighbor discovery protocol achieving full discovery channel diversity, the MTTD-FD between any pair of nodes a and b, denoted by L, is lower-bounded by $N^2 d_a d_b$, where d_a and d_b denote the reciprocals of the duty cycles of a and b, i.e., $d_a = \frac{1}{\delta_a}$ and $d_b = \frac{1}{\delta_b}$.

Proof: Let T_a and T_b denote the period of $\mathbf{x_a}$ and $\mathbf{x_b}$, i.e., the neighbor discovery schedules of a and b. It can be noted that regardless of the clock drift, the neighbor discovery schedules of a and b repeats every $T_a T_b$ time slots. Hence, if they can discovery each other with full discovery diversity regardless of the clock drift, the worst-case discovery delay until full diversity L is upper-bounded by $T_a T_b$.

Without loss of generality, we fix $\mathbf{x_a}$ and cyclically rotate $\mathbf{x_b}$ by l slots, denoted as $\mathbf{x_b}(l)$, where $l = 0, 1, \ldots, T_a T_b - 1$. Now consider $\mathbf{x_a}$ and $\mathbf{x_b}(l)$. Recall that the MTTD-FD is the worst-case discovery delay until full diversity among all initial clock phases of a and b, there must be at least N discovery slots each L slots where both a and b wakes up in the slot, resulting a minimal number of discovery slots $\frac{N T_a T_b}{L}$ within consecutive $T_a T_b$ slots. Let S denote the total number of accumulated discovery slots within consecutive $T_a T_b$ slots between $\mathbf{x_a}$ and $\mathbf{x_b}(l)$ as l is incremented from 0 to $T_a T_b - 1$, we have

$$S \ge \frac{N (T_a T_b)^2}{L}. \tag{1}$$

On the other hand, let i_a^h (i_b^h, respectively) denote the number of time slots in $\mathbf{x_a}$ ($\mathbf{x_b}$, respectively) in which $a(b)$ wakes up on channel h within consecutive $T_a T_b$ slots. We can express the reciprocals of the duty cycles of a and b as

$$d_a = \frac{T_a T_b}{\sum_{h \in \mathcal{N}} i_a^h}, \quad d_b = \frac{T_a T_b}{\sum_{h \in \mathcal{N}} i_b^h}.$$

After some algebraic operations, we obtain

$$T_a T_b = \sum_{h \in \mathcal{N}} d_a i_a^h = \sum_{h \in \mathcal{N}} d_b i_b^h = \sum_{h \in \mathcal{N}} \frac{d_a i_a^h + d_b i_b^h}{2}. \tag{2}$$

Since $\mathbf{x}_a$ and $\mathbf{x}_b(l)$ achieve full discovery diversity, for any channel h, the total accumulated number of discoveries between $\mathbf{x}_a$ and $\mathbf{x}_b(l)$, as l is incremented from 0 to $T_a T_b - 1$, in which the discovery channel is h, is $i_a^h \cdot i_b^h$.

Hence the total number of accumulated discoveries, as l is incremented from 0 to $T_a T_b - 1$, is $S = \sum_{h \in \mathcal{N}} i_a^h i_b^h$.

Noticing $d_a i_a^h d_b i_b^h \le \left(\frac{d_a i_a^h + d_b i_b^h}{2}\right)^2$, it follows from (2) that

$$S = \sum_{h \in \mathcal{N}} i_a^h \cdot i_b^h = \frac{\sum_{h \in \mathcal{N}} d_a i_a^h \cdot d_b i_b^h}{d_a d_b} \le \frac{(T_a T_b)^2}{d_a d_b N}.$$

Slot index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	...
Node *a* :	0	0	0	0	1	0	1	0	0	1	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	...
Node *b* :	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	...

Fig. 2. MCD in single-channel case: $d_a = 3$, $d_b = 5$.

It then follows from (1) that $\frac{N(T_aT_b)^2}{L} \leq \frac{(T_aT_b)^2}{d_ad_bN}$, which leads to $L \geq N^2 d_a d_b$. $\square$

Theorem 1 derives the performance limit of any neighbor discovery algorithm. We can further generalize Theorem 1 on the pair-wise neighbor discovery to the network-wise neighbor discovery, as stated in the following corollary.

Corollary 1: For any network where the largest two duty cycles of nodes are $\delta_1 = \frac{1}{d_1}$ and $\delta_2 = \frac{1}{d_2}$, the MTTD-FD between any pair of nodes in the network is lower-bounded by $N^2 d_1 d_2$ for any neighbor discovery protocol. Asymptotically, when $d_1 \simeq d_2 \simeq O(d)$, $L \simeq O(N^2 d^2)$.

Corollary 1 can also be viewed from another angle: to achieve a target MTTD-FD L, the duty cycle reciprocal d should be upper-bounded by $O\left(\frac{\sqrt{L}}{N}\right)$. Consequently, the energy consumption cannot be lower than $O\left(\frac{\sqrt{L}}{N}\right)$.

V. MCD: Single-Channel Case

A. Motivation and Protocol Design

In the single-channel case, the neighbor discovery schedule $\mathbf{x}_u$ for each node u degenerates to a binary sequence where

$$x_u^t = \begin{cases} 1 & u \text{ wakes up in slot } t, \\ 0 & u \text{ sleeps in slot } t. \end{cases}$$

Each node wakes up periodically to discover its neighbors. The wake-up period is determined by its duty cycle. Specifically, we consider two neighboring nodes a and b with duty cycles $\delta_a = \frac{1}{d_a}$ and $\delta_b = \frac{1}{d_b}$. To discover each other, nodes a and b wake up every d_a and d_b slots, i.e., $x_a(t) = 1$ for $t = kd_a$ and $x_b(t) = 1$ for $t = kd_b + \delta_{ab}$ where δ_{ab} is the clock offset between a and b, $k = 1, 2, \ldots$. It follows from the Chinese Remainder Theorem [21] that if d_a and d_b are co-prime to each other, the two nodes are ensured to discover each other regardless of δ_{ab}, i.e., there exists t_d such that $x_a^{t_d} = x_b^{t_d}(\delta_{ab}) = 1, \forall \delta_{ab}$.

However, assigning co-prime numbers to each node in a distributed way is far from trivial. A commonly adopted solution is to use only prime numbers because two distinct prime numbers are by definition co-prime to each other, as in Disco [8] and U-Connect [14]. However, limiting the choices to prime numbers fail to support all the duty cycles due to the limited number of prime numbers. Note that among natural numbers smaller than 1000, only $\frac{1}{6}$ are prime numbers.

Motivated by the above analysis, we devise the following neighbor discovery schedule in MCD. For each node u with duty cycle $\delta_u = \frac{1}{d_u}$,

$$x_u(t) = \begin{cases} 1 & t \text{ is divisible by either } 2d_u - 1 \text{ or } 2d_u + 1, \\ 0 & \text{otherwise.} \end{cases}$$

Example 2: Consider two nodes a and b with duty cycles $\delta_a = \frac{1}{3}$, $\delta_b = \frac{1}{5}$ with a clock offset $\delta_{ab} = 1$. Under MCD, using the time of a as reference, a wakes up in slots $5k$ and $7k$, i.e., $5, 7, 10, 14, 15, 20, 21, \ldots$, b wakes up in slots $9k + 1$ and $11k + 1$, i.e., $10, 12, 19, 23, \ldots$, as illustrated in Fig. 2. The discovery happens in slot 10.

The period of $\mathbf{x}_u$ in MCD is $(2d_u - 1)(2d_u + 1)$, in which there are $4d_u - 1$ active slots.[3] Hence, the actual average duty cycle, denoted as $\widehat{\delta}_u$, is $\frac{4d_u - 1}{(2d_u-1)(2d_u+1)}$ which approaches to the required duty cycle $\delta_u = \frac{1}{d_u}$ when d_u is large. Generally, the relative error between $\widehat{\delta}_u$ and δ_u is upper-bounded by $\frac{1}{4d_u}$, as established in the following lemma.

Lemma 1: The relative error between the duty cycle of the neighbor discovery schedule $\widehat{\delta}_u$ and the required duty cycle δ_u is upper-bounded by $\frac{1}{4d_u}$.

