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

On Oblivious Neighbor Discovery in Distributed Wireless Networks With Directional Antennas: Theoretical Foundation and Algorithm Design

Lin Chen, *Member, IEEE*, Yong Li, *Senior Member, IEEE*, and Athanasios V. Vasilakos

***Abstract*— Neighbor discovery, one of the most fundamental bootstrapping networking primitives, is particularly challenging in decentralized wireless networks where devices have directional antennas. In this paper, we study the following fundamental problem, which we term oblivious neighbor discovery: How can neighbor nodes with heterogeneous antenna configurations discover each other within a bounded delay in a fully decentralised manner without any prior coordination or synchronisation? We establish a theoretical framework on the oblivious neighbor discovery and the performance bound of any neighbor discovery algorithm achieving oblivious discovery. Guided by the theoretical results, we then devise an oblivious neighbor discovery algorithm, which achieves guaranteed oblivious discovery with order-minimal worst case discovery delay in the asynchronous and heterogeneous environment. We further demonstrate how our algorithm can be configured to achieve a desired tradeoff between average and worst case performance.**

***Index Terms*— Neighbor discovery, directional antenna, discovery oblivity.**

I. INTRODUCTION

DIRECTIONAL antennas have been widely used in emerging wireless networks given the capability in limiting interference, enlarging transmission range and hence boosting network capacity and reducing energy consumption. For example, direction antennas are particularly attractive in the 60GHz networks to ensure high transmission quality and acquire sufficient link budget to cater Gbps data rate. As another example, in some wireless item tracking systems, devices are attached with directional antennas allowing them to scan an area for specific items.

Despite significant performance gain brought by directional antennas, their deployment brings particular design challenges for many fundamental communication and networking functionalities, some of which require a complete rethinking or redesign. In this paper, we focus on neighbor discovery, a supporting primitive that discovers all the neighbors in a device's communication range. It is one of the bootstrapping primitives supporting many basic network functionalities, such as topology control, clustering, medium access control, etc. Compared to the traditional omni-direction antenna paradigm, neighbor discovery with directional antennas is intuitively more challenging as directional antennas can only cover a fraction of the azimuth. Hence, neighbor discovery algorithms need to be carefully designed in order to guarantee that any pair of neighbor nodes can eventually steer their antennas toward each other at certain time instance. Moreover, nodes may not be synchronised and their antennas can be heterogeneous in terms of beamwidth. Neighbor discovery algorithms should be able to guarantee discovery in this challenging environment in a fully decentralised manner without any prior coordination.

Formally, we formulate the following *oblivious neighbor discovery* problem. How can neighbor nodes with heterogeneous antenna beamwidth discover each other within a bounded delay in a fully decentralised manner without any prior coordination or synchronisation? Particularly, the following requirements should be satisfied:

- Bounded and minimum worst-case discovery delay;
- Discovery oblivity, the capability of guaranteeing discovery regardless of the antenna beamwidth and the relative positions of nodes. This requirement is particular in the neighbor discovery with directional antennas.

As summarized in Section II, no existing work to our knowledge can satisfy the above requirements simultaneously. Aiming at providing a comprehensive investigation on oblivious neighbor discovery, we articulate our work as follows:

- *Theoretical framework.* We establish a theoretical framework on oblivious neighbor discovery and establish the performance bound of any oblivious neighbor discovery algorithm. Our theoretical results not only shed light on the structure of the problem, but also serve as design guidelines for oblivious neighbor discovery algorithms.
- *Algorithm design.* Guided by the theoretical results, we further design an oblivious neighbor discovery algorithm and prove that it achieves guaranteed oblivious discovery with order-minimal worst-case discovery delay in the asynchronous and heterogeneous environment. We further demonstrate how the algorithm can be configured to achieve a desired trade-off between average and worst-case performance.

Manuscript received April 5, 2016; revised December 21, 2016; accepted February 1, 2017; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor K. Jamieson. A preliminary version of this paper [1] was presented at INFOCOM 2016. *(Corresponding author: Yong Li.)*

L. Chen is with the Laboratory Recherche Informatique, CNRS UMR 8623, University of Paris-Sud, 91405 Orsay, France (e-mail: chen@lri.fr).

Y. Li is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: liyong07@tsinghua.edu.cn).

A. V. Vasilakos is with the Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 931 87 Skellefteå, Sweden (e-mail: th.vasilakos@gmail.com).

Digital Object Identifier 10.1109/TNET.2017.2673862

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

The paper is organised as follows. Section II gives a brief overview of related work. Section III formulates the oblivious neighbor discovery problem. Section IV establishes the theoretic performance bound for any oblivious neighbor discovery algorithm. Section V presents the design of an order-optimal oblivious neighbor discovery algorithm and mathematically establishes its performance. Section VI analyzes the performance when slots length at different nodes are not identical. Section VII addresses the issue of discovery beacon scheduling to ensure mutual discovery. Section VIII further demonstrates how the algorithm can be configured to achieve a desired trade-off between worst-case and average delay. Section IX presents the simulation results. Section X concludes the paper.

II. Related Work

As discussed in Section I, designing efficient neighbor discovery algorithms for devices with directional antennas is particularly challenging. A natural approach to contour the challenge is to use omni-directional antennas in the neighbor discovery process [2], [3] (cf. [4]–[12] for major neighbor discovery algorithms with omni-directional antennas). The main disadvantages of this approach is two-fold. Firstly, it requires an additional omni-directional antenna; Secondly, the discovered neighbor set using the omni-directional antenna can be significantly different from that using the directional one.

Neighbor discovery algorithms using purely directional antennas can be categorised into two classes, *probabilistic* and *deterministic* algorithms. In probabilistic approaches [13]–[21], each node randomly chooses a direction to steer its antenna. Probabilistic algorithms have the advantages of being memoryless and stationary and thus are especially robust and suitable in decentralised environments where no prior coordination or synchronisation is available. 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 algorithms [18], [22], [23], where each node points its antenna based on a predefined sequence, are proposed to provide guaranteed upper-bound on the worst-case discovery delay. However, existing deterministic neighbor discovery solutions with directional antennas either fail to achieve bounded discovery delay, or require time synchronisation among nodes, which may be not be practical in many applications or require prior coordination among nodes.

In spite of the existing research in the literature, *none of them can solve the oblivious neighbor discovery problem by ensuring nodes with heterogeneous antenna configurations and without clock synchronization to discover each other within a bounded delay in a fully decentralised manner without any prior coordination*, which is the focus of this paper.

III. System Model and Problem Formulation

A. System Model

We consider a time-slotted (but not necessarily synchronised) two-dimensional wireless network operating on a single frequency band. The set of nodes in the network is denoted by $\mathcal{S}$ with cardinality $S \triangleq |\mathcal{S}|$. Each node $i \in \mathcal{S}$ is equipped with a directional antenna with beamwidth θ_i ($0 < \theta_i \leq 2\pi$). When $\theta_i = 2\pi$, the antenna of node i degenerates to an omni-directional one. Under such generic antenna model, the communication range of node i can be divided into $N_i \triangleq \frac{2\pi}{\theta_i}$ non-overlapping sectors, indexed from 0 to $N_i - 1$ in clockwise.[1]

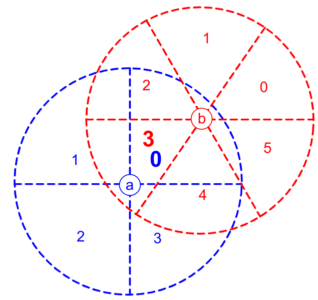

Fig. 1. Example of antenna configuration: the dotted blue and red circles represent the communication range of node a and b; a has $N_a = 4$ antenna sectors, b has $N_b = 6$ sectors; a is situated in sector $h_b = 3$ of b, b is situated in sector $h_a = 0$ of a.

To discover its neighbors, each node $i \in \mathcal{S}$ lets its antenna scan the communication range which is a disk around itself. As analysed in Section II, any probabilistic antenna scan strategy cannot achieve bounded discovery delay and suffers from the long-tail discovery latency problem in which two neighbor nodes a and b within the communication range of each other may experience extremely long delay before they can discover each other. Motivated by this observation, we consider deterministic neighbor discovery algorithms in which each node switches its antenna in each slot based on a specific pattern so as to discover its neighbors. We term such antenna pattern the *antenna scan pattern*, or *antenna scan sequence* and give its formal definition in the following.

Definition 1 (Antenna Scan Sequence): The antenna scan sequence is defined as a sequence $\mathbf{u} \triangleq \{u_t\}_{0 \leq t \leq T_u - 1}$ where u_t is the index of sector at which the antenna is steered, T_u is the period of the sequence.[2]

Now consider a pair of neighbor nodes a and b and assume that a is situated in the sector $h_b \in [0, N_b - 1]$ of b and b is situated in the sector $h_a \in [0, N_a - 1]$ of a, they can discover each other if and only if they steer their antennas towards each other. Formally, let $\mathbf{u}$ and $\mathbf{v}$ denote the antenna scan sequences of a and b, with periods T_a and T_b, if there exists $t \in [0, T_a T_b - 1]$ such that $u_t = h_a$ and $v_t = h_b$, a and b can discover each other in slot t. Figure 1 and Example 1 further illustrate the above definition. Table I lists the major notations used in the paper.

