

# Energy-Efficient Neighbor Discovery in Mobile Ad Hoc and Wireless Sensor Networks: A Survey

Wei Sun, *Student Member, IEEE*, Zheng Yang, *Member, IEEE*,  
Xinglin Zhang, *Student Member, IEEE*, and Yunhao Liu, *Senior Member, IEEE*

**Abstract**—Due to slow advance in battery technology, power remains a bottleneck to limit wide applications of mobile ad hoc and wireless sensor networks. Among all extensive studies on minimizing power consumption, neighbor discovery is one of the fundamental components focusing on communication and access. This work surveys research literature on *neighbor discovery protocols (NDPs)*. In general, they can be roughly classified by four underlying principles: randomness, over-half occupation, rotation-resistant intersection, and coprime cycles. The Birthday protocols act as representatives of NDPs using randomness, in which a node decides to listen, transmit, or sleep with probabilities. The original idea of over-half occupation is to be active over at least half of each period, though several refinements have been proposed to decrease its high duty cycle. Methods of rotation-resistant intersection formulate the problem of discovery using combinatorial characteristics of discrete time slots, and guarantee discovery at least once per period. Moreover, neighbor discovery can also be guaranteed within a worst-case bound, as shown by methods adopting coprime cycles. In this paper, we elaborate on these ideas and present several representative protocols, respectively. In particular, we give an integrative analysis of deterministic protocols via a generic framework. A qualitative comparison incorporating multiple criteria and a quantitative evaluation on energy efficiency are also included. Finally, we point out promising research directions towards energy-efficient neighbor discovery.

**Index Terms**—Neighbor discovery protocols, energy efficiency, mobile ad hoc networks, wireless sensor networks.

## I. INTRODUCTION

WITH the rapid proliferation of miniaturized wireless devices such as PDAs, smartphones, and sensors, mobile ad hoc networks (MANETs) [1] and wireless sensor networks (WSNs) [2] have attracted significant interest and progressed substantially during the past decade. On one hand, MANETs consist of devices that are autonomously self-organizing, whereas most of today's wireless communication depends on expensive, centrally deployed hub-and-spoke networks. Such a large degree of freedom and self-organizing capacities make them especially suitable for environments and situations in which pre-defined network infrastructure goes out of service or does not exist at all. One typical application scenario would be

disaster areas where communication between rescue workers, search teams, and medical staff needs to be established in spite of the destruction of network infrastructure. Besides, their applications in vehicles have given birth to the development of vehicular ad hoc networks (VANETs) [3], where moving vehicles communicate with each other to enhance road safety [4], [5] and transportation efficiency [6], [7]. On the other hand, WSNs are composed of low-power wireless sensors with data collection, processing, and transmission capacities. They are particularly attractive to those applications that need to collect and analyze environmental data (*e.g.*, temperature, humidity, or concentration of carbon dioxide) in a large area. Traditional applications of WSNs include habitat monitoring [8], environmental monitoring [9], and sea monitoring [10]. In addition, newly emerging applications such as hiker logging [11], object tracking [12], and social networking [13], are penetrating into our daily life.

Driven by the need for connectivity maintenance [14] and context awareness [15], discovery among neighboring nodes or *neighbor discovery* for short, serves as a prerequisite for both types of networks. Only after an initial discovery can a node set up communications with others. A trivial solution to the problem is to keep radio on all the time such that neighboring nodes can discover each other shortly. However, the crux lies in the power scarcity: nodes are generally battery-powered and current battery capacities cannot afford always-on radio for a network life time system operators may expect. It is known that idle listening dominates the system power budget [16]. As a compromise, nodes have to turn radio on/off from time to time, with the portion of time in the ON state characterized by *duty cycle*. Despite the deduction of power consumption, this leads to the uncertainty in discovery latency. Typically, duty cycle and discovery latency are two key metrics by which the energy efficiency of neighbor discovery is evaluated. It is desirable to have a low duty cycle and a low discovery latency simultaneously, but they are in conflict with one another: a lower duty cycle usually leads to a higher discovery latency, and vice versa. It is this trade-off that makes energy-efficient neighbor discovery challenging.