Proof: Denote the relative error between $\widehat{\delta}_u$ and δ_u as ϵ, noticing that $d_u = \frac{1}{\delta_u} \geq 1$, we have:

$$\begin{aligned} \epsilon &= \left| \frac{4d_u - 1}{(2d_u - 1)(2d_u + 1)} - \frac{1}{d_u} \right| \Big/ \frac{1}{d_u} = \frac{d_u - 1}{4d_u^2 - 1} \\ &= \frac{d_u - 1}{4d_u(d_u - 1) + 4d_u - 1} < \frac{1}{4d_u}. \end{aligned}$$

$\square$

B. MCD Core Idea: Regular Duty Cycles

Following the Chinese Remainder Theorem, the mutual discovery of two neighbor nodes a and b in MCD, regardless of their clock drift, requires at least one of $2d_a \pm 1$ to be co-prime with at least one of $2d_b \pm 1$. In the vast majority of cases, this requirement can be satisfied. To illustrate this, if we allow the maximum duty cycle reciprocal D to be 100, then all duty cycles $\frac{1}{d}$ except $d = 17$ and 38 can be supported by MCD; if we allow D to be 1000, only 43 duty cycles cannot be supported, i.e., MCD can support nearly 96% of all duty cycles.

In this subsection, we conduct a formal analysis on the design idea of MCD. We start by formulating the definition of *regular duty cycles* that are natively supported by MCD.

Definition 3 (Regular Duty Cycle): Given the duty cycle reciprocal upper-bound D, we call a duty cycle $\delta = \frac{1}{d} (d \leq D)$ a regular duty cycle if for any $2 \leq d' \leq D$, at least one number from $2d \pm 1$ is co-prime with at least one number from $2d' \pm 1$.

For two nodes a and b, if at least one of their duty cycles is regular, a and b can discover each other. Reconsider Example 2 with $D = 1000$, it follows from Definition 3 that the duty cycles $\delta_a = \frac{1}{3}$ and $\delta_b = \frac{1}{5}$ are both regular. Evidently the two nodes can discover each other, as illustrated in Fig. 2.

We conclude this subsection by stating the following properties of regular duty cycles:

- *The vast majority of duty cycles are regular*. As illustrated in Table I, more than 97% duty cycles are regular when D

[3] Note that $(2d_u - 1)(2d_u + 1)$ is divisible by both $2d_u - 1$ and $2d_u + 1$.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
NUMBER OF NON-REGULAR DUTY CYCLES AS A FUNCTION OF D.

Duty cycle reciprocal upper-bound D	100	300	500	700	900
Number of non-regular duty cycles	2	11	20	29	38

varies from 100 to 900. In contrast, in existing solutions based on prime numbers, only a small portion of duty cycles can be supported due to the limited choice of prime numbers.

- *There are no three consecutive non-regular duty cycle reciprocals*. In other words, if $\frac{1}{d}$ is non-regular, at least one from $\frac{1}{d\pm 1}$ is regular. This implies that if the required duty cycle $\frac{1}{d}$ happens to be non-regular, the node can operate on $\frac{1}{d+1}$ or $\frac{1}{d-1}$. If we take the case of using $\frac{1}{d-1}$, the effective duty cycle is $\frac{4(d-1)-1}{[2(d-1)-1][2(d-1)+1]}$. The relative error to the required duty cycle can be computed as

$$\begin{aligned}\epsilon &= \left|\frac{4(d-1)-1}{[2(d-1)-1][2(d-1)+1]} - \frac{1}{d}\right| / \frac{1}{d}\\ &= \frac{4(d_u-1)+1}{4(d_u-1)^2-1},\end{aligned}$$

which is decreasing in d. In the case of $D = 1000$, the smallest non-regular duty cycle reciprocal being 17, ϵ is upper-bounded by 4.5% for all non-regular duty cycles.

The analysis in this subsection demonstrates that MCD can support most duty cycles and even in the cases where the duty cycles cannot be directed supported, MCD can use neighboring duty cycles with almost negligible errors.

Concerning the implementation of MCD, we would like to emphasize that the regular duty cycles can be pre-calculated offline by exhaustive search and stored in a look-up table.

C. Discovery Delay Upper-Bound

In the single-channel case, only the first performance metric (maximum time to discovery, MTTD) is applicable. In Theorem 2, we derive the MTTD of MCD between two nodes a and b if at least one of δ_a and δ_b is regular.[4]

Theorem 2 (Discovery Delay Upper-Bound): Given any two nodes a and b, if at least one of their duty cycles $\frac{1}{d_a}$ and $\frac{1}{d_b}$ is regular, they are ensured to discover each other within at most $(2d_a+1)(2d_b+1)$ slots.

Proof: Recall the definition of the regular duty cycle, at least one of $2d_a \pm 1$ is co-prime with at least one of $2d_b \pm 1$. It follows from The Chinese Remainder Theorem that a and b can discover each other within at most $(2d_a+1)(2d_b+1)$ slots, regardless of their clock offset. □

D. MCD Optimization: Supporting More Duty Cycles

From previous analysis, we can see that using non-regular duty cycles may result in discovery failure. For example, if the duty cycle reciprocal upper-bound is $D = 100$, node a with duty cycle $\frac{1}{d_a} = \frac{1}{17}$ may never discover node b with duty cycle $\frac{1}{d_b} = \frac{1}{38}$ because neither $2d_a \pm 1$ (33 and 35) is co-prime to either $2d_b - 1$ (75) or $2d_b + 1$ (77). A direct solution is to remove $\frac{1}{17}$ and $\frac{1}{38}$ from the usable duty cycle set and use $\frac{1}{16}$ and $\frac{1}{37}$ instead. However, since one of d_a and d_b being regular is sufficient to ensure discovery, we can do better by removing only $\frac{1}{38}$ from the set of usable duty cycles. Now $\frac{1}{17}$ becomes usable as for any duty cycles other than $\frac{1}{38}$, discovery is guaranteed. In this subsection, we explore the natural question of constructing the usable duty cycle set with the maximum number of elements, formalized as follows.

[4]Throughout our analysis, we focus on the pair-wise discovery between any pair of neighbor nodes a and b. The obtained results can be readily generated to the network level where each node should discover all its neighbor nodes by following the same way as Theorem 1 and Corollary 1.

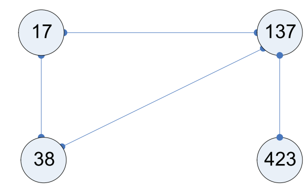


Fig. 3. Duty cycle graph (part).

Problem 2: Let U denote the usable duty cycle set with the duty cycle reciprocal upper-bound D,

$$\begin{aligned}&maximize && |U|\\ &subject\ to && \forall \frac{1}{d_1}, \frac{1}{d_2} \in U, d_1 \neq d_2, at\ least\ one\ of\ 2d_1 \pm 1\\ &&& is\ co\text{-}prime\ with\ at\ least\ one\ of\ 2d_2 \pm 1.\end{aligned}$$

Solving Problem 2: A Graph-Based Approach: We address Problem 2 by transforming it to a problem on a graph. Specifically, we construct a graph in which each vertex represents a duty cycle reciprocal and there exists an edge between two vertexes if the duty cycle reciprocals represented by the two vertexes may fail to discover each other (mathematically, neither $2d_a \pm 1$ is co-prime to either $2d_b - 1$ or $2d_b + 1$). We observe that the duty cycle graph typically consists of a number of non-connected clusters. Fig. 3 illustrates one of such cluster for $D = 1000$. We seek to remove the minimal number of vertexes (and the edges connected to them) such that each of the remaining vertex is isolated, meaning that the remaining vertexes represent the usable duty cycles. In the cluster of Fig. 3, we need to remove at least two nodes, e.g., 38 and 137.

To solve Problem 2, we find the maximum independent sets (MaxIS) [9] in the duty cycle graph. An independent set (IS) of a graph is a set of vertices, no two of which are adjacent. That is, it is a set $\mathcal{I}$ of vertices such that for any two vertices in $\mathcal{I}$, there is no edge connecting them. Equivalently, each edge in the graph has at most one endpoint in $\mathcal{I}$. A MaxIS is an IS with maximum cardinality, i.e., contains the maximum number of vertices. Consider Fig. 3, an MaxIS is {17, 423}.

Given its NP-hardness of finding an MaxIS, we develop a heuristic *polynomial-time* algorithm (Algorithm 1) to solve Problem 2 based on the observation that the duty cycle graph is only loosely connected and that the maximum degree of the graph is limited (typically no more than 3). The heuristic algorithm consists of iteratively adding the vertex with the smallest degree and removing the edges and vertexes connected to it until when the graph becomes empty.

Theorem 3 ($\frac{1}{\Delta+1}$-Optimality of Algo 1): Algo. 1 gives a $\frac{1}{\Delta+1}$-approximation for the maximum usable duty cycle set in a duty cycle graph with the maximum degree Δ.