Example 1: Consider the setting of Figure 1 with the following two scenarios:

- *Scenario 1: $\mathbf{u} = \{0, 1, 2, 3\}$ with $T_a = 4$ and $\mathbf{v} = \{5, 4, 3, 2, 1, 0\}$ with $T_b = 6$, i.e., a lets its antenna scan counter-clockwise while b lets its antenna scan clockwise;*

[1] To make our analysis concise, we assume that N_i is an integer. The generation to non-integer N_i is trivial by letting the last sector be partially overlapped with its neighbor sectors.

[2] A probabilistic neighbor discovery strategy can be regarded as a special case where $T_u \to \infty$.

TABLE I
MAJOR NOTATIONS

$\mathcal{S}$	Set of nodes in the network
θ_i	Beamwidth of node i's antenna
N_i	Number of antenna sectors of node i
$\mathbf{u}$, $\mathbf{v}$	Antenna scan sequences
$\mathbf{u}(k)$	Cyclic rotation of $\mathbf{u}$ by k positions
t_i^0	Node i's initial clock offset
$u_t(t_i^0)$	Position of antenna at slot t with initial clock offset t_i^0
$\mathbf{i}$	ID of node i
$\mathbf{I}$	Extended ID of node i
l	Length of IDs
L	Length of extended IDs
p_i	Smallest odd prime no smaller than N_i
q_i	Smallest power-multiple of 2 no smaller than N_i
ρ_{ab}	Relative clock skew rate between nodes a and b
Λ_i	Slot duration of node i

Fig. 2. Example of antenna scan sequences: upper: Scenario 1; lower: Scenario 2.

- *Scenario 2:* $\mathbf{u} = \{0,1,2,3\}$ *with* $T_a = 4$ *and* $\mathbf{v} = \{0,1,2,3,4,5\}$ *with* $T_b = 6$*, i.e., both of them let their antenna scan counter-clockwise.*

If b *is situated in sector* 0 *of* a *(*$h_a = 0$*) and* a *in sector* 3 *of* b *(*$h_b = 3$*), it can be checked that they can discover each other in slot* 8 *in Scenario 1, while they cannot discover each other in Scenario 2. The antenna scan sequences and the discovery process are illustrated in Figure 2.*

To model the situation where nodes are not synchronised, we apply the concept of *cyclic rotation* to antenna scan sequences. Given an antenna scan sequence $\mathbf{u}$, a cyclic rotation of $\mathbf{u}$ by k positions simply moves the first k elements in $\mathbf{u}$ to the end, while shifting all other elements to the left. Specifically, given an antenna scan sequence $\mathbf{u}$, we denote $\mathbf{u}(k)$ a cyclic rotation of $\mathbf{u}$ by k positions where k is referred to as the *cyclic rotation phase*. Consider an example where $\mathbf{u} = \{0,1,2,3\}$ with $T_u = 4$, we have $\mathbf{u}(2) = \{2,3,0,1\}$. The situation where k is fractional, corresponding to the case where time slots of different nodes are not aligned, is analysed in Section VI.

B. Problem Formulation

From Example 1, we can see that the antenna scan sequences should be carefully devised to guarantee discovery between any pair of neighbor nodes. To evaluate the performance of a neighbor discovery algorithm, we introduce the following two performance metrics:

- *Discovery oblivity.* The first metric, specific for the problem of neighbor discovery with directional antennas, is the discovery oblivity, which characterizes the capability of a neighbor discovery algorithm of discovering neighbors regardless of their antenna beamwidth, relative positions and clock drift. A neighbor discovery algorithm is oblivious if it can guarantee discovery between any pair of neighbors a and b for any combination $(N_a, N_b) \in \mathbb{N}^2$, $(h_a, h_b) \in [0, N_a - 1] \times [0, N_b - 1]$, and any initial clock offset combination $(t_a^0, t_b^0) \in [0, T_a - 1] \times [0, T_b - 1]$.
- *Worst-case discovery delay.* Given two nodes a and b, the worst-case discovery delay between them is defined as the upper-bound of the latency (in number of slots) before successful discovery for all possible clock drfits.

Armed with the above definitions and related mathematic notations introduced in this section, we can formulate the oblivious neighbor discovery problem.

Problem 1 (Oblivious neighbor discovery problem): The oblivious neighbor discovery problem is defined as follows:

$$\begin{aligned} &\textit{minimise} \quad T, \\ &\textit{subject to} \quad \forall t_a^0 \in [0, T_a - 1], t_b^0 \in [0, T_b - 1], \forall N_a, N_b \in \mathbb{N}, \\ &\qquad \textit{and } \forall h_a \in [0, N_a - 1], h_b \in [0, N_b - 1], \\ &\qquad \exists t \le T \textit{ such that } u_t(t_a^0) = h_a, v_t(t_b^0) = h_b. \end{aligned}$$

That is, devising antenna scan sequences to minimize T*, the worst-case discovery delay, while guaranteeing discovery between any pair of neighbor nodes* a *and* b *for any combination of* (N_a, N_b)*, any combination of* $(h_a, h_b) \in [0, N_a - 1] \times [0, N_b - 1]$*, and any combination of* (t_a^0, t_b^0)*.*

IV. Theoretical Foundation and Bound

Before presenting the design of oblivious neighbor discovery algorithm, we first lay the theoretical foundation and bound for any oblivious neighbor discovery algorithm. We start by proving a structural property of the antenna scan sequence of any oblivious neighbor discovery algorithm.

Lemma 1: If two nodes a *and* b *can achieve oblivious discovery with the worst-case discovery delay* D *by using the antenna scan sequences* $\mathbf{u}$ *and* $\mathbf{v}$*, then for any combination of cyclic rotation phases* (t_a^0, t_b^0) *and any combination* $(h_a, h_b) \in [0, N_a - 1] \times [0, N_b - 1]$*, there exists* $t < D$ *such that* $u_t(t_a^0) = h_a$ *and* $v_t(t_b^0) = h_b$*.*

Proof: We prove the lemma by contradiction. Assume that there exits a combination (h_a^0, h_b^0) such that there does not exist $t < D$ such that $u_t(t_a^0) = h_a^0$ and $v_t(t_b^0) = h_b^0$. Then consider the case where a is situated in the sector h_b^0 of b and b is situated in the sector h_a^0 of a, it can be noted that a and b cannot discover each other within D slots in this case, which contradicts the condition that they can achieve oblivious discovery within D slots. □

Lemma 1 implies that for any t_a^0 and t_b^0, the couple $(u_t(t_a^0), v_t(t_b^0))$ $(0 \le t < D)$ needs to cover all couples in $[0, N_a - 1] \times [0, N_b - 1]$ so as to ensure discovery within D slots. This observation readily leads to the following theorem on the worst-case discovery delay bound for any oblivious neighbor discovery algorithm.

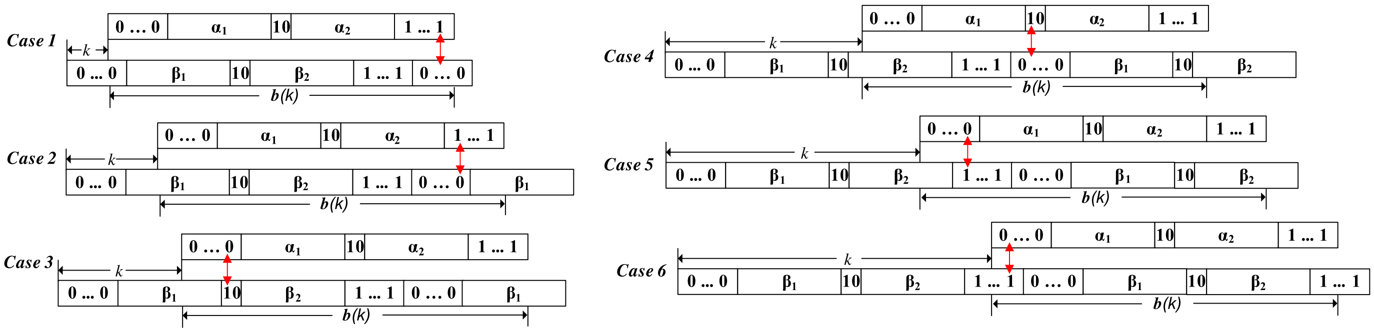

Fig. 3. Illustration of the six cases in the proof of Lemma 2.

Theorem 1 (Worst-case Discovery Delay Bound): For any oblivious neighbor discovery algorithm, the worst-case discovery delay between any pair of neighbor nodes a and b cannot be lower than $N_a N_b$.

Theorem 1 derives the performance limit of any oblivious neighbor discovery algorithm. We can further generalise 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 antenna sector numbers of neighbor nodes are N_1 and N_2, the worst-case discovery delay for any pair of neighbor nodes in the network to discover each other, denoted by D_n, is lower-bounded by $N_1 N_2$ for any oblivious neighbor discovery algorithm. Asymptotically, when $N_1 \simeq N_2 \simeq O(N)$, $D_n \simeq O(N^2)$.

V. Distributed Algorithm Design

In this section, we devise a neighbor discovery algorithm achieving oblivious discovery between any pair of neighbors a and b and approaching the theoretical performance limit established in Section IV. Our design is composed of two steps. In the first step, each node constructs a binary sequence such that the sequences of any two distinct nodes are cyclic rotationally distinct. Given any two sequences $\mathbf{a}$ and $\mathbf{b}$ of l bits, they are cyclic rotationally distinct to each other other if and only if $\mathbf{a} \neq \mathbf{b}(k)$ for any $k \in [0, l-1]$. In the second step, each node constructs its antenna scan sequence using the sequence constructed in the first step.