In this paper, we review research literature on neighbor discovery protocols (NDPs). Based on underlying design principles, they can be roughly classified into four categories: randomness, over-half occupation, rotation-resistant intersection, and coprime cycles. In the context of neighbor discovery, randomness takes effect in making nodes listen, transmit, or sleep with probabilities. By leveraging the idea of the

Manuscript received May 30, 2103; revised November 6, 2013.

W. Sun and X. Zhang are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: wsunaa@cse.ust.hk, xzhangae@cse.ust.hk).

Z. Yang and Y. Liu is with the School of Software, Tsinghua National Laboratory of Information Science and Technology, Tsinghua University, Beijing, P. R. China (e-mail: yang@greenorbs.com, yunhao@greenorbs.com).  
Digital Object Identifier 10.1109/SURV.2013.012414.00164

Birthday Paradox, fast neighbor discovery in the average case can be achieved while keeping its duty cycle low. Over-half occupation is a simple idea to guarantee discovery when a node's radio is ON over at least half of a period. Considering its high duty cycle requirement, several proposals have been developed to ease the pain. Combinatorial characteristics of discrete time slots is applied to methods using rotation-resistant intersection, in which neighbor discovery can be mathematically abstracted as a block design problem. Finally, discovery can also be guaranteed with coprime cycles thanks to the Chinese Remainder Theorem.

While elaborating on these principles, we present several representative protocols, such as the Birthday protocols [17], Quorum [18], Disco [19], U-Connect [20], and Searchlight [21]. The protocols, except the Birthday ones, fit into the domain of deterministic protocols. We show that despite different underlying principles, they can be incorporated into a generic framework under symmetric duty cycles. We also make a qualitative comparison among the protocols including multiple criteria. Furthermore, the performance on energy efficiency under different settings is evaluated through simulation. Finally, we shed light on future research directions towards energy-efficient neighbor discovery.

**Contribution.** We highlight the contributions of this paper as follows:

- **A taxonomy and an analysis of NDPs.**
- **A qualitative comparison among representative NDPs.**
- **A quantitative evaluation of representative NDPs on energy efficiency.**
- **An insight into future research directions.**

Prior to this, several surveys on closely related topics can be found at [22], [23], [24]. Sreekanth *et al.* [22] focused more on discoveries after an initial one, and included very limited analysis of existing NDPs. Roslin *et al.* [23] targeted at topology control in WSNs, taking neighbor discovery as given. The work of Galluzzi *et al.* [24] introduced discovery mechanisms in WSNs and is thus most close to ours, but we make a step further to expose the interconnection among the deterministic protocols via the framework and to manifest the energy efficiency of NDPs in greater detail.

**Organization.** In the following, we present preliminary knowledge of neighbor discovery for readers unfamiliar with this topic. Section III explores concrete NDPs under the four principles and analyzes the deterministic protocols in depth. A comparison and an evaluation of representative protocols is given in Section IV. Future research directions are listed in Section V. Section VI concludes this survey.

## II. PRELIMINARIES

Before delving into specific NDPs, we present several basic and common concepts used in the literature for general readers. Proper terms and jargons are first introduced in Section II-A. In order to guide the design and evaluate the performance of NDPs, metrics need to be carefully selected according to various application requirements. Section II-B presents two fundamental criteria for performance evaluation of NDPs. We briefly examine three general strategies for neighbor discovery, namely synchronization, MAC-based protocols, and NDPs, in

Section II-C. A widely adopted time-slotted model is illustrated in Section II-D. Last but not least, common assumptions employed by most NDPs are listed in Section II-E.

### A. Terminologies

For power conservation, a node turns off its radio between communications. When its radio is off, it can neither send messages nor respond to information queries. Other nodes within its transmission range are thus unable to detect its presence at the moment. We therefore say a node is *idle* or in *idle state* when its radio is OFF. On the contrary, a node is *active* or in *active state* when its radio is ON. During an active state, it may transmit, listen, or alternate between both. Two nodes are defined to be *neighbors* if they hear beacons from each other, *i.e.*, they are within the transmission range of each other when both are active.

As nodes switch between active and idle states, a measurement called *duty cycle* is introduced to describe the time division between the two. Formally, duty cycle is the fraction of time a node spends on active states. For example, a duty cycle of 2% indicates that a node spends 2% of the overall observation time in active states. By intuition, a larger duty cycle results in shorter life time due to greater power consumption in the radio module. On the other hand, a smaller duty cycle extends the life time to some extent, but at the cost of increasing the probability that the node is undiscovered.