Algorithm 1 Calculate the heuristic usable duty cycle set

Input: Duty cycle graph G
Output: Usable duty cycle set U
  Initialization: $U \leftarrow \emptyset$
  while G is not empty **do**
    Find a vertex v of minimum degree in G
    $U \leftarrow U \cup \{v\}$
    Remove v and its neighbors from G
  end while

Proof: We show that the output of Algorithm 1 U satisfies $|U| \geq \frac{|U^*|}{\Delta+1}$ where U^* is the maximum usable duty cycle set. To this end, we upper-bound the number of vertexes in $V \setminus U$. It follows from Algorithm 1 that a vertex u is in $V \setminus U$ because it is removed as a neighbor of some node $v \in U$ when v is added to U. Since any vertex v has at most Δ neighbors, it holds that $|V \setminus U| \leq \Delta|U|$. Hence, we have $|U| \geq \frac{|V|}{\Delta+1} \geq \frac{|U^*|}{\Delta+1}$ by noticing that U^* is a subset of V. □

Though Algo. 1 provides a polynomial-time algorithm that solves Problem 2 with constant-factor approximation, it fails to find the optimum solution due to the NP-completeness of finding MaxIS in generic graphs. Motivated by this argument, in the rest of this section we demonstrate that due to the particularity of Problem 2 in practical scenarios, we can find the exact optimum in linear time.

Our analysis is based on the following observation on a structural property of the duty cycle graph:

In the duty cycle graph under practical settings, there does not exist any path longer than 4.[5]

By practical settings, we mean a reasonable duty cycle reciprocal upper-bound D. Specifically, we have tested that the above observation holds for $D = 10000$, which, to our knowledge, covers the vast majority of mobile sensing applications. In the following theorem, we prove that under such parameter settings, Problem 2 can be solved exactly in linear time.

Theorem 4: Problem 2 can be solved exactly in linear time under practical parameter settings ($D = 10000$).

Proof: The proof employs the concept of outerplanar graphs and the structural properties of an outerplanar graph, as introduced in the following.

Definition 4 (Outplaner Graphs [6]): A graph is called outerplanar if it has a drawing in which every vertex lies on the boundary of the outer face.

The following lemma gives a necessary and sufficient condition for a graph to be outerplaner.

Lemma 2 ([24]): A graph is outerplanar if and only if it contains neither K_4 nor $K_{2,3}$ as a minor.

The following lemma states that a MaxIS can be found in linear time for outerplanar graphs.

Lemma 3 ([3]): The maximum independent set problem can be solved in linear time for outerplanar graphs.

To prove Theorem 4, we show that the duty cycle graphs under practical parameter settings are outerplaner graphs. In this regard, recall the observation that in duty cycle graphs under practical settings there does not exist any path longer than 4, we have that any duty cycle graph does not contain K_4 nor $K_{2,3}$ as a minor. It then follows from Lemma 2 that any duty cycle graph is outerplanar. It then follows from Lemma 3 that an MaxIS of any duty cycle graph can be found in linear time, meaning that Problem 2 can be solved exactly in linear time. □

[5]In case of a cycle, each node is counted only once in the path.

TABLE II
LOOK-UP TABLES OF NON-USABLE DUTY CYCLE RECIPROCALS IN MCD BEFORE AND AFTER OPTIMIZATION.

Reciprocals of non-supported duty cycles before optimisation	17, 32, 38, 43, 77, 103, 137, 143, 162, 178, 218, 241, 247, 248, 263 302, 332, 333, 347, 368, 389, 423, 437, 472, 487, 500
Reciprocals of non-supported duty cycles after optimisation	38, 137, 241, 247, 248, 263, 332, 333, 368, 389, 472, 487

Note that both Algo. 1 and the algorithm of finding an MaxIS for outerplaner graphs can be executed off-line to generate a look-up table that contains all non-supported duty cycles in MCD. Each node only need to check if its required duty cycle is in the table each time when it needs to set/reset the MCD parameters.

By these algorithms, we can typically reduce the non-supported duty cycles by more than 50%. For example, for $D = 100$, we can support all duty cycles except $\frac{1}{38}$; for $D = 500$, the number of duty cycles that cannot be supported by MCD reduces from 26 to 12, i.e., less than 2.5% of the total duty cycles; for $D = 1000$, the same number reduces from 43 to only 18, i.e., less than 2% of the total duty cycles. As an illustrative example, Table II compares the number of non-supported duty cycles in the look-up table before and after optimization.

VI. MCD: MULTI-CHANNEL CASE

A. Neighbor Discovery Schedule Construction

The neighbor discovery schedule of MCD for each node in the multi-channel case is constructed based on its globally unique ID such as its MAC address, which can be mathematically expressed as a binary sequence of length l. Using globally unique IDs is a typical method to break the symmetry of any pair of nodes. The neighbor discovery schedule construction process is composed of three steps, summarized here and detailed in the following analysis.

- *Step* 1: Each node u independently generates a padded binary sequence $\mathbf{o}_u$ based on its ID such that the padded binary sequences of any two nodes are *cyclic rotationally distinct* one to the other;
- *Step* 2: Each node u independently generates a sequence $\mathbf{s}_u$ based on $\mathbf{o}_u$ such that for any two nodes a, b and any initial time offset t_a^0 and t_b^0, there always exist four time slots $l_{ij}(i, j \in \{0, 1\})$ such that $s_a^{l_{ij}}(t_a^0) = i$ and $s_b^{l_{ij}}(t_b) = j$. We denote such sequences $\mathbf{s}_u$ as *regular sequences*;
- *Step* 3: Each node u generates its neighbor discovery schedule based on $\mathbf{s}_u$.

Step 1: Constructing Cyclic Rotationally Distinct Padded Binary Sequence: In the first step, each node independently generates a binary sequence based on its ID such that the binary sequences of any two nodes are *cyclic rotationally distinct* one

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

to the other. Note that the sequences resulting from cyclic rotations of a sequence are not considered to be cyclic rotationally distinct with respect to each other and the original sequence. We next show how to construct such cyclic rotationally distinct binary sequences.

Let α denote the ID of a node a. For an integer $r \in [2, l]$, let $l' \triangleq \frac{l}{r}$.[6] We divide α into r subsequences of length l', denoted by $\alpha_1, \alpha_2, \ldots, \alpha_r$. Let $\mathbf{0}$ denote a sequence of 0 of length $l'+1$. We construct the padded ID of a, denoted by $\mathbf{a}$, as follows:

$$\mathbf{a} = \alpha_1 ||1|| \alpha_2 ||1|| \cdots \alpha_r ||1|| \mathbf{0} ||1,$$

where $||$ denotes the operation of concatenation. By the following lemma, we show that the padded ID sequences generated in such way based on different ID sequences are cyclic rotationally distinct one to another.

Lemma 4: Given any two padded ID sequences $\mathbf{a}$ and $\mathbf{b}$ generated from two ID sequences α and β, it holds that

$$\alpha \neq \beta \quad \Longrightarrow \quad \mathbf{a} \neq \mathbf{b}(k),\ \forall k \in [1, L_o - 1],$$

where $\mathbf{b}(k)$ is $\mathbf{b}$ with a cyclic rotation of k bits, $L_o = l + l' + r + 2$ is the length of padded ID sequences.

Proof: The theorem follows from the construction of the padded ID sequences:

- It follows from the construction of $\mathbf{a}$ that $\forall k_a \in [L_o - l', L_o - 1]$, it holds that $a_{k_a} = 0$.
- It follows from the construction of $\mathbf{b}$ that $\forall k \in [0, L_o - 1]$, it holds that $b_{L_o - k_r}(k) = 1$, where $k_r = k \bmod l'$.

It then follows that

$$b_{L_o - k_r}(k) = 1 \neq a_{L_o - k_r} = 0,$$

which completes the proof. □

Lemma 4 holds for any value of r. In the context of neighbor discovery, we are particularly interested in seeking the expanded ID sequences with minimum length. In this regard, noticing that

$$L_o = l + l' + r + 2 = l + \frac{l}{r} + r + 2,$$

it holds that L_o is minimized when $r = \sqrt{l}$ with the minimum value $L_o^{\min} \simeq l + O(\sqrt{l})$.

Step 2: Generating Regular Sequence: Denote the padded ID sequence as $\mathbf{o}_u$ for node u, the next step for each node is to generate a sequence $\mathbf{s}_u$ based on $\mathbf{o}_u$ such that for any two nodes a, b and any initial time offset t_a^0 and t_b^0, there always exist four time slots $l_{ij}(i, j \in \{0, 1\})$ such that $s_a^{l_{ij}}(t_a^0) = i$ and $s_b^{l_{ij}}(t_b) = j$. We denote such sequences $\mathbf{s}_u$ as *regular sequences*. In the following we develop an algorithm to generate regular sequences.