A. Constructing Cyclic Rotationally Distinct Sequence

In our approach, the antenna scan sequence for each node is constructed based on its globally unique ID (e.g., address), which can be mathematically expressed as a binary sequence of length l. Using globally unique IDs is a typical symmetry breaking technic in distributed computing.

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 to the other. We term the sequences generated from the ID sequences the *extended ID sequences*.

A simple way of constructing cyclic rotationally distinct extended ID sequences has been proposed in [24] as summarised below: let $\mathbf{i}$ denote the ID of node i, which is an l-bit binary sequence; let $\mathbf{1}(k)$ ($\mathbf{0}(k)$) denote a binary sequence of 1 (0) of length k; construct the following binary sequence $\mathbf{I} \triangleq \mathbf{i}||\mathbf{1}(l)||\mathbf{0}(l)$. It is proved in [24] that sequences constructed in this way are cyclic rotationally distinct to each other. A generalised algorithm has then been developed in [1] producing shorter sequence.

We further improve the algorithms in [1] and [24] to the algorithm below. The reason of developing a new algorithm is two-fold: (1) The establishment of bounded discovery delay in our problem requires the length of the extended ID sequence to be odd, (cf. proof of Theorem 2) which cannot be achieved by the algorithm in [24]; (2) The length of the extended ID sequence generated by our algorithm is significantly shorter than those in [1] and [24], which leads to shorter discovery delay.

Our algorithm works as follows. Let $\mathbf{i}$ denote the ID of node i whose length is l bits ($l \geq 2$) where $i(k)$ denotes bit k of $\mathbf{i}$. Let $\mathbf{i_1}$ and $\mathbf{i_2}$ denote the subsequence from $i(1)$ to $i(k_m - 1)$ and from $i(k_m)$ to $i(l-2)$, where $k_m = \lfloor \frac{l}{2} \rfloor$. The extended ID sequence of i, denoted by $\mathbf{I}$, is constructed as below.

$$\mathbf{I} \triangleq \mathbf{0}(l_1)||\mathbf{i_1}||10||\mathbf{i_2}||\mathbf{1}(l_2),$$

For example, if $\mathbf{i} = 11010010$ and $l_1 = 3$, $l_2 = 2$, we have $\mathbf{I} = 000110110001011$. The lemma below proves that the extended ID sequences generated by our algorithm are cyclic rotationally distinct one to each other.

Lemma 2: Given any extended ID sequences $\mathbf{a}$ and $\mathbf{b}$ generated from two ID sequences α and β, if $l_1 > l_2$ and $l_1 + l_2 > k_m$, it holds that $\mathbf{a}$ and $\mathbf{b}$ are cyclic rotationally distinct to each other, i.e., it holds that

$$\alpha \neq \beta \implies \mathbf{a} \neq \mathbf{b}(k), \ \forall k \in [0, L),$$

where $\mathbf{b}(k)$ is $\mathbf{b}$ with a cyclic rotation of k bits, $L = l + l_1 + l_2 + 2$ is the length of the extended ID sequences.

Proof:

We prove the lemma by considering the six possible scenarios illustrated in Figure 3, and showing, in each scenario, that a bit in $\mathbf{a}$ and another bit in $\mathbf{b}(k)$ have different values although the two bits are in the same position within the respective extended ID sequences. This is sufficient to prove that the two extended ID sequences $\mathbf{a}$ and $\mathbf{b}$ are cyclic rotationally distinct one to the other.

Case 1: $k \in (0, l_1]$. As indicated by the arrow in Figure 3, it holds that $a_{L-1} = 1$ and $b_{L-1}(k) = 0$.

Case 2: $k \in (l_1, l_1 + l_2)$. As indicated by the arrow in Figure 3, it holds that $a_{l+l_1} = 1$ and $b_{l+l_1}(k) = 0$.

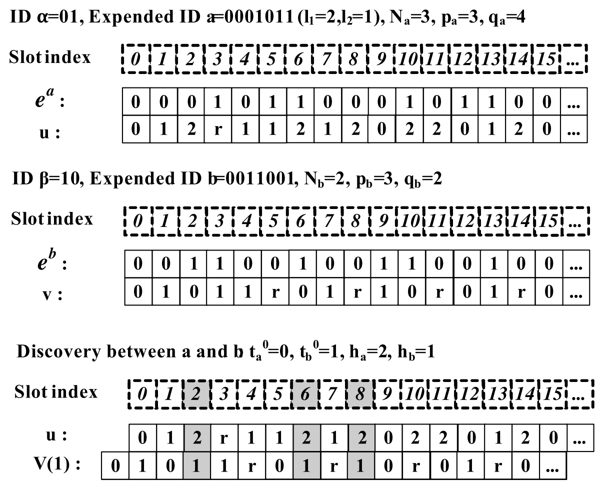

Fig. 4. Example of antenna scan sequences for node a and b (r denotes a randomly number in $[0, N_i - 1]$): they can discover each other in slots 2, 6 and 8.

Case 3 (Case 3 exists if $k_m > l_2$, otherwise this case does not exist and we proceed directly to Case 4): $k \in [l_1 + l_2, l_1 + k_m]$. Since $l_1 + l_2 > k_m$, as indicated by the arrow in Figure 3, it holds that $a_{l_1-1} = 0$ and $b_{l_1-1}(k) = 1$.

Case 4: $k \in (l_1 + k_m, L - k_m]$. As indicated by the arrow in Figure 3, it holds that $a_{l_1+k_m} = 1$ and $b_{l_1+k_m}(k) = 0$.

Case 5 (Case 5 exists if $k_m > l_1$, otherwise this case does not exist and we proceed directly to Case 6): $k \in (L - k_m, L - l_1)$. As indicated by the arrow in Figure 3, it holds that $a_{l_1-1} = 0$ and $b_{l_1-1}(k) = 1$.

Case 6: $k \in [L - l_1, L)$. Since $l_1 > l_2$, as indicated by the arrow in Figure 3, it holds that $a_1 = 0$ and $b_1(k) = 1$.

Noticing that $\alpha \neq \beta \implies \mathbf{a} \neq \mathbf{b}$, we thus conclude that $\mathbf{a} \neq \mathbf{b}(k),\ \forall k \in [0, L)$. □

In our design, we set l_1 and l_2 such that the total length of the extended ID sequence L is odd and minimum, e.g., when l is even (hence $k_m = \frac{l}{2}$), we set $l_1 = \lceil \frac{k_m}{2} \rceil + 1$ and $l_2 = \lceil \frac{k_m}{2} \rceil$; the length of the extended ID sequences is $L = l + 2\lceil \frac{l}{4} \rceil + 3 \simeq 1.5l$. Note that the length of the extended ID sequences in [1] and [24] are asymptotically $3l$ and $2l$.

B. Constructing Antenna Scan Sequence

In the second step, each node i constructs its antenna scan sequence $\mathbf{u}$ based on its extended ID sequence, denoted as $\mathbf{e}^i$, generated in the first step by choosing l_1 and l_2 such that the resulting sequence length L is odd. Specifically, let p_i denote the smallest odd prime number not smaller than N_i and co-prime to L; let b_i denote the smallest integer satisfying $2^{b_i} \geq N_i$ and set $q_i = 2^{b_i}$; the antenna scan sequence of node i, $\mathbf{u}$, is constructed as follows:

$$u_t = \begin{cases} t \bmod p_i & e^i_t = 0 \text{ and } t \bmod p_i < N_i, \\ t \bmod q_i & e^i_t = 1 \text{ and } t \bmod q_i < N_i, \\ \text{rand}(N_i - 1) & \text{otherwise}, \end{cases} \quad (1)$$

where $\text{rand}(N_i - 1)$ denotes a random integer in $[0, N_i - 1]$. It can be noted that the period of the antenna scan sequence $\mathbf{u}$ is Lp_iq_i without taking into account the random part. Figure 4 provides an example of the antenna scan sequences for two nodes a and b and their discovery process.

C. Discovery Delay Analysis

In the following theorem, we prove the correctness of our algorithm in achieving oblivious discovery and establish the worst-case discovery delay bound.

Theorem 2: (Correctness and Worst-Case Discovery Delay Bound:) Our neighbor discovery algorithm can ensure oblivious discovery between any pair of neighbors a and b. The worst-case discovery delay between them is upper-bounded by $L \max\{p_aq_b, p_bq_a\}$, asymptotically $O(N_aN_b)$.

Proof: For any system parameter combination (t^0_a, t^0_b), (N_a, N_b) and (h_a, h_b), it follows from Lemma 2 that there exists $0 \leq l_0 < L$ such that $e^a_{l_0}(t^0_a) \neq e^b_{l_0}(t^0_b)$. Assume that $e^a_{l_0}(t^0_a) = 0$ while $e^b_{l_0}(t^0_b) = 1$.