*Discovery latency* is another term often discussed together with duty cycle. As suggested literally, discovery latency measures how long a node needs to wait until it discovers its neighbor. In response to two cases of discovery, it is calculated in two ways. For initial discovery, latency accumulation starts when nodes come into the transmission range of each other, and ends when they receive a beacon from one another. And for subsequent discoveries [25], discovery latency is defined as the interval between two consecutive discoveries. Since it is more challenging to achieve a fast initial discovery (a node has no idea about its neighbor at all) than subsequent discoveries, most work emphasizes on the former case.

Besides duty cycle and discovery latency, synchronicity and symmetricity are the two features we have to consider for neighbor discovery. According to the necessity for time synchronization, neighbor discovery strategies are categorized into *synchronous* and *asynchronous* ones. Besides, strategies that require the same scheduling pattern of state transition (*i.e.*, the same duty cycle) are called *symmetric*, or *asymmetric* otherwise.

### B. Performance Metrics

Among the extensive research literature on neighbor discovery, two performance indicators, namely duty cycle and discovery latency, are usually of top concern. While duty cycle is well-defined with little ambiguity, the statistical properties of discovery latency may differ in various application contexts. Generally, the *mean* or *maximum* latency is adopted by most research work. However, some application scenarios (*e.g.*, highly dynamic mobile networks) may take the latency of initial discovery of newcomers as the only concern. In any

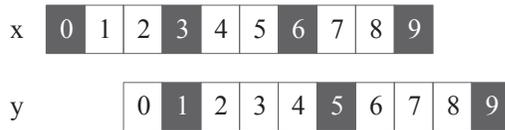


Fig. 1. Aligned slots and mutual discovery.

case, it is desirable to have a low duty cycle and a low discovery latency.

However, it is not difficult to observe a trade-off between duty cycle and discovery latency. A lower duty cycle usually leads to a higher discovery latency, and vice versa. In fact, from the perspective of energy efficiency, duty cycle corresponds to the “energy” aspect, while discovery latency corresponds to the “efficiency” aspect. Therefore, how to balance these two conflicting metrics becomes the key to achieve energy efficiency. To simplify the discussion, Kandhalu *et al.* proposed a composite metric called the *power-latency product* [20], which is the product of the average power consumption (*i.e.*, duty cycle) with the worst-case discovery latency. In general, it serves as a good metric as when either factor is held constant, the other needs to be minimized. Besides, using a single scalar value resolves the dilemma of comparing two schemes where one adopts a lower duty cycle while the other achieves a lower discovery latency. Note that it may not be universally appropriate for all applications; power-exhausting nodes might put a greater weight on power consumption. However, this is out of the scope of this paper.

For practicability concerns, it is also preferable to work with asynchronous clocks and under asymmetric duty cycles. Global synchronization among all nodes in a sensor network has been demonstrated to be difficult or energy-expensive [26]. Moreover, as nodes may be assigned with tasks of different energy requirements or left with different energy budgets, limiting all nodes to run at a given duty cycle is too restrictive and inefficient.

### C. General Strategies

Based on the above-mentioned metrics, we examine general strategies to fulfill neighbor discovery. On the whole, they can be roughly classified into three categories: time synchronization, MAC design, and neighbor discovery protocols.

1) *Time synchronization*: The solution to ensuring neighbor discovery is trivial once clocks can be synchronized, through GPS [27], [28] for example. In this case, nodes can discover each other as long as they deliberately switch to active state during the same time period. Apparently, the discovery latency would be low. However, synchronization by itself is often too expensive to be affordable in real deployment. This necessitates a mechanism that ensures discovery between two nodes while supporting *asynchronous* clocks.

2) *MAC design*: One strategy that supports asynchronous clocks is through medium access control (MAC) protocols, such as B-MAC [29] and S-MAC [30]. Achieving neighbor discovery as a byproduct, such protocols employ low power listening (LPL), a technique by which a node goes to idle state when no activity is detected. The problem of this strategy is that they often assume symmetric idle/active patterns among

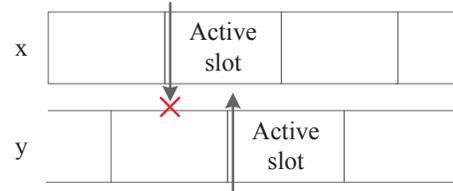


Fig. 2. Unaligned slots may result in failure of mutual discovery.

all nodes. Due to the differences in energy budgets or task requirements, nodes usually prefer *asymmetric* duty cycles such that the lifetime would be prolonged or the performance would be upgraded.