Lemma 5: The sequence generated by Algo 2 is regular.

Proof: It suffices to show that for any $k \in [0, 8L_o - 1]$, there always exist four time slots $l_{ij}(i, j \in \{0, 1\})$ such that $s_a^{l_{ij}} = i$ and $s_b^{l_{ij}}(k) = j$. Let $k = 8k_1 + k_2$ where $k_1 \triangleq \lceil \frac{k}{8} \rceil - 1 \in [0, L_o - 1]$ and $k_2 \triangleq k \bmod 8 \in [0, 7]$, we distinguish the following two cases.

- *Case 1:* $k_2 \leq 3$. Let $\mathbf{o}_a$ and $\mathbf{o}_b$ denote the padded ID sequences of a and b. Recall Lemma 4 and the notations in Section III, there exists t^* such that $o_a^{t^*} \neq o_b^{t^*}(k)$. Without loss of generality, assume that $o_a^{t^*} = 1$ and $o_b^{t^*}(k) = 0$. It follows from Algorithm 2 that the eight bits of $\mathbf{s_a}$ starting from $s_a^{8t^*+1}$ are 01010101 and the eight bits of $\mathbf{s_b}(k)$ starting from $s_b^{8t^*-k_2+1}(k)$ are 00001111. It follows that when $k_2 \leq 3$, there always exist four time slots $l_{ij}(i, j \in \{0, 1\})$ such that $s_a^{l_{ij}} = i$ and $s_b^{l_{ij}}(k) = j$. Fig. 4 illustrates the case with $k_2 = 1$.
- *Case 2:* $k_2 \geq 4$. Recall Lemma 4, there exists t^* such that $o_a^{t^*} \neq o_b^{t^*+1}(k)$. Without loss of generality, assume that $o_a^{t^*} = 1$ and $o_b^{t^*+1}(k_1) = 0$. It follows from Algorithm 2 that the eight bits of $\mathbf{s_a}$ starting from $s_a^{8t^*+1}$ are 01010101 and the eight bits of $\mathbf{s_b}(k)$ starting from $s_b^{8(t^*+1)-k_2+1}(k)$ are 00001111. It follows that when $k_2 \geq 4$, there always exist four time slots $l_{ij}(i, j \in \{0, 1\})$ such that $s_a^{l_{ij}} = i$ and $s_b^{l_{ij}}(k) = j$. Fig. 4 illustrates the case with $k_2 = 6$.

The sequence generated by Algo. 2 is thus regular. □

Algorithm 2 Construct a regular sequence $\mathbf{s}_u$

Input: ID sequence $\mathbf{o}_u$ of L_o bits
Output: Regular sequence $\mathbf{s}_u$
  for $i = 1$ to L_o **do**
    switch o_u^i **do**
      case 1: expand o_u^i into eight bits 01010101
      case 0: expand o_u^i into eight bits 00110011
    end switch
  end for
  $\mathbf{s}_u \leftarrow$ the expanded sequence of $\mathbf{o}_u$

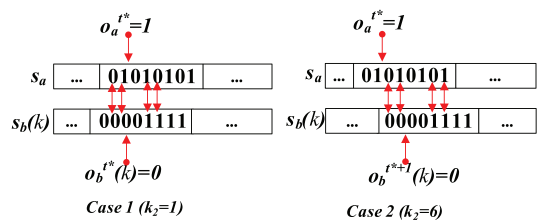


Fig. 4. Illustration of the two cases in the proof of Lemma 5.

Step 3: Generating Neighbor Discovery Schedule: In the last step, the neighbor discovery schedule is constructed as follows. Each node u hops across different channels $h \in \mathcal{N}$ and wakes up based on the following schedule[7]:

$$x_u^t = \begin{cases} h & t - hd_u \text{ is divisible by } 2Nd_u \pm 1, \\ 0 & \text{otherwise,} \end{cases}$$

where $x_u^t = h$ signifies that u wakes up on channel h in slot t while $x_u^t = 0$ indicates that u sleeps in the slot, $\frac{1}{Nd_u}$ is chosen from the usable duty cycle set as analyzed in Section V-D.

The above construction of $\mathbf{x}_u$ does not take into account the case where there exist two different channels $h^c (c = 0, 1)$ such that $t - h^0 d_u$ is divisible by $2Nd_u - 1$ and $t - h^1 d_u$ by $2Nd_u + 1$. To resolve such conflict, let $t' = t \bmod L_s$, u operates on channel

[6]To better present the idea, we assume that l is divisible by r. However, our analysis can be easily extended to the case where l is not divisible by r.

[7]To make the notation concise, we adopt the notation that $t - hd_u$ is divisible by $2Nd_u \pm 1$ denotes that $t - hd_u$ is divisible by $2Nd_u - 1$ or $2Nd_u + 1$ or both.

h^c if $s_u^{t'} = c$. We refer to the slots where u operates on channel h^c in case of conflict as type-c slots.

To intuitively see that the discovery is ensured between any pair of nodes a, b (the detailed proof is presented in the next subsection), note that if $\frac{1}{Nd_u}$ belongs to the usable duty cycle set derived previously, i.e., at least one of $2Nd_a \pm 1$ is co-prime with at least one of $2Nd_b \pm 1$, discovery can be guaranteed for any initial time offset t_a^0 and t_b^0 because there always exist four time slots $l_{ij}(i, j \in \{0, 1\})$ such that $s_a^{l_{ij}}(t_a^0) = i$ and $s_b^{l_{ij}}(t_b) = j$ following the regularity of $\mathbf{s}_a$ and $\mathbf{s}_a$.

B. Discovery Delay Upper-Bound

This subsection studies the theoretical performance of MCD in the multi-channel environment. In multi-channel case, the second metric on discovery diversity and the third metric on MTTD-FD (worst-case discovery delay with full diversity) are applicable.

Theorem 5: If $\frac{1}{Nd_a}$ and $\frac{1}{Nd_b}$ belong to the usable duty cycle set, the MTTD-FD between two nodes a and b is $O(L_s N^2 \max\{d_a^2, d_b^2\})$, where L_s denotes the length of regular sequences, specifically, $O(L_s N^2 d^2)$ if $d_a \simeq d_b \simeq O(d)$.

Proof: Without loss of generality, assume that $2Nd_a + 1$ is co-prime with $2Nd_b - 1$. It follows from The Chinese Remainder Theorem that before resolving conflicts (i.e., assume each node can operate on two channels simultaneously), for any channel h, there exists $t_0 < (2Nd_a + 1)(2Nd_b - 1)$ such that $x_a^{t_0}(t_a^0) = x_b^{t_0}(t_b^0) = h$ and it holds that on slots $t_k = t_0 + k(2Nd_a + 1)(2Nd_b - 1)$, a can also discovery b before resolving conflicts. However, in realistic settings with conflicts, to ensure discovery, we need to show that there exists k such that a operates in type-1 slot at slot t_k while b operates in type-0 slot at slot t_k. To that end, noticing that $L_s = 8L_o$ is an even number and thus is co-prime with $(2Nd_a + 1)(2Nd_b - 1)$, there must exist $k < L_s$ such that $t_k \mod L_s = l_0$, where l_0 denoted the bit index such that in $s_a^{l_0}(t_a^0) = 1$ and $s_b^{l_0}(t_b^0) = 0$. It follows from Lemma 5 that such l_0 exists. It follows from the construction of x_u that a operates in type-1 slot at slot t_k while b operates in type-0 slot at slot t_k, which leads to discovery. It further follows from $k < L_s$ that the MTTD-FD is $O(L_s N^2 \max\{d_a^2, d_b^2\})$. □

The capability to achieve discovery on every channel within bounded delay significantly improves neighbor discovery robustness in wireless environment where channel conditions are unpredictable and may vary in both time and space.

C. Robustness Against Asymmetrical Channel Perception

In previous analysis, we implicitly assume that a and b have the same channel perception, i.e., they have symmetrical knowledge on $\mathcal{N}$. In this subsection, we relax this assumption to investigate the scenario where each node u has its own perception on $\mathcal{N}$, denoted by $\mathcal{N}_u$, which is a subset of $\mathcal{N}$. In this context, the neighbor discovery schedule in MCD becomes

$$x_u^t = \begin{cases} h & t - hd_u \text{ is divisible by } 2N_u d_u \pm 1, \\ 0 & \text{otherwise.} \end{cases}$$

Specifically, the channel perception asymmetry between a and b can be characterized at two levels:

- *Asymmetry on accessible channel set:* They have asymmetrical perceptions on the global channel set $\mathcal{N}$, i.e., $\mathcal{N}_a \neq \mathcal{N}_b$ and $\mathcal{N}_a \bigcap \mathcal{N}_b \neq \emptyset$;
- *Asymmetry on channel index:* They have asymmetrical perceptions on the channel index, i.e., channel $h \in \mathcal{N}$ is indexed h_a by a and h_b by b where $h_a \in \mathcal{N}_a$ and $h_b \in \mathcal{N}_b$ but $h_a \neq h_b$.