As p_a is an odd prime and q_b is a power-multiple of 2, it holds that p_a is co-prime with q_b. Let $\mathbf{u}$ and $\mathbf{v}$ denote the antenna scan sequences of a and b. We examine the slots $t_k = l_0 + kL$ where $k \in \mathbb{N}$ by considering $\{u_{t_k}(t^0_a)\}$ and $\{v_{t_k}(t^0_b)\}$, i.e., the antenna scan sequences $\mathbf{u}(t^0_a)$ and $\mathbf{v}(t^0_b)$ at these slots. It follows from (1) that

$$\begin{cases} u_{t_k}(t^0_a) = t^0_a + l_0 + kL \bmod p_a, \\ v_{t_k}(t^0_b) = t^0_b + l_0 + kL \bmod q_b. \end{cases}$$

Note that (1) L is odd, (2) p_a is an odd prime and co-prime to L, (3) q_b is a power-multiple of 2, it holds that L, p_a and q_b are co-prime one to another. By applying the Chinese Remainder Theorem [25], we have for any parameter settings (t^0_a, t^0_b), (N_a, N_b) and (h_a, h_b), we can find $k_0 < p_aq_b$ where

$$\begin{cases} k_0L \bmod p_a = h_a - t^0_a - l_0 \bmod p_a, \\ k_0L \bmod q_b = h_b - t^0_b - l_0 \bmod q_b. \end{cases}$$

It then follows that

$$\begin{cases} u_{t_k}(t^0_a) = t^0_a + l_0 + kL \bmod p_a = h_a, \\ v_{t_k}(t^0_b) = t^0_b + l_0 + kL \bmod q_b = h_b. \end{cases}$$

Hence, a and b can discover each other in slot t_{k_0} with the worst-case discovery delay bounded by Lp_aq_b.

Symmetrically, when $e^a_{l_0}(t^0_a) = 1$ while $e^b_{l_0}(t^0_b) = 0$, we can show by the same reasoning that the worst-case discovery delay is upper-bounded by Lq_ap_b. Therefore, it holds that the worst-case discovery delay of our algorithm is upper-bounded by $L \max\{p_aq_b, q_ap_b\}$. In the asymptotical case, we have $p_a \simeq q_a \simeq N_a$ and $p_b \simeq q_b \simeq N_b$ and hence the delay upper-bound is $O(N_aN_b)$. □

We end this subsection with the following two remarks:

- *Tightness of worst-case discovery delay.* Theorem 2 establishes the worst-case discovery delay bound as $L \max\{p_aq_b, p_bq_a\}$. We illustrate via an example in Figure 5 that this bound is actually very tight. In the example where the initial clock drift is $t^0_a = 56$ and $t^0_b = 0$, a and b discover each other only at slot 82, which corresponds to the discovery delay of 83 slots. The worst-case discovery delay bound derived by Theorem 2 in this example is 84.
- *Upper-bound of average discovery delay.* We can derive the upper-bound of the average discovery delay by using the same technique as the proof of Theorem 2. Specifically, using the same notation, given a random pair of

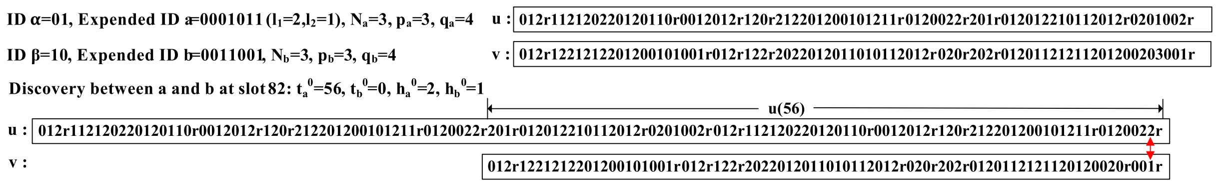

Fig. 5. Tightness of worst-case discovery delay.

t_a^0 and t_b^0, the expectation of k_0 is bounded by $\frac{p_a q_b - 1}{2}$. Assume that node IDs can be regarded as random binary sequences, the expectation of l_0 is bounded by $\frac{L}{2}$. The average discovery delay is thus upper-bounded by $\frac{L p_a q_b}{2}$, and asymptotically when $p_i \simeq q_i \simeq N_i \simeq N$, it can be bounded by $\frac{LN^2}{2}$. Note that this is a very conservative and thus loose bound, as illustrated in the simulations.

D. An Improved Algorithm

Recall Theorem 1 that the worst-case discovery delay lower bound is $N_a N_b$. In our algorithm, the worst-case delay is determined by $\max\{p_a q_b, p_b q_a\}$. Without loss of generality, assume that $p_a q_b \geq p_b q_a$ and hence the worst-case delay is $p_a q_b$. In our design, p_a is the smallest odd prime number N_a. It is well-known that for any N_a, we can find p_a which is very close to N_a. However, q_b needs to be a power-multiple of two. In the extremely unlucky case where N_b is in the form of $2^n + 1$, we may have $q_b \simeq 2N_b$, thus leading to $p_a q_b \simeq 2N_a N_b$.

Motivated by the above observation, we propose an improved algorithm to further limit the worst-case discovery delay even in the extremely unlucky case. In the improved algorithm, each node i independently chooses p_i as the smallest odd prime number larger than $\max\{3, N_i\}$ and $q_i = 2^{b_{1,i}} 3^{b_{2,i}}$ where $b_{1,i}$ and $b_{2,i}$ are integers chosen from $[0, \lceil \log_2 N_i \rceil]$ and $[0, \lceil \log_3 N_i \rceil]$ that minimizes $q_i - N_i$ under the constraint $q_i \geq N_i$, i.e.,

$$(b_{1,i}, b_{2,i}) = \operatorname*{argmin}_{b_{1,i}, b_{2,i}} \left(2^{b_{1,i}} 3^{b_{2,i}} - N_i\right), \ s.t. \ 2^{b_{1,i}} 3^{b_{2,i}} \geq N_i.$$

Lemma 3 proves that q_i is asymptotically close to N_i.

Lemma 3: For any $\epsilon > 0$, given N_i sufficiently large, there exist $b_{1,i} \in [0, \lceil \log_2 N_i \rceil]$ and $b_{2,i} \in [0, \lceil \log_3 N_i \rceil]$ such that $q_i = 2^{b_{1,i}} 3^{b_{2,i}} \geq N_i$ and $q_i - N_i \leq \epsilon$, i.e., N_i can be arbitrarily closely approximated by q_i.

Proof: We give the proof sketch. We prove the lemma by showing for large enough N_i that there exist $b_{1,i} \in [0, \lceil \log_2 N_i \rceil]$ and $b_{2,i} \in [0, \lceil \log_3 N_i \rceil]$ such that

$$\log_2 N_i \leq b_{1,i} + b_{2,i} \log_2 3 < \log_2 N_i + \epsilon.$$

This follows from the fact that the fractional part of $x \log_2 3$ where $x \in \mathbb{N}$, i.e., $x \log_2 3 - \lfloor x \log_2 3 \rceil$, are dense in $[0, 1]$. In fact, given any $\epsilon > 0$, if we choose non-negative integers $\{x_i\}$ so that fractional parts of $x \log_2 3$ form an $\frac{\epsilon}{2}$-set of $[0, 1]$, then we can choose the appropriate integer $b_{2,i}$ and then $b_{1,i}$, which is feasible provided that N_i is large enough. □

We can then use the same analysis as that of proof of Theorem 2 to show that the worst-case discovery delay of the improved algorithm is $L \max\{p_a q_b, p_b q_a\}$ if L is configured to be co-prime to 2 and 3. This result decreases by half the worst-case asymptotic delay of the original algorithm in the extremely unlucky case.

VI. Discovery Analysis With Non-Aligned Slots

Our previous results implicitly assume slots are aligned. We now relax this assumption to study the effect of slot non-alignment and asymmetrical slot length due to lack of clock synchronization and relative clock drift between any two nodes.

Analytically, for any pair of nodes a and b, we can write the local time at b, termed as t_b, as a function of the local time of a, termed as t_a, as below.

$$t_b(t_a) = \rho_{ab}(t_a - t_0) + t_b(t_0),$$

where ρ_{ab} is the relative rate of clock of node b as a function of the clock of a, $t_b(t_0) = t_0$ is the initial clock offset between a and b. If a and b are perfectly synchronized, we have $\rho_{ab} = 1$ and $t_b(t_0) = 0$. However, ρ_{ab} may drift away from 1 in practice. That is

$$1 - \Delta\rho_{max} \leq \rho_{ab} \leq 1 + \Delta\rho_{max},$$

where $\Delta\rho_{max}$ denote the upper-bound of the drift and is around the order of 10^{-6} in practice. Without losing generality, we consider the case where the clock of b goes no slower than a: $\rho_{ab} \leq 1$. We then distinguish two cases: (1) $\rho_{ab} = 1$, i.e., although the local time at a may differ to the local time at b, their slot duration is the same; (2) $\rho_{ab} \neq 1$, i.e., the slot duration of a and b is different.

A. Identical Slot Duration

We first investigate the first case $\rho_{ab} = 1$. In neighbor discovery, it is required that a neighbor discovery algorithm should be able to ensure that any pair of neighbor nodes can discover each other with an overlap of α slot where $\alpha \in (0, 1]$ is a system-dependent parameter.[3] A typical condition widely imposed in the literature is to require a discovery to last at least half of the slot duration, i.e., $\alpha = 0.5$.

We next demonstrate that our algorithm can achieve the above practical objective. To show this, consider two nodes a and b whose extended ID sequences are denoted as e^a and e^b. Given any parameter setting (t_a^0, t_b^0), (N_a, N_b) and (h_a, h_b) with non-aligned slots, it holds that either $u_t(t_0^a)$ and $v_t(t_0^b)$ overlap for at least half slot duration for any

[3] A practical example is that switching antenna from one direction to another incurs non-negligible delay.