3) *Neighbor discovery protocols*: In response to the asynchronous and asymmetric requirements, a class of dedicated neighbor discovery protocols (NDPs) have evolved. The most well-known probabilistic approach is a family of the birthday protocols [17] where nodes transmit, listen, or sleep with probabilities. Deterministic protocols, on the other hand, schedule active states elaborately using combinatorial characteristics [18], coprime properties [19], [20], *etc.* These protocols typically relieve themselves from synchronization, and support asymmetric duty cycles. Therefore, we limit our discussion to NDPs for the remaining of this paper, and will examine them in detail in the next section.

### D. Time-Slotted Model

For almost all NDPs, a *time-slotted model* is usually adopted for analysis simplicity. Continuous time is separated into discrete interval called *slot*, whose length should be enough for basic communication (or for neighbor discovery at least). A node decides to be active or idle in any given slot. In an active slot, a node may first transmit a beacon to claim its presence, and spend the rest of the time listening for beacons of others. For presentation convenience, we usually refer to a time slot by the value of a fictional counter. The counter may start counting from 0 when the node is powered on, and increases by one every slot. With this time-slotted model, designing an NDP is equivalent to finding a schedule of active and idle states to minimize discovery latency while keeping the duty cycle low.

As an easy start, we discuss the use of the time-slotted model by assuming *slot alignment*. In other words, we suppose that slot boundaries are aligned for all nodes. With such simplification, the formulation of neighbor discovery is defined as follows. Suppose nodes  $x$  and  $y$  are within the transmission range of  $k$ , but have not yet discovered, each other. If  $x$  is active at slot  $k_1$  and  $y$  at slot  $k_2$ , and these two slots (fully) overlap, they discover each other.

Fig. 1 illustrates a slot-aligned scenario. Node  $x$  is active at slot 0, 3, 6, and 9, while node  $y$  is at slot 1, 5, and 9. Due to the lack of time synchronization, they have a time displacement of two slots. That is, slot  $k$  ( $k \geq 0$ ) of node  $y$  corresponds to slot  $k+2$  of node  $x$ . We observe that slot 3 of  $x$  and slot 1 of  $y$  are fully overlapped, leading to their first discovery. From then on, they become neighbors of each other.

Though slot alignment significantly lowers the complexity of neighbor discovery, it is almost impractical to achieve in

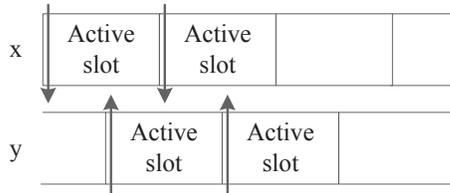


Fig. 3. Add one extra active slot at the forward position.

practice. Slots are rarely aligned as nodes work independently and do not set up a global time reference. Usually there exists displacement among slot boundaries. As a result, one node may not be able to discover the other even if their active slots overlap. This is illustrated in Fig. 2 where arrows stand for beacons. Though node  $x$  can successfully receive the beacon from node  $y$  and recognize  $y$  as its neighbor,  $y$  is unaware of  $x$ 's presence as the beacon from  $x$  arrives at  $y$  when  $y$  is idle.

One straightforward solution to address this problem is to stretch the length of active state (say one extra active slot at the forward or backward position) for sufficient overlap (see Fig. 3). But it is not efficient due to a higher duty cycle and redundant beacons. Alternatively, Dutta and Culler [19] proposed a new way to deal with unaligned slots: a beacon is transmitted at both the start and the end of an active slot. In this case, an adequate overlap (could be small) of active slots suffices for mutual discovery (see Fig. 4). They reported that only in 2% of the trials did the empirical discovery latency exceed the worst-case one obtained through simulation.

#### E. Assumptions for Neighbor Discovery Protocols

Before we investigate NDPs in detail, it will definitely ease the presentation to give some common assumptions held by most of them. These assumptions are made to simplify protocol design, neglecting factors that have minor effect in discovery or are difficult to address by NDPs themselves only.