The following theorem established the performance of MCD in such context.

Theorem 6: MCD under asymmetrical channel perceptions achieves the same MTTD-FD as under symmetrical channel perceptions, i.e., within at most $O(L_s \max\{N_a^2 d_a^2, N_b^2 d_b^2\})$ (specifically, $O(L_s N^2 d^2)$ if $d_a \simeq d_b \simeq O(d)$ and $N_a \simeq N_b \simeq O(N)$) slots, the discovery between a and b occurs on each channel $h \in \mathcal{N}_a \bigcap \mathcal{N}_b$.

Proof: Without loss of generality, assume that $2N_a d_a + 1$ is co-prime with $2N_b d_b - 1$. It follows from The Chinese Remainder Theorem that before resolving conflicts (i.e., assume u can operate on two channels), for any channel h indexed as h_a (h_b) by a (b), there exists $t_0 < (2N_a d_a + 1)(2N_b d_b - 1)$ such that $x_a^{t_0}(t_a^0) = h_a$ and $x_b^{t_0}(t_b^0) = h_b$. Then using the similar analysis as the proof of Theorem 5, we can show that the MTTD-FD is $O(L_o \max\{N_a^2 d_a^2, N_b^2 d_b^2\})$. □

Theorem 6 shows that MCD is robust against asymmetrical channel perceptions, either on the channel set or index.

D. Robustness Against Slot Non-Alignment and Arbitrary Clock Drift

In this subsection we study the effect of slot non-alignment caused by relative clock drift between the neighbor nodes.

We first briefly introduce the clock model. Each node is equipped with a local clock, which is a time measurement device composed of a hardware oscillator and an accumulator. Mathematically, consider two nodes a and b, we can express the local time at b, denoted as t_b, as a function of the local time of a, denoted as t_a, by the following formula

$$t_b(t_a) = \int_{t_0}^{t_a} \rho_{ab}(\tau) d\tau + t_b(t_0),$$

where $\rho_{ab}(\tau)$ denotes the relative frequency drift rate of the oscillator of b as a function of a at time τ, $t_b(t_0)$ is the initial clock offset between them.

If a and b are ideally synchronized, it holds that $\rho_{ab}(\tau) = 1$ and $t_b(t_0) = 0$. In practice, $\rho_{ab}(\tau)$ may drift away from each other, as formalized in the following:

$$\rho_{ab} - \Delta\rho_{\max} \leq \rho_{ab}(\tau) \leq \rho_{ab} + \Delta\rho_{\max},$$

where $\Delta\rho_{\max}$ is bounded by 10^{-6} in practice. Hence we can regard $\rho_{ab}(\tau)$ as a constant ρ_{ab} during the discovery process. Without loss of generality, we assume that the clock of b advances no slower than that of a, i.e., $\rho_{ab} \leq 1$.

When $\rho_{ab} = 1$, i.e., the clock difference between a and b remains $t_b(t_0)$, we distinguish the following two cases (to facilitate presentation, we normalize the slot duration of a):

- Case 1: $t_b(t_0) = k \in \mathbb{Z}$: this is the case with aligned slots addressed in previous analysis;

- Case 2: $t_b(t_0) = k + \delta$ with $k \in \mathbb{Z}$ and $\delta \in (-1/2, 1/2]$: the previous analysis can be directly adapted to this case, the difference being that instead of ensuring entire overlap, a discovery in this case is a partial overlap of time $1 - \delta$.

We now investigate the case where $\rho_{ab} < 1$, meaning that if we regard the slot duration of a as unit time, the slot duration of b is $\rho_{ab} < 1$. We first establish the following property that is useful in the later analysis.

Lemma 6: Given any $\rho_{ab} \in \mathbb{R}$, let $p(n)$ denote the nth prime number, it holds that

$$\lim_{n\to\infty} \frac{p(\lfloor \rho_{ab} n \rfloor)}{p(n)} = \rho_{ab}.$$

That is, ρ_{ab} can be well approximated by $p(\lfloor \rho_{ab} n \rfloor)/p(n)$.

Proof: Recall the prime number theorem (PNT) [12] that $\lim_{n\to\infty} p(n) = n \ln(n)$, we have

$$\begin{aligned}\lim_{n\to\infty} \frac{p(\lfloor \rho_{ab} n \rfloor)}{p(n)} &= \frac{\rho_{ab} n \ln(\rho_{ab} n)}{n \ln(n)} \\ &= \lim_{n\to\infty} \frac{\rho_{ab} n \ln(\rho_{ab}) + \rho_{ab} n \ln(n)}{n \ln(n)} = \rho_{ab},\end{aligned}$$

which completes the proof. □

The following theorem establishes the discovery performance of MCD with arbitrary clock drift with $\rho_{ab} < 1$.

Theorem 7: Regard the slot of a as unit time, a and b can discover each other on each channel h within at most $O(\rho_{ab} L_s N^2 \max\{d_a^2, d_b^2\})$ time.

Proof: We decompose each slot of nodes a and b into mini-slots of duration $\frac{1}{p(n)}$ where n is sufficiently large. Under the decomposition, each slot of a and b contain $p(n)$ and $p(\lfloor \rho_{ab} n \rfloor)$ mini-slots, respectively. We can express the neighbor discovery schedules of a and b in mini-slots.

$$x_a^t = \begin{cases} h & t - h d_a p(n) + l_a \text{ is divisible by} \\ & (2Nd_a \pm 1)p(n), 0 \le l_a \le p(n) - 1, \\ 0 & \text{otherwise.} \end{cases}$$
$$x_b^t = \begin{cases} h & t - h d_b p(\lfloor \rho_{ab} n \rfloor) + l_b \text{ is divisible by} \\ & (2Nd_b \pm 1)p(\lfloor \rho_{ab} n \rfloor), 0 \le l_b \le p(n) - 1, \\ 0 & \text{otherwise.} \end{cases}$$

The conflicts are resolved in the same as in previous analysis by attributing sequence $\mathbf{s}_u$ to u. If both $\frac{1}{Nd_a}$ and $\frac{1}{Nd_b}$ are usable duty cycles, as required in previous analysis, noticing that $p(n)$ and $p(\lfloor \rho_{ab} n \rfloor)$ are sufficiently large prime numbers (specifically, larger than $2Nd_a + 1$ and $2Nd_b + 1$), at least one of $(2Nd_a \pm 1)p(n)$ must be co-prime with at least one of $(2Nd_b \pm 1)p(\lfloor \rho_{ab} n \rfloor)$. By the same analysis as that in Theorem 5, for any channel h, we can show that within at most $O(L_o N^2 p(n) p(\lfloor \rho_{ab} n \rfloor) \max\{d_a^2, d_b^2\})$ mini-slots (i.e., $O(\rho_{ab} L_o N^2 \max\{d_a^2, d_b^2\})$ time by regarding the slot of a as unit time), there exist a mini-slot t_m such that $t_m^a - h d_a p(n)$ is divisible by $(2Nd_a \pm 1)p(n)$ and $t_m^b - h d_b p(\lfloor \rho_{ab} n \rfloor)$ is divisible by $(2Nd_b \pm 1)p(\lfloor \rho_{ab} n \rfloor)$ and both a and b operating on channel h, where t_m^a and t_m^b denotes the local time at a and b for time t_m. It then follows that from mini-slot t_m to $t_m + p(\lfloor \rho_{ab} n \rfloor)$ (i.e., a slot for b), both a and b wake up on channel h, and can thus discover each other on h. □

TABLE III
COMPARISON OF MCD AND MAJOR EXISTING DETERMINISTIC NEIGHBOR DISCOVERY PROTOCOLS: $p_u p_u^1$ AND p_u^2 ARE PRIMES; t_u, i AND d_u ARE NATURAL NUMBERS.

Nbr discovery protocol	Worst-case discovery delay	Duty cycles supported	Support multiple channels
Disco	$O(d_a d_b)$	$\frac{p_u^1 + p_u^2}{p_u^1 p_u^2}$	No
U-Connect	$O(d_a d_b)$	$\frac{3p_u + 1}{2p_u^2}$	No
Searchlight	$O(d_a d_b)$	$\frac{2}{t_u^i}$	No
MCD	$O(L_o N^2 d_a d_b)$	$\frac{4d_u - 1}{(2d_u+1)(2d_u-1)}$	Yes

The results obtained in this subsection, particularly Theorem 7, demonstrate that the discovery performance established in previous analysis holds even when the clocks of a and b drift away from each other for an arbitrary amount of time. In other words, MCD is robust against clock drift and slot non-alignment.