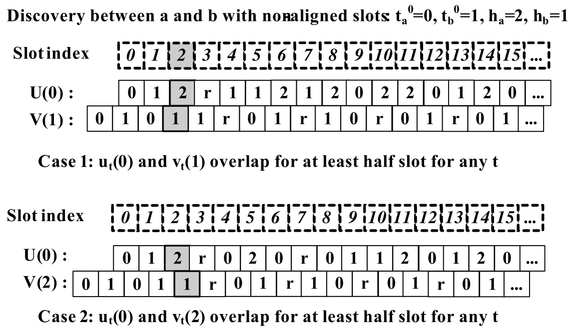

Fig. 6. Neighbor discovery with non-aligned slots but identical slot duration.

$t \geq 0$ or $u_t(t_0^a)$ and $v_t(t_0^b+1)$ overlap for at least half slot duration for any $t \geq 0$. We thus investigate these two cases:

- *Case 1:* $u_t(t_0^a)$ and $v_t(t_0^b)$ overlap for at least half slot duration for any $t \geq 0$. In this case, the previous analysis can be directly applied. The only difference is that instead of an entire overlap, a discovery in this case is a partial overlap of at least half slot duration.
- *Case 2:* $u_t(t_0^a)$ and $v_t(t_0^b+1)$ overlap for at least half slot duration for any $t \geq 0$. In this case, since $\mathbf{u}$ and $\mathbf{v}$ are cyclic rotationally distinct to each other, we can prove in the same way as Theorem 2 that within the same delay bound, there exists t^* such that $u_{t^*}(t_0^a) = h_a$ and $v_{t^*}(t_0^b+1) = h_b$. Hence a and b can discovery each other in slot t^* with an overlap of at least half slot duration.

Figure 6 illustrates the two cases of the neighbor discovery with non-aligned slots with the scan sequences of the example in Figure 4. As proved in this subsection as well as illustrated in Figure 6, in both cases, a and b can discover each other within the worst-case delay derived in Theorem 2 with an overlap of more than half slot.

B. Non-Identical Slot Duration

We now investigate the more practical case with non-identical slot durations where $\rho_{ab} < 1$, Clearly the slots of a and b are not necessarily aligned to each other. Mathematically, we assume that each slot of b lasts ρ_{ab} by normalizing the slot duration of a as unit time. We first state the following property [10].

Lemma 4: 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 $\frac{p(\lfloor \rho_{ab} n \rfloor)}{p(n)}$.

The theorem below establishes the worst-case discovery delay bound of our approach in this case. For presentation conciseness, we establish our result for our baseline discovery algorithm. The same bound holds straightforwardly for the improved approach.

Theorem 3: (Worst-Case Discovery Delay Bound: Non-Identical Slot Duration:) The worst-case discovery delay of our algorithm is upper-bounded by $\rho_{ab} L \max\{p_a q_b, q_a p_b\}$.

Proof: We divide each slot of nodes a and b into mini-slots of duration $\frac{1}{p(n)}$ where n is sufficiently large as in [10]. Consequently, each slot of a and b contain $p(n)$ and $p(\lfloor \rho_{ab} n \rfloor)$ mini-slots. The antenna scan sequences of a and b in (1) can then be written in mini-slots as below:

$$u_t^a = \begin{cases} tp(n) - l_a & e_t^a = 0, t \bmod p_a p(n) < N_a, \\ \bmod p_a p(n) & \text{and } 0 \leq l_a \leq p(n) - 1, \\ tp(n) - l_a & e_t^a = 1, t \bmod q_a p(n) < N_a, \\ \bmod q_a p(n) & \text{and } 0 \leq l_a \leq p(n) - 1, \\ \text{rand}(N_a - 1) & \text{otherwise}, \end{cases}$$

$$u_t^b = \begin{cases} tp(\lfloor \rho_{ab} n \rfloor) - l_b & e_t^b = 0, t \bmod p_b p(\lfloor \rho_{ab} n \rfloor) < N_b, \\ \bmod p_b p(\lfloor \rho_{ab} n \rfloor) & \text{and } 0 \leq l_b \leq p(\lfloor \rho_{ab} n \rfloor) - 1, \\ tp(\lfloor \rho_{ab} n \rfloor) - l_b & e_t^b = 1, t \bmod q_b p(\lfloor \rho_{ab} n \rfloor) < N_b, \\ \bmod q_b p(\lfloor \rho_{ab} n \rfloor) & \text{and } 0 \leq l_b \leq p(\lfloor \rho_{ab} n \rfloor) - 1, \\ \text{rand}(N_b - 1) & \text{otherwise}, \end{cases}$$

Given that n is sufficiently large, specifically, $p(\lfloor \rho_{ab} n \rfloor) > L \max\{p_a q_b, q_a p_b\}$, we can use the same co-primarity analysis as that in the proof of Theorem 2 to show that within at most $L \max\{p_a q_b, q_a p_b\} p(\lfloor \rho n \rfloor)$ mini-slots (i.e., $\rho_{ab} L \max\{p_a q_b, q_a p_b\}$ time by normalizing the a's slot to unit time), we can find a mini-slot t_m such that a and b can discover each other. We then have that from mini-slot t_m to $t_m + p(\lfloor \rho_{ab} n \rfloor)$ (i.e., one slot duration for b), both a and b point his antenna towards each other, and can thus discover each other with an overlap of at least half slot duration. $\square$

VII. Discovery Beacon Scheduling

Our theoretical analysis hinges on the fact that two neighbor nodes are able to discover each other once they steer their antennas to each other at the same slot for at least half of a slot during which they exchange discovery beacons. This assumption is also largely made in the literature. In this section, we design discovery beacon scheduling to achieve discovery once an overlap of at least half slot occurs. By overlap, we mean that in the overlapping slot, a and b steer their antennas toward each other.

Before motivating and discussing our design, we present a beacon scheduling mechanism initially proposed in [6] and improved in [8]. In this approach, each node sends two beacons each active slot, one at the beginning of the slot, the other at the end. The node remains in listening mode in the intermediate period. Under the condition that the slots of two nodes are not perfectly aligned, they can receive a beacon from the other node in each overlapping active slots. To handle perfect slot alignment, the slot overflowing scheme is developed in [8], where each active slot overflows by δ, a small amount that is sufficient to receive a beacon from another node. However, their approach cannot be applied in our context as it requires that active slots are separated by inactive slots to allow slot overflow, but in our context a node remains active in each slot, making slot overflow impossible.

Motivated by the above argument, we devise the following beacon scheduling scheme.

- Consider node i in slot t, we call slot t a p-slot if $e_t^i = 0$ and $t \bmod p_i < N_i$, i.e., the condition of

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

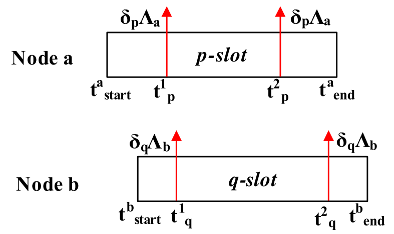

Fig. 7. Illustration of our beacon scheduling and the resulting mutual discovery.

the first line of (1) is satisfied; in the same way we define the q-slot. If the condition of the third line of (1) holds, the node randomly chooses between a p-slot and a q-slot. Recall the proof of Theorem 2, given any pair of neighbors a and b, there must exist an overlap between a p-slot of a and a q-slot of b and between a q-slot of a and a p-slot of b.

- At each p-slot of duration Λ_i, node i sends two beacons, one beacon scheduled $\delta_p\Lambda_i$ ($\delta_p < \frac{1}{2}$) after the beginning of the slot and the other scheduled $\delta_p\Lambda_i$ before the end of the slot, as illustrated in Figure 7. The beacon schedule in the q-slots proceeds in the same way with the parameter δ_q ($\delta_q < \frac{1}{2}$).

The following theorem formally proves that the proposed beacon scheduling mechanism can guarantee mutual discovery under arbitrary clock drift.

Theorem 4: Consider a pair of neighbor nodes a and b whose slot durations are Λ_a and Λ_b. Without loss of generality, assume $\Lambda_a \geq \Lambda_b$. Our beacon scheduling mechanism can guarantee mutual discovery between a and b if $\delta_p > \delta_q$ and the following condition holds:

$$\frac{\Lambda_b}{\Lambda_a} \geq \max\left\{\frac{1}{2}, \frac{\delta_p}{1-\delta_q}, \frac{1-2\delta_p}{1-\delta_q}\right\}. \tag{2}$$

Proof: Recall Theorem 2 and Theorem 3, there exists an overlap of at least $\frac{\Lambda_b}{2}$ between a p-slot of a and a q-slot of b or between a p-slot of b and a q-slot of a. Without loss of generality, assume that there exists an overlap of at least $\frac{\Lambda_b}{2}$ between a p-slot of a and a q-slot of b, as illustrated in Figure 7. Let t^i_{start} and t^i_{end} ($i \in \{a, b\}$) denote the starting and end time of the p-slot of a and the q-slot of b. Let t^x_p, t^x_q ($x = 1, 2$) denote the time when a and b transmit their p-beacons and q-beacons. Without loss of generality, assume $t^a_{start} = 0$. In this context, we have $t^a_{end} = \Lambda_a$, $t^b_{end} = t^b_{start} + \Lambda_b$, $t^1_p = \delta_p\Lambda_a$, $t^2_p = (1-\delta_p)\Lambda_a$, $t^1_q = t^b_{start} + \delta_q\Lambda_b$, $t^2_q = t^b_{start} + (1-\delta_q)\Lambda_b$.