1) *Bidirectional links*: Most of the research literature on neighbor discovery assumes bidirectional communication links. In other words, all nodes have the same transmission range. Such a communication symmetry simplifies protocol design and analysis, as we only need to focus on how a node can discover the other, leaving the discovery of the inverse direction out of account (slot non-alignment has been addressed before). Note that this assumption may not hold in practical applications; a node  $x$  could take node  $y$  as its neighbor, whereas  $y$  is unaware of  $x$ 's presence due to the shorter communication range of  $x$ . Possible conditions to break this assumption might be power dynamics, obstacle blockage, signal reflection and absorption, *etc.*

2) *No decoding failures*: Usually, we ignore the possibility of decoding failures probably caused by message corruption or signal interference. The rationale behind this assumption is that an effective neighbor discovery protocol as an ongoing process, can guarantee the eventual discovery of neighboring nodes, taking more time than expected if considering decoding failures. Moreover, dealing with decoding failures is the job of MAC protocols for collision avoidance. Pure neighbor discovery protocols can cooperate with MAC protocols in real world applications.

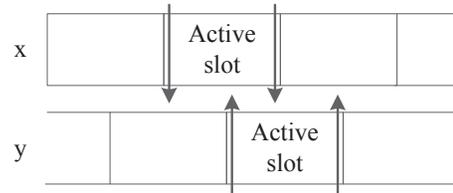


Fig. 4. Two beacon at the beginning and the end of active slots.

3) *Perfect timing*: We assume no warmup delay from idle to active state and no clock drift. Put it strictly, nodes are unable to switch from an idle state to an active state without consuming any time. Frequency oscillator needs to be activated while entering an active state, which introduces a delay in the order of milliseconds [24]. One way to counteract the warmup delay is to start the power-up process at the last part of an idle slot, making it negligible compared with slot duration (typically tens of milliseconds [31], [19]). In addition, we assume that clocks of all nodes run at the exact same pace. In practice, clock hardware is selected for low cost and may deviate from ideal behavior. However, such deviation is also neglected for simplification.

### III. NEIGHBOR DISCOVERY PROTOCOLS

We conduct a taxonomy and present representative NDPs in this section. Despite different features claimed by inventors of a variety of protocols, they take advantage of three basic techniques in designing neighbor discovery protocols. First, randomness can be applied to scheduling active slots. A node decides to be active in a given slot with a probability that is predefined or adjusted dynamically. Second, a node can remain active for a number of consecutive slots to ensure neighbor discovery. One such extreme case is to remain active all the time. Third, explicit patterns of active slots may contribute to limiting the upper bound of discovery time while operating at low duty cycles.

In general, NDPs can be roughly classified based on their underlying principles: randomness, over-half occupation, rotation-resistant intersection, and coprime cycles. These principles will be discussed in detail with representative protocols in subsequent subsections. In particular, deterministic protocols can be incorporated into and jointly analyzed by a generic framework. Note that these principles can be well combined to create a *hybrid* solution, as done in [21] combining randomness and over-half occupation. However, we omit special discussion of it as the analysis would be easy from each individual principle.

#### A. Randomness

In order to design a schedule of active states ensuring neighbor discovery, a simple once-for-all method is to let nodes be active/idle with a given probability, which is the main idea of randomness. The Birthday protocols [17] are among the most well-known methods adopting randomness. The inspiration roots in the *Birthday Paradox* [32] in which we compute the probability that at least a pair out of a set of  $n$  randomly chosen people have the same birthday. While apparently the

probability reaches 100% when  $n$  reaches 367 (considering 29 Feb.) by the pigeonhole principle, it is surprising that 99% can be reached with just 57 people and 50% with 23 people. This idea is applied to the scenario of channel access. Over a period of  $n$  slots, two nodes independently and randomly select  $k$  slots, one for transmitting beacons and the other for listening. They both remain idle in the remaining  $n - k$  slots. Under this scenario, the probability that the listening node can hear the transmitting one is given by

$$P(n, k) = 1 - \frac{\binom{n-k}{k}}{\binom{n}{k}}. \quad (1)$$

The probability approaches 1 when the ratio  $k/n$  is relatively small. For example,  $P(1000, 70) \approx 0.995$ . That is, a very high probability (99.5%) of discovery can be achieved in the presence of a low duty cycle (7%).