VII. Performance Evaluation

To evaluate the performance of MCD, we first conduct a comparative analysis between MCD and major existing deterministic neighbor discovery protocols. We then perform a serious of simulations evaluate MCD in several typical application scenarios ranging from the synchronized single-channel case to the heterogeneous asynchronous and asymmetrical multi-channel case.

A. Performance Comparison

Table III compares MCD with major existing deterministic neighbor discovery protocols Disco [8], U-Connect [14] and Searchlight [4] in terms of discovery delay, duty cycle granularity and multi-channel support.

- Disco uses two primes p_u^1 and p_u^2 for each node u. The supported duty cycles are thus restricted to $\frac{p_u^1 + p_u^2}{p_u^1 p_u^2}$. The worst-case discovery delay between a and b is $\min_{i,j\in\{0,1\}}\{p_a^i p_b^j\}$.
- In U-Connect, each node u chooses a prime number p_u and wakes up one slot every p_u slots and also $\frac{p_u+1}{2}$ slots every p_u^2 slots. The supported duty cycles are in the form of $\frac{3p_u+1}{2p_u^2}$. The worst-case discovery delay between a and b is $p_a p_b$.
- Searchlight uses a parameter t_u for each node u where t_u must be a power-multiple of the smallest chosen number (e.g., $2, 4, 8, 16, \ldots$, or $3, 9, 27, 81, \ldots$), guaranteeing that any two nodes' numbers are multiples of each other. The supported duty cycles are in the form of $\frac{2}{t_u^i} (i \ge 1)$. The worst-case discovery delay between a and b is $O(t_a t_b)$.
- MCD supports all duty cycles in the usable duty cycle set, which contains the vast majority (more than 96%) of all duty cycles in practical settings. The worst-case discovery delay is $O(d_a d_b)$ in the single-channel case and $O(L_s N^2 d_a d_b)$ in the multi-channel case. Among the major deterministic neighbor discovery protocols, MCD is the only protocol that has multi-channel support.

We can notice that all of the four protocols achieve the same order of worst-case discovery delay. However, Disco, U-Connext and Searchlight only support the single-channel case and a subset of duty cycles, i.e., they can only support a limited

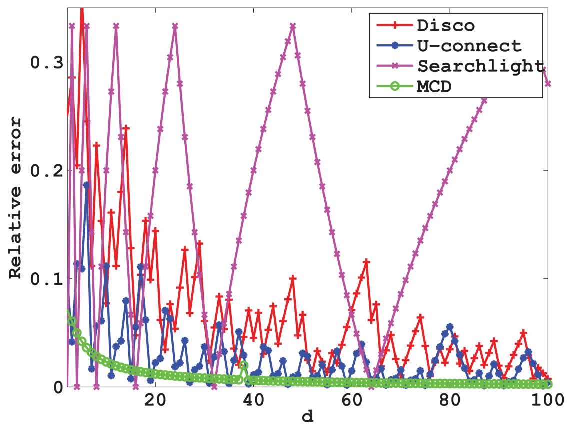


Fig. 5. Comparison between MCD and major neighbor discovery protocols: relative error as function of d.

choices of energy conservation levels. In contrast, MCD, can work in the multi-channel case and support almost all duty cycles in practical settings.

B. Supported Duty Cycles

The first numerical experiment is a comparative analysis on the supported duty cycles in Disco, U-Connect, Searchlight and MCD. To that end, for each possible required duty cycle $\delta = \frac{1}{d}$ with $1 \leq d \leq 100$, we study the relative error in supporting it, denoted as $\epsilon \triangleq \frac{|d'-d|}{d'}$ where $\frac{1}{d'}$ is the the closest duty cycle supported by the simulated protocol w.r.t. $\frac{1}{d}$. Note that a smaller ϵ implies that the protocol can support more energy conservation levels with finer granularity. For Disco, in which the choice of prime numbers depends on the target discovery delay upper-bound, we configure the protocol by aligning the discovery delay bound to MCD in order to provide a common comparison baseline. For Searchlight, we set the smallest duty cycle unit as 2 to allow the finest duty cycle granularity.

The results are illustrated in Fig. 5, from which we make the following observations:

1) Searchlight has the worst performance on supporting duty cycles, because it restricts the duty cycle reciprocals to a power-multiple of the smallest one, i.e., 2, 4, 8 etc. As a natural consequence, when the required duty cycle reciprocal goes away from the power-multiples, the related error increases significantly. Such power-multiple-based error trend can be demonstrated by the power-multiple gap between neighboring delay peaks in the figure.
2) MCD achieves the best performance with ϵ monotonously decreasing in d except for $d = 38$ which corresponds to the only duty cycle not supported, meaning that a duty cycle $\frac{1}{37}$ or $\frac{1}{39}$ should be used. The result confirms the design philosophy of MCD stated in Section V. The highest error is around 7%, which confirms the analysis in Lemma 1.
3) The performance of Disco and U-Connect is between that of Searchlight and MCD. Compared to MCD, the performance variations are much more important.

C. Performance in Single-Channel Case

We now study the discovery performance of MCD in the single-channel case by comparing the worst-case discovery delay for the four protocols. We first investigate the neighbor discovery between two nodes a and b by simulating three representative scenarios depending on the duty cycles of a and b:

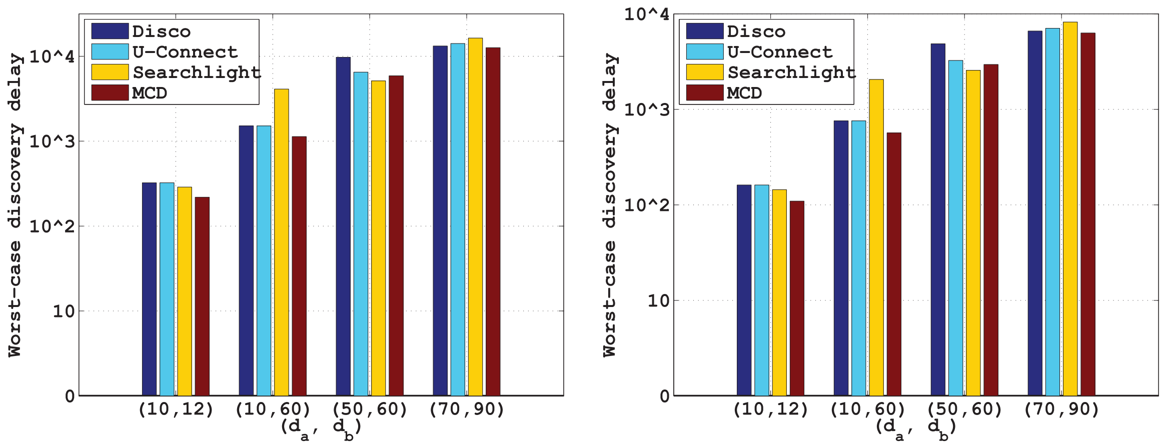
1) Both a and b have large duty cycles: $d_a = 10$, $d_b = 12$;
2) Both of them have small duty cycles: $d_a = 50$, $d_b = 60$ and $d_a = 70$, $d_b = 90$;
3) a has large duty cycle while b has small duty cycle.

For the three scenarios, we simulate under ns 2 both the case where the slots of a and b are aligned (their clocks are not synchronized) and where the slots are not aligned. In the latter case, we adopt the solution in [4] to let both nodes emit discovery beacons both at the beginning and at the end of each slot to increase the chance of discovery. In the asynchronous scenario, we set the clock skew of each node to be randomly distributed within $[-50 \text{ ppm}, 50 \text{ ppm}]$, given that a skew of 50 ppm corresponds to typical crystal clocks operating at extremes of their temperature specification. Neighbor discovery beacons are short messages containing node ID whose duration is around 1 ms. The results are plotted Figs. 6(a) and 6(b). Throughout our simulations, each point represents the worst-case value of a number of independent simulation runs, with the required number of simulation runs calculated using "independent replications" [29].

From the results, we can see that the worst-case delay of the simulated protocols does not have significant difference, except for the case (10, 60) where the delay of Searchlight outweighs the others. This is because approximating the duty cycle reciprocal 10 by a power-multiple 16 has a pronounced negative impact on the worst-case discovery delay. Moreover, the performance with non-aligned slots outperforms that with aligned slots, due to the adopted optimization technique to emit beacons both at the beginning and at the end of each slot. As a result, when the slots are not aligned, the probability of a partial overlap between two active slots is higher.