Since the overlap between the two slots in Figure 7 is at least $\frac{\Lambda_b}{2}$, we can bound t^b_{start} as follows.

- If $t^b_{start} \geq 0$, it must hold that $t^a_{end} - t^b_{start} \geq \frac{\Lambda_b}{2}$. Hence $t^b_{start} \leq t^a_{end} - \frac{\Lambda_b}{2} = \Lambda_a - \frac{\Lambda_b}{2}$.
- If $t^b_{start} < 0$, it must hold that $t^b_{end} - t^a_{start} \geq \frac{\Lambda_b}{2}$. Hence $t^b_{start} = t^b_{end} - \Lambda_b \geq t^a_{start} - \frac{\Lambda_b}{2} = -\frac{\Lambda_b}{2}$.

We then prove the theorem by distinguishing four cases: (1) $0 \leq t^b_{start} \leq t^1_p$, (2) $t^1_p < t^b_{start} \leq \Lambda_a - \frac{\Lambda_b}{2}$, (3) $-t^1_q \leq t^b_{start} < 0$, and (4) $-\frac{\Lambda_b}{2} \leq t^b_{start} < -t^1_q$. We detail the proof of the first case. The proofs of other cases follow similarly and are thus omitted.

To prove the first case $0 \leq t^b_{start} \leq t^1_p$. We first prove b can hear at least one p-beacon of a. Since $\delta_p < \frac{1}{2}$, under the condition (2) we have

$$t^b_{start} \leq t^1_p = \delta_p\Lambda_a \leq \frac{\Lambda_a}{2} \leq \Lambda_b < t^b_{start} + \Lambda_b = t^b_{end}.$$

That is, $t^1_p \in [t^b_{start}, t^b_{end})$. Hence b can hear the first p-beacon of a if it does not collide with one of the q-beacons sent by b. Note that under the condition (2)

$$t^2_q = t^b_{start} + (1-\delta_q)\Lambda_b \geq (1-\delta_q)\Lambda_b \geq \delta_p\Lambda_a = t^1_p,$$

the first p-beacon of a cannot collide with the second q-beacon of b. We hence only need to consider the case where the first p-beacon of a collides with the first q-beacon of b, i.e., $t^1_p = t^1_q$. In this case, we have $t^b_{start} = \delta_p\Lambda_a - \delta_q\Lambda_b$. Recall $\delta_q < \delta_p$ and $\Lambda_b \leq \Lambda_a$, under the condition (2) it holds that

$$\begin{aligned} t^2_q &= t^b_{start} + (1-\delta_q)\Lambda_b = \delta_p\Lambda_a + (1-2\delta_q)\Lambda_b \\ &< (1-\delta_p)\Lambda_a = t^2_p, \\ t^b_{end} &= t^b_{start} + \Lambda_b = \delta_p\Lambda_a + (1-\delta_q)\Lambda_b > (1-\delta_p)\Lambda_a = t^2_p. \end{aligned}$$

That is, $t^2_p \in (t^2_q, t^b_{end})$. Hence the second p-beacon of a can be successfully received by b between t^2_q and t^b_{end}. Therefore, b can successfully receive at least one p-beacon of a. Similarly, we can prove that at least one q-beacon of b can be successfully received by a, thus leading to mutual discovery. □

It is insightful to note that the condition (2) is easy to satisfy in practice. To this end, suppose that we choose $\delta_p = 0.2$ and $\delta_q = 0.1$, under which (2) becomes $\frac{\Lambda_b}{\Lambda_a} \geq \frac{2}{3}$. By using clock model introduced in Sec. VI, we have $\frac{\Lambda_b}{\Lambda_a} \geq 1 - \Delta\rho_{max}T_{syn}$, where T_{syn} denotes the maximum interval between two consecutive execution of network clock synchronization. In practice, $\Delta\rho_{max} = O(10^{-6})$. If T_{syn} is set no shorter than 3×10^5 seconds, i.e., the network is synchronized at least once each 3.5 days, we can ensure that (2) is satisfied.

The devised beacon scheduling mechanism can be further adapted in the collision-prone environment when the network scales. In this context, more than one simultaneously transmitted beacons lead to collision and thus cannot be recovered at the receiver. To limit collision, we can desynchronize p-beacons and q-beacons by adding a small random time drift to δ_p and δ_q. Note that in such context, discovery delay cannot be bounded due to collision. The utility of our neighbor discovery algorithm is to ensure that any pair of neighbors will eventually steer their antennas toward each other, without which discovery can never be achieved.

VIII. Balancing Worst-Case and Average Discovery Delay

In the previous section, we have shown that the worst-case discovery delay of our algorithm is bounded by $L\max\{p_aq_b, p_bq_a\}$ and this bound is very tight, i.e., there are extremely unlucky cases where discovery cannot be achieved before the worst-case delay. On the other hand, it is easy to see that a purely random strategy where each node points its antenna to a random direction each slot leads to an average delay of N_aN_b even in such extremely unlucky cases.

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

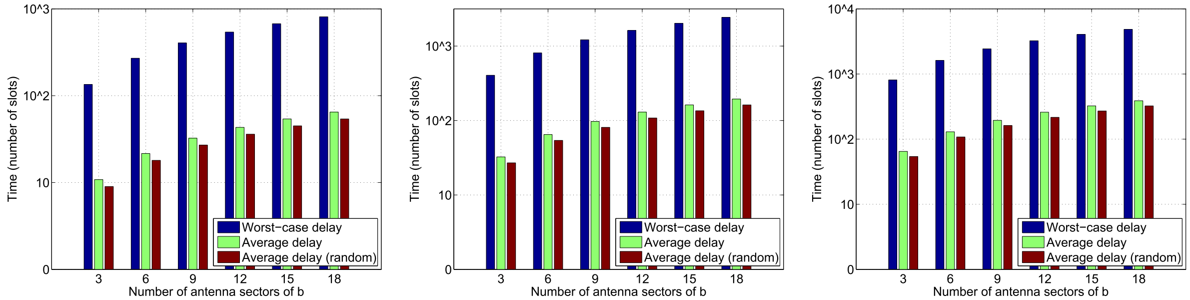

Fig. 8. Discovery delay comparison between our algorithm and the random discovery strategy with aligned slots under fixed N_a and varying N_b: left $N_a = 3$, middle $N_a = 9$, right $N_a = 18$.

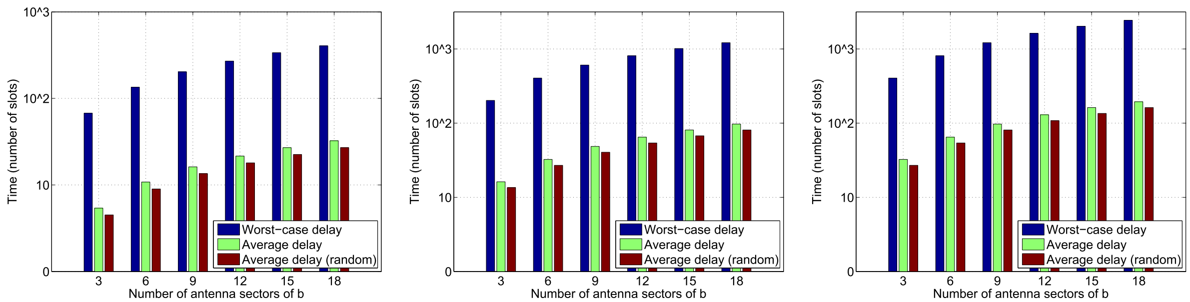

Fig. 9. Discovery delay comparison between our algorithm and the random discovery strategy with drifted slots under fixed N_a and varying N_b: left $N_a = 3$, middle $N_a = 9$, right $N_a = 18$.

However, the worst-case discovery delay of any random discovery strategy cannot be bounded. Generally, random or probabilistic neighbor discovery algorithms usually perform well in the average case by limiting the expected discovery delay, with the main drawback being the lack of performance guarantee in terms of worst-case discovery delay. The following question naturally arises: *How to decrease the average discovery delay of our algorithm in those extremely unlucky cases while still ensuring a bounded worst-case delay*.

One solution to trade-off the worst-case and the average discovery delay is to interleave random slots into the antenna scan sequence. To that end, noticing that the antenna scan sequence in our neighbor discovery algorithm (i.e., eq (1)) naturally has a set of random slots where $N_i \leq t \bmod p_i$ (when $e_t^i = 0$) and $N_i \leq t \bmod q_i$ (when $e_t^i = 1$), during which each node i randomly points its antenna. By tuning p_i and q_i, we can set a desired number of random slots while still guaranteeing that the worst-case discovery delay is bounded. Specifically, larger values of p_i and q_i leads to more random slots and hence decreases the average discovery delay, with the increase in the worst-case delay. By tuning p_i and q_i, the worst-case and the average discovery delay can be traded off.

IX. Performance Evaluation

In this section, we conduct a suite of numerical analysis and simulations to illustrate the theoretical results established in previous analysis and to evaluate the performance of our neighbor discovery algorithm in several typical settings.