With such observation, the Birthday protocols work as follows. At the start of each slot, a node chooses with probability  $p_l$ ,  $p_t$ , and  $p_s$  whether the state for that slot is to be listening, transmitting, or sleeping (idle). For the purpose of saving energy during the deployment of nodes and maximizing the probability of discovery, the authors of [17] refined the method by arranging nodes to operate in different modes with different probability settings.

Given the probabilistic nature, it is natural to look at its average latency. For simplicity, we consider the case where nodes transmit and listen in an active slot with probability of  $p$ . Then the probability that their first discovery occurs in the  $n$ -th slot is  $(1 - p^2)^{n-1}p^2$ . Consequently, the expectation of discovery latency is given by

$$E(l) = p^2 \sum_{n=1}^{\infty} n(1 - p^2)^{n-1}. \quad (2)$$

With the known result

$$\sum_{n=1}^{\infty} nx^{n-1} = \frac{1}{(1-x)^2}, \quad (3)$$

the average latency of the Birthday protocol is  $1/p^2$ . That is, a duty cycle of 5% yields an average latency of 400 slots.

Analysis of the Birthday protocols can also be found in [33], [34]. In [33], the authors analyzed the energy cost of the general birthday protocol and a probabilistic round robin birthday protocol in which

$$\begin{aligned} p_t &\leftarrow \frac{1}{N}, \\ p_l &\leftarrow 1 - \frac{1}{N}, \\ p_s &\leftarrow 0, \end{aligned} \quad (4)$$

where  $N$  is the number of nodes. The authors in [34] derived the expected time equal to  $Ne(\ln N + c)$  for some constant  $c$ , when all neighbors are discovered using the birthday protocols.

Overall, the Birthday protocols can achieve fast neighbor discovery in the average case. This is because its probabilistic nature makes discovery independent of time displacement. Besides, the median latency would be shorter than the average latency due to the unbounded worst-case latency. They also

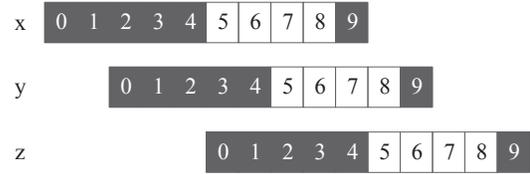


Fig. 5. Naïve implementation of over-half occupation.

support asymmetric operations by selecting different wake-up probabilities. However, the probabilistic nature also leads to aperiodic and unpredictable discovery latency, and thus long tails in proportion of discoveries. For applications that put a hard constraint on maximum latency, the Birthday protocols might not be an appropriate solution. Moreover, they usually aim at *static* ad hoc wireless networks, which is not the case for MANETs and WSNs such as flock monitoring and asset tracking.

### B. Over-half Occupation

The most straightforward way to ensure neighbor discovery deterministically is to be active at least half of the slots in each period. For example, for each period containing  $n$  slots, a node is active in the first  $\lceil (n+1)/2 \rceil$  slots and idle in the remaining. We refer to such a periodic scheduling as *over-half occupation*, or the “51%” solution named in [24].

It is easily observed that two nodes operating in this mode can discover each other regardless of time displacement, as illustrated in Fig. 5 for the case of  $n = 9$ . In addition, the discovery is assured within a period, optimal for any periodic methods. The cost to achieve such optimum is the comparatively high energy consumption; the duty cycle is over 50%, undesirable for low-power operations. We call this a naïve implementation of over-half occupation.

One way to deal with the excessive energy consumption in the naïve method is to spread active slots across multiple cycles. Specifically, we consider the active slots except for the first active one, of which the number is denoted by  $k = \lceil (n+1)/2 \rceil - 1$ . For any divisor  $r$  of  $k$  other than 1, we partition the  $k$  slots into  $r$  sequences, namely  $S_1, S_2, \dots, S_r$ , each containing  $k/r$  slots. These  $r$  sequences of active slots are allocated across  $r$  cycles, which compose a larger period. The first slot of each cycle remains active. That is, in the first cycle, the first and the following  $k/r$  slots ( $S_1$ ) are scheduled as active slots; in the second cycle