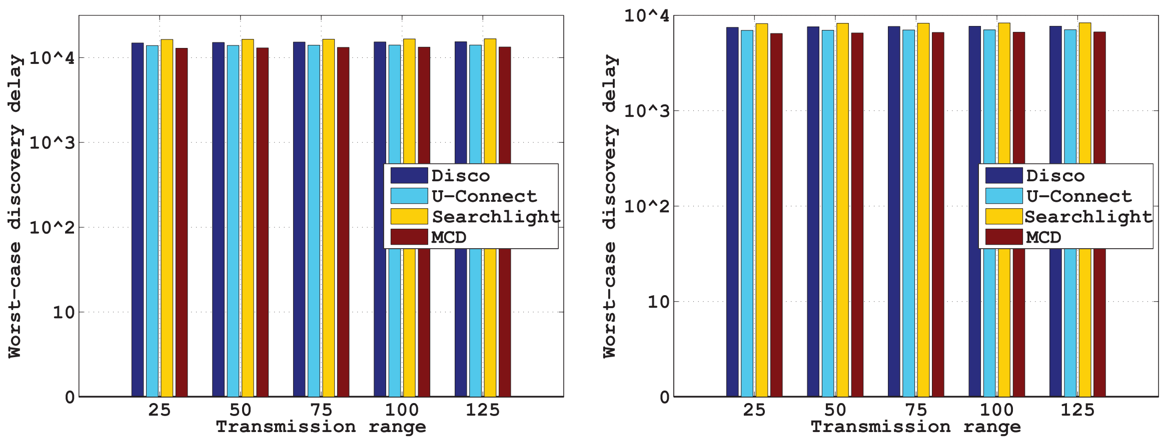
We then move to a more complex scenario of a randomly deployed mobile sensor network. To that end, we simulate in ns-2 a network with 100 nodes randomly deployed in a 200 m $\times$ 200 m square. We use the random waypoint mobility model [1], with the average speed of 2 m/s. We vary the transmission range of nodes from 25 m to 125 m such that the average number of neighbors of a node varies from around 3 to more than 50. The slot length is set to 25 ms. The duty cycle length of each node is randomly chosen from $i \times 100$ ms with $1 \leq i \leq 25$, $i \in \mathbb{N}$, i.e., $d_{\min} = 4$ and $d_{\max} = 100$. This setting is motivated by the fact that the usual duty cycles considered in mobile sensing applications are multiples of 100 ms in order to keep TCP traffic stable.

We trace the worst-case discovery delay among all pairs of neighbors in the network under both aligned and drifted slots, as illustrated in Figs. 7(a) and 7(b). The simulation results show similar characteristic as that in the pairwise case: all the simulated neighbor discovery algorithms have similar performance in terms of worst-case discovery delay; nevertheless, the delay of Searchlight outweighs the other three because the worst-case delay is dominated by that between nodes with the longest duty cycle 100, which is approximated by the power-multiple 128, thus leading to larger discovery delay. Another observation we can draw is that the discovery delay does not vary significantly



(a) (b)

Fig. 6. Performance comparison between MCD and major neighbor discovery protocols, single-channel case: worst-case discovery delay between two nodes with (a) aligned slots and (b) drifted slots.



(a) (b)

Fig. 7. Performance comparison between MCD and major neighbor discovery protocols, single-channel case: worst-case discovery delay of the entire network with (a) aligned slots and (b) drifted slots.

w.r.t. the transmission range (equivalently the number of neighbors). More specifically, we observe only a negligible slight increase of the delay as the transmission range increases.

D. Performance in Multi-Channel Case

We now evaluate MCD in the multi-channel case as among the simulated four deterministic neighbor discovery protocols, MCD is the only one supporting multiple channels. Specifically, we start with the pairwise neighbor discovery between two nodes a and b by simulating the following two scenarios for a system of $N = 10$ channels. The choice of 10 channels corresponds to Zigbee on 915 MHz band (we have also investigated the scenario with 16 channels corresponding to Zigbee on 2450 MHz band and the results are coherent with that of 10 channels presented in the paper).

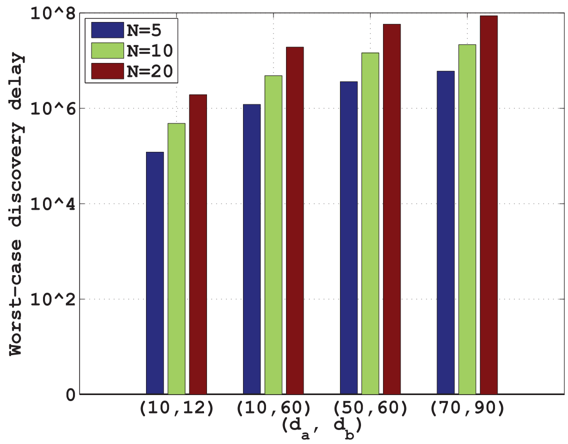
1) Both a and b have the same channel perception, i.e., $\mathcal{N}_a = \mathcal{N}_b = \mathcal{N}$. We simulate the sub-scenarios of both aligned and non-aligned slots for different N.
2) a and b have asymmetrical channel perceptions and drifted slots. We further simulate three sub-scenarios:
   a) There is only one common channel between them and $N_a = N_b = 3$;
   b) There are $N_c = N/2$ common channels and $N_a = N_b = 8$;
   c) The number of common channels N_c is randomly distributed in $[1, N]$ with random N_a, N_b supporting N_c.

From the simulation results in Figs. 8 and 9, we make the following observations:

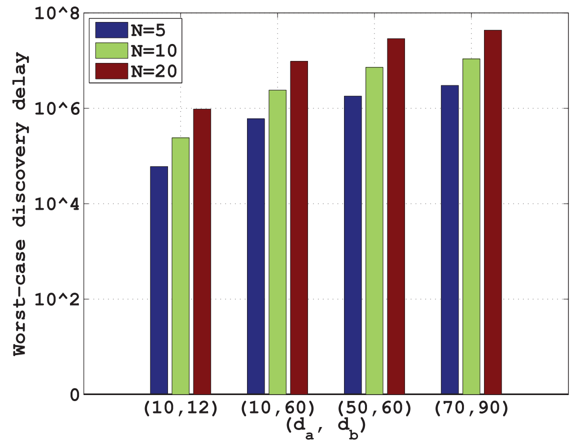
- As the system scales in terms of N, the discovery delay also increases. Moreover, we report that the delay increases squarely with the channel numbers, which is in accordance with the analytical results.
- Discovery is achieved on each channel that both nodes can access, even in the case where a and b have asymmetrical channel perceptions and drifted slots. This property makes MCD especially adapted in the decentralized mobile applications with heterogeneous wireless nodes.

We then investigate the neighbor discovery process with the same network setting as the single-channel case, except that here we set $N = 10$ channels and each node can access

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a)



(b)

Fig. 8. The worst-case discovery delay to full diversity under symmetrical channel perception: (a) aligned slots, (b) drifted slots.

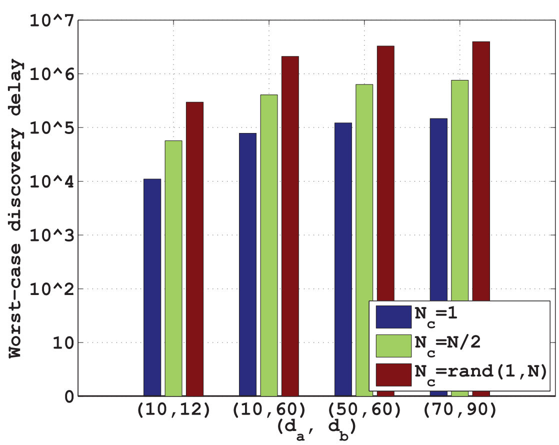


Fig. 9. The worst-case discovery delay to full diversity under asymmetrical channel perceptions and drifted slots.

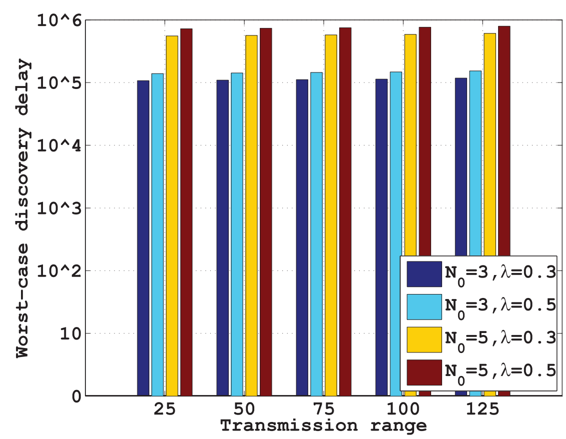


Fig. 10. The worst-case discovery delay in a multi-channel network under asymmetrical channel perceptions and drifted slots.

N_0 channels randomly chosen. To simulate the imperfectness of wireless medium, we set a parameter λ as the probability at which a transmission fails on any channel. We trace the worst-case discovery delay among all pairs of neighbors (if they have commonly accessible channel(s)) in such multi-channel network with drifted slots, as illustrated in Fig. 10. It can be observed that having more operational channels increases the discovery delay. As in the single-channel case, the discovery delay does not vary significantly w.r.t. the transmission range. The impact of collisions in our simulation is not pronounced (discovery can still be achieved at the price of longer delay) due to the following reasons (1) nodes are spread among different channels in the multi-channel case; (2) the clocks are not synchronized; (3) the beacon length is short.

VIII. Conclusion

In this paper, we have investigated the multi-channel neighbor discovery problem in multi-channel wireless networks. Our developed protocol *MCD* can achieve mutual discovery at minimal and bounded latency with full discovery diversity, even when the network nodes have asynchronous clocks and asymmetrical channel perceptions.

Our analysis also sheds light on the theoretical performance bound of any neighbor discovery protocol by relating the two important performance metrics, discovery delay and duty cycle. We believe that this fundamental result can provide useful guidelines on the design of other neighbor discovery protocols in future research.

References

[1] D. Alparslan and K. Sohraby, "Two-dimensional modeling and analysis of generalized random mobility models for wireless ad hoc networks," *EEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 616–629, Jun. 2007.

[2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 7, no. 3, pp. 537–568, May 2009.

[3] B. S. Baker, "Approximation algorithms for NP-complete problems on planar graphs," *J. ACM*, vol. 41, no. 1, pp. 153–180, Jan. 1994.

[4] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: Won't you be my neighbor," in *Proc. Mobicom*, 2012, pp. 185–196.

[5] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson, "EasyTracker: Automatic transit tracking, mapping, and arrival time prediction using smartphones," in *Proc. SenSys*, 2011, pp. 68–81.

[6] G. Chartrand and F. Harary, "Planar permutation graphs," *Ann. L'inst. Henri Poincaré (B) Probab. Statist.*, vol. 3, no. 4, pp. 433–438, 1967.

[7] P. Dutta *et al.*, "Common sense: Participatory urban sensing using a network of handheld air quality monitors," in *Proc. SenSys*, 2009, pp. 349–350.

[8] P. Dutta and D. Culler, “Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications,” in *Proc. SenSys*, 2008, pp. 71–84.
[9] C. Godsil and G. Royle*, Algebraic Graph Theory*. New York, NY, USA: Springer, 2001.
[10] K. Han, J. Luo, Y. Liu, and A. Vasilakos, “Algorithm design for data communications in duty-cycled wireless sensor networks: A survey,” *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 107–113, Jul. 2013.
[11] T. Huang, H. Chen, L. Cui, and Y. Zhang, “EasiND: Neighbor discovery in duty-cycled asynchronous multichannel mobile WSNs,” *Int. J. Distrib. Sensor Netw.*, vol. 2013, p. 403165, 2013.
[12] G. J. O. Jameson*, The Prime Number Theorem*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[13] J.-R. Jiang, “Expected quorum overlap sizes of quorum systems for asynchronous power-saving in mobile ad hoc networks,” *Comput. Netw.*, vol. 52, no. 17, pp. 3296–3306, 2008.
[14] A. Kandhalu, K. Lakshmanan, and R. Rajkumar, “U-Connect: A low-latency energy-efficient asynchronous neighbor discovery protocol,” in *Proc IPSN*, 2010, pp. 350–361.
[15] N. Karowski, A. C. Viana, and A. Wolisz, “Optimized asynchronous multi-channel discovery of IEEE 802.15.4-based wireless personal area networks,” *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1972–1985, Oct. 2013.
[16] S. Lai, “Heterogenous quorum-based wakeup scheduling for duty-cycled wireless sensor networks,” Ph.D. dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, 2009.
[17] J. Li, Z. Wang, L. Yao, and L. Guo, “Mnd: An information of detected neighbors based multi-channel neighbor discovery protocol for wireless sensor networks,” in *Advances in Wireless Sensor Networks*. New York, NY, USA: Springer, 2013, vol. 334, pp. 167–180.
[18] M. J. McGlynn and S. A. Borbash, “Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks,” in *Proc. MobiHoc*, 2001, pp. 137–145.
[19] T. Meng, F. Wu, and G. Chen, “On designing neighbor discovery protocols: A code-based approach,” in *Proc. IEEE INFOCOM*, 2014, pp. 1689–1697.
[20] E. Miluzzo *et al.*, “Sensing meets mobile social networks: The design, implementation and evaluation of the CenceMe application,” in *Proc. SenSys*, 2008, pp. 337–350.
[21] I. Niven, H. S. Zuckerman, and H. L. Mongomery*, An Introduction to the Theory of Numbers*. New York, NY, USA: Wiley, 1991.
[22] G. Sun, F. Wu, and G. Chen, “Neighbor discovery in low-duty-cycle wireless sensor networks with multipacket reception,” in *Proc. IEEE ICPADS*, 2012, pp. 490–497.
[23] W. Sun, Z. Yang, K. Wang, and Y. Liu, “Hello: A generic flexible protocol for neighbor discovery,” in *Proc. IEEE INFOCOM*, 2014, pp. 540–548.
[24] M. M. Syslo, “Characterizations of outerplanar graphs,” *Discrete Math.*, vol. 26, no. 1, pp. 47–53, 1979.
[25] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, “Cooperative transit tracking using smart-phones,” in *Proc. SenSys*, 2010, pp. 85–98.
[26] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, “Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks,” in *Proc. IEEE INFOCOM*, 2002, pp. 200–209.
[27] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, “Neighbor discovery in wireless networks and the coupon collector's problem,” in *Proc. MobiCom*, 2009, pp. 181–192.
[28] K. Wang, X. Mao, and Y. Liu, “BlindDate: A neighbor discovery protocol,” in *Proc. ICPP*, 2013, pp. 120–129.
[29] W. Whitt, “The efficiency of one long run versus independent replications in steady-state simulation,” *Manage. Sci.*, vol. 37, no. 6, pp. 645–666, 1991.
[30] T. Yan, V. Kumar, and D. Ganesan, “CrowdSearch: Exploiting crowds for accurate real-time image search on mobile phones,” in *Proc. MobiSys*, 2010, pp. 77–90.
[31] W. Zeng, S. Vasudevan, X. Chen, B. Wang, A. Russell, and W. Wei, “Neighbor discovery in wireless networks with multipacket reception,” in *Proc. MobiHoc*, 2011, Art. no. 3.
[32] R. Zheng, J. C. Hou, and L. Sha, “Asynchronous wakeup for ad hoc networks,” in *Proc MobiHoc*, 2003, pp. 35–45.

Lin Chen (S’07–M’10) received his B.E. degree in radio engineering from Southeast University, China, in 2002 and the Engineer Diploma from Telecom ParisTech, Paris, France, in 2005. He also holds an M.S. degree of networking from the University of Paris 6. He currently works as an Associate Professor in the Department of Computer Science of the University of Paris-Sud. He serves as Chair of the IEEE Special Interest Group on Green and Sustainable Networking and Computing with Cognition and Cooperation, IEEE Technical Committee on Green Communications and Computing. His main research interests include modeling and control for wireless networks, distributed algorithm design, and game theory.

Kaigui Bian received his Ph.D. degree in computer engineering from Virginia Tech, Blacksburg, VA, USA, in 2011. He is currently an Associate Professor in the Institute of Network Computing and Information Systems, School of EECS at Peking University. His research interests include mobile computing, cognitive radio networks, network security, and privacy.

Meng Zheng received the B.S. degree in information and computing science, and the M.S. degree in operational research and cybernetics from Northeastern University, Shenyang, China, and the Ph.D. degree in mechatronic engineering from Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, in 2005, 2008, and 2012, respectively. He had been a visiting student from 2010 to 2012 at the Fraunhofer Institute for Telecommunication, Heinrich-Hertz-Institute, Berlin, Germany. Currently, he is serving as an Associate Professor at Shenyang Institute of Automation, Chinese Academy of Sciences. His research interests include wireless ad hoc and sensor networks, cognitive radio network, and security in smart grid.