A. Numerical Experiments: Pair-Wise Neighbor Discovery

We start with a set of numerical experiments to study the baseline scenario of pair-wise discovery between a pair of neighbor nodes a and b. Specifically, we implement our algorithm using Matlab and trace the discovery delay for different antenna configurations of a and b, i.e., different combinations of (N_a, N_b). The relative positions of a and b, represented by (h_a, h_b) is also randomly generated. The initial clock drift between a and b is randomly generated from $[0, 1000]$ slots with the relative rate of clock drift ρ_{ab} is randomly generated from $[1 - 10^{-6}, 1 + 10^{-6}]$ which corresponds to a typical wireless node setting. $\delta_p = 0.2$ and $\delta_q = 0.1$. Both a and b have an ID of 8 bits randomly attributed to them. The results in our experiments correspond to an average of 10000 runs with confidence interval 95%.

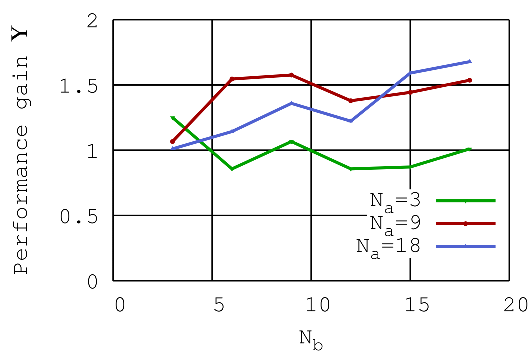

Fig. 10. Worst-case discovery delay comparison between the baseline and the improved algorithms.

Figure 8 and 9 trace the worst-case and average discovery delay of our algorithm for 2 cases: (1) slots of a and b are aligned with identical slot duration, (2) slots of a and b are not aligned and follow the model described above. For comparison, we also trace the average discovery delay of the random strategy where each node steers its antenna at

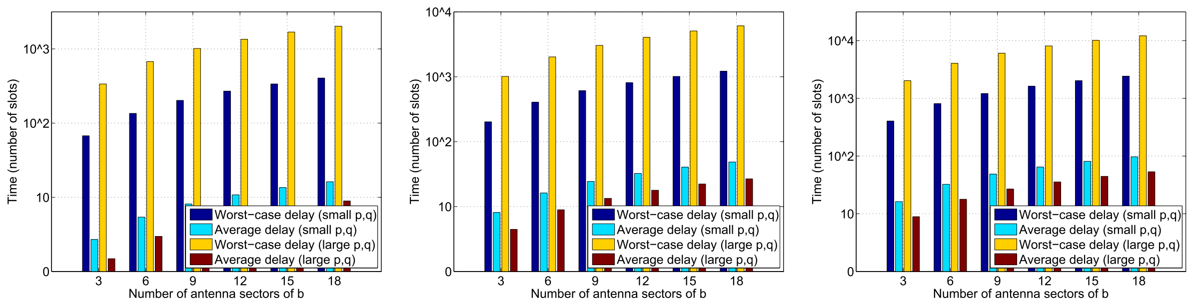

Fig. 11. Trading off worst-case and average discovery delay under fixed N_a and varying N_b: left $N_a = 3$, middle $N_a = 9$, right $N_a = 18$.

a random direction each slot. We cannot trace the worst-case delay of the random strategy because in some cases, discovery cannot be achieved within the simulation duration which is set to 5 times the worst-case delay of our discovery algorithm. From the results, we make the following observations.

- In all the simulated cases, the worst-case of our algorithm is bounded, which is in accordance of our theoretical result established in Theorem 2. However, the worst-case delay of random algorithms is not bounded. Moreover, the results also demonstrate the the worst-case and average discovery delay trade-off between the random strategy and the deterministic one as ours.
- The performance with non-aligned slots outperforms that with aligned slots, due to the optimisation 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 2 slots is higher.

Our simulation results seem to favor a well-designed deterministic strategy as we see limited performance loss in terms of average delay with the advantage of having strict worst-case delay bound.

We then proceed to study the performance of the improved algorithm analyzed in Sec. V-D. To this end, we trace the performance gain in terms of the ratio between the worst-case discovery delay of the baseline algorithm and the improved algorithm for drifted slots, defined as Υ. The results are shown in Figure 10. We can see from the results that the improved algorithm can achieve shorter discovery delay when the system scales, i.e., N_a and N_b are not too small. This is due to the requirement in the baseline algorithm that q_i needs to be power-multiples of two and may be significantly larger than N_i, in the worst case $q_i \simeq 2N_i$; while in the improved algorithm, q_i is asymptotically close to N_i. Note that the baseline algorithm outperforms the improved algorithm in some cases when N_a or N_b is 3; this is because the improved algorithm requires p_i is set to a prime number larger than 3.

We then investigate trading off worst-case and average discovery delay by incorporating the mechanism proposed in Section VIII. To that end, we pick settings where a and b can only discover each other almost with the worst-case discovery delay. We implement the mechanism proposed in Section VIII and trace the resulting trade-off between the worst-case and average discovery delay in Figure 11. Specifically, we implement two settings: (1) Small p and q, in this setting, p and q are chosen as the smallest eligible values larger than $2N$; (2) Large p and q, in this setting, p and q are chosen as the smallest eligible values larger than $10N$. We run the experiment under non-aligned slots using the improved algorithm. The results in Figure 11 clearly demonstrate the trade-off between worst-case and average discovery delay: with larger p and q, the worst-case discovery delay is more important, while the average delay is less. The trade-off can thus be parameterised to satisfy specific application requirement by tuning p and q.

B. Simulation Analysis: Network-Wide Neighbor Discovery

In this subsection, we further evaluate our algorithm under realistic conditions by performing a serious simulations in 60GHz networks, where directional antennas are particularly attractive in to ensure high transmission quality and acquire sufficient link budget to cater Gbps data rate.

In order to have an accurate simulation environment for investigating 60 GHz networks, we have developed an NS-2 extension that incorporates new features such as effective SINR calculation for modeling co-channel interference and support of directional antennas by following the development in [26], which is is based on an NS-2 extension called IEEE 802.11 Overhaul [27] which addresses the shortcomings in the default NS-2 MAC and PHY modules. Specifically, we configure the PHY layer parameters based on the High Speed Interface (HSI) mmWave PHY modes in IEEE 802.15.3c standard draft to simulate bi-directional high speed communications in 60 GHz bands [28]. We also consider three realistic antenna models of Ansys Corporation [29] also used in [26] with beamwidth of 90, 60, 15 degree and a maximum gain of 8.6, 12.1 and 17.7 dB, referred to as ant-90, ant-60 and ant-15, respectively. These realistic antenna propagation patterns allow us to capture the behaviors of realistic directional antennas with side and back lobes.

We simulate a 60 GHz wireless network composed of 100 static nodes randomly deployed in a $100m \times 100m$ square. We use the TwoRayGround propagation model with a path loss exponent of 3.5, which is commonly used to model signal propagation in the 60 GHz Non Line-of-Sight (NLOS) indoor environment [28]. We set the clock skew of each node to be randomly distributed within [-10ppm, 10ppm], given that a skew of 10ppm corresponds to typical crystal clocks operating at extremes of their temperature specification. $\delta_p = 0.2$ and $\delta_q = 0.1$. Each node is randomly attributed an antenna among ant-90, ant-60 and ant-15. Each node steers

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

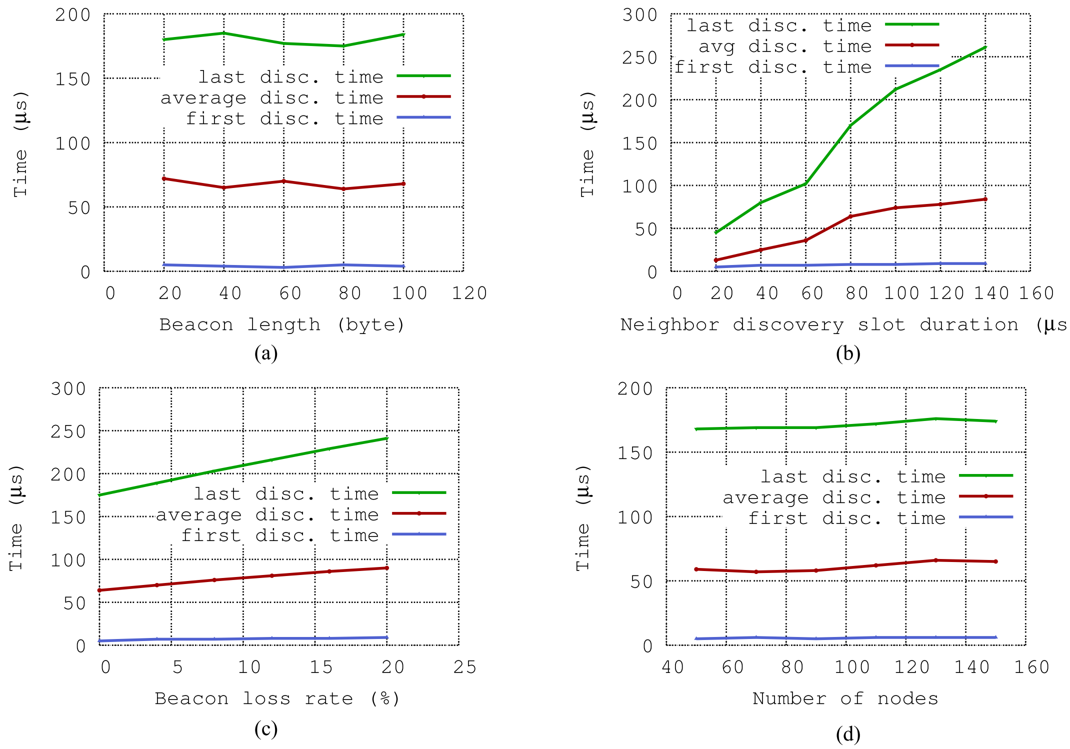

Fig. 12. Simulation results for network-wide neighbor discovery: (a) upper left, (b) upper right, (c) lower left, (d) lower right.

its antenna toward a direction for $80\mu s$ before moving to another direction if not specified otherwise, i.e., the discovery slot duration is $80\mu s$ based on local clock. The results in our simulations correspond to an average of 10000 runs with 95% confidence interval.

Throughout our simulation, we focus on the following performance metrics: (1) first discovery delay, the delay until the first neighbor is discovered; (2) average discovery delay; (3) last discovery delay, the delay until the last neighbor is discovered. These metrics are also used in [30] for simulation. As justified in [30], the fast discovery of the first node is motivated by emergency applications, where the transmission of a message needs to be performed as soon as possible and consequently to a next hop. The average discovery time characterizes the average performance of any neighbor discovery algorithm. The last discovery delay is important in applications where the discovery of all neighbors of a node is required.

We first study the impact of the discovery beacon length on the discovery performance of our algorithm. Figure 12(a) traces the discovery delay under different beacon length. From the results, we observe that the beacon length does not have significant impact on the discovery performance. This is due to the following two reasons. First, our algorithm can ensure an overlap of at least half of a slot duration, which is sufficient to receive discovery beacons of different length. Secondly, the result also reflects the fact that in the simulated cases, collisions among beacons only have limited impact on the discovery performance because beacons are relatively short compared to the slot duration, and antennas are directional, which limits the collision probability.

We then study the impact of the neighbor discovery slot duration on the performance. As illustrated in Figure 12(b), the last and average discovery delay increase almost linearly w.r.t. the neighbor discovery slot duration. Naturally, short neighbor discovery slot duration facilitates the process of neighbor discovery, but at the price of more frequent beacon transmission. We further trace the impact of beacon loss on the discovery performance. To this end, we simulate different beacon loss rate and trace the corresponding discovery delay. The results, as plotted in Figure 12(c), demonstrate that the beacon loss has more impact on the last discovery delay. This is intuitive as the loss of a beacon does not increase significantly the first discovery delay because there are other nodes that can be discovered shortly after, but the last discovery delay may be largely increased as the concerned node needs to wait for the next beacon which may take a while to arrive.

We complete the simulation by investigating the impact of node density on the discovery performance. We thus vary the number of nodes from 50 to 150 and trace the discovery delay in Figure 12(d). We observe from the results that the discovery delay does not vary significantly w.r.t. the node density (or, the number of neighbors). This again reflects that the impact of beacon collisions in our simulation is not pronounced as nodes are equipped with directional antennas and clocks are not synchronized which also desynchronises the beacon transmission.

X. Conclusion

We have formulated and studied the oblivious neighbor discovery problem. We have established the performance bound of any neighbor discovery algorithm achieving oblivious discovery. Guided by the theoretical results, we have designed an oblivious discovery algorithm and proved that it achieves guaranteed oblivious discovery with order-minimal worst-case discovery delay in the asynchronous and heterogeneous environment. We have then studied how our algorithm can be configured to achieve a desired trade-off between average and worst-case performance. In future research, we plan to investigate the energy-constrained case where nodes stay in dormant state most of time while only wakes up periodically.

REFERENCES

[1] L. Chen, Y. Li, and A. V. Vasilakos, "Oblivious neighbor discovery for wireless devices with directional antennas," in *Proc. INFOCOM*, Apr. 2016, pp. 1–9.

[2] R. Ramanathan, J. Redi, C. Santivanez, D. Wiggins, and S. Polit, "Ad hoc networking with directional antennas: A complete system solution," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 3, pp. 496–506, Mar. 2005.

[3] R. A. Santosa, B.-S. Lee, C. K. Yeo, and T. M. Lim, "Distributed neighbor discovery in ad hoc networks using directional antennas," in *Proc. IEEE CIT*, Sep. 2006, p. 97.

[4] 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.

[5] 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.

[6] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. Sensys*, 2008, pp. 71–84.

[7] 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.

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

[9] T. Meng, F. Wu, and G. Chen, "On designing neighbor discovery protocols: A code-based approach," in *Proc. INFOCOM*, 2014, pp. 1689–1697.

[10] L. Chen, K. Bian, and M. Zheng, "Heterogeneous multi-channel neighbor discovery formobile sensing applications: theoretical foundationand protocol design," in *Proc. MobiHoc*, 2014, pp. 307–316.

[11] L. Chen *et al.*, "On heterogeneous neighbor discovery in wireless sensor networks," in *Proc. IEEE INFOCOM*, May 2015, pp. 693–701.

[12] L. Chen and K. Bian, "Neighbor discovery in mobile sensing applications: A comprehensive survey," *Ad Hoc Netw.*, vol. 48, pp. 38–52, Sep. 2016.

[13] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," in *Proc. INFOCOM*, Mar. 2005, pp. 2502–2512.

[14] G. Jakllari, W. Luo, and S. V. Krishnamurthy, "An integrated neighbor discovery and MAC protocol for ad hoc networks using directional antennas," *IEEE Trans. Wireless Commun.*, vol. 6, no. 3, pp. 1114–1124, Mar. 2007.

[15] Z. Zhang and B. Li, "Neighbor discovery in mobile ad hoc self-configuring networks with directional antennas: Algorithms and comparisons," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1540–1549, May 2008.

[16] J. S. Park, S. W. Cho, M. Y. Sanadidi, and M. Gerla, "An analytical framework for neighbor discovery strategies in ad hoc networks with sectorized antennas," *IEEE Commun. Lett.*, vol. 13, no. 11, pp. 832–834, Nov. 2009.

[17] X. An, R. V. Prasad, and I. Niemegeers, "Impact of antenna pattern and link model on directional neighbor discovery in 60 GHz networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 5, pp. 1435–1447, May 2011.

[18] J. Du, E. Kranakis, O. M. Ponce, and S. Rajsbaum, "Neighbor discovery in a sensor network with directional antennae," in *Proc. Algosensors*, 2011, pp. 57–71.

[19] J. Ning, T.-S. Kim, S. V. Krishnamurthy, and C. Cordeiro, "Directional neighbor discovery in 60 GHz indoor wireless networks," *Perform. Eval.*, vol. 68, no. 9, pp. 897–915, 2011.

[20] H. Cai, B. Liu, L. Gui, and M.-Y. Wu, "Neighbor discovery algorithms in wireless networks using directional antennas," in *Proc. ICC*, Jun. 2012, pp. 767–772.

[21] H. Cai and T. Wolf, "On 2-way neighbor discovery in wireless networks with directional antennas," in *Proc. IEEE INFOCOM*, May 2015, pp. 702–710.

[22] W. Xiong, B. Liu, and L. Gui, "Neighbor discovery with directional antennas in mobile ad-hoc networks," in *Proc. IEEE GLOBECOM*, Dec. 2011, pp. 1–5.

[23] R. Murawski *et al.*, "Neighbor discovery in wireless networks with sectored antennas," *Ad hoc Netw.*, vol. 10, pp. 1–18, Jan. 2012.

[24] K. Bian and J.-M. Park, "Maximizing rendezvous diversity in rendezvous protocols for decentralized cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1294–1307, Jul. 2013.

[25] I. Niven, H. S. Zuckerman, and H. L. Mongomery, *An Introduction to the Theory of Numbers*. Hoboken, NJ, USA: Wiley, 1991.

[26] W. Zhou, S. Nandagopalan, and D. Qiao, "A simulation study of CSMA/CA performance in 60 GHz WPANs," in *Proc. IEEE GLOBECOM*, Dec. 2009, pp. 1–6.

[27] Q. Chen *et al.*, "Overhaul of IEEE 802.11 modeling and simulation in ns-2," in *Proc. ACM MSWiM*, 2007, pp. 159–168.

[28] *IEEE P802.15.3 High Rate (HR) Task Group (TG3) for Wireless Personal Area Networks (WPANs)*, accessed on Feb. 2016. [Online]. Available: http://www.ieee802.org/15/pub/TG3.html

[29] *Ansys Corporation*, accessed on Feb. 2016. [Online]. Available: www.ansys.com/

[30] N. Karowski, A. C. Viana, and A. Wolisz, "Optimized asynchronous multichannel discovery of IEEE 802.15.4-based wireless personal area networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1972–1985, Oct. 2013.

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

Yong Li (M'09–SM'16) received the B.S. degree in electronics and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. He is currently a Faculty Member with the Department of Electronic Engineering, Tsinghua University. His papers have total citations of more than 2300 (six papers exceed 100 citations, according to Google Scholar). Among them, eight are ESI Highly Cited Papers in Computer Science, and four received conference Best Paper (runner-up) Awards. He received the IEEE 2016 ComSoc Asia-Pacific Outstanding Young Researchers. He has served as the General Chair, the TPC Chair, and a TPC Member for several international workshops and conferences, and he is on the Editorial Board of three international journals.

Athanasios V. Vasilakos is currently a Professor with the Luleå University of Technology, Sweden. He served or is serving as an Editor for many technical journals, such as the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON CYBERNETICS, the IEEE TRANSACTIONS ON NANOBIOSCIENCE, and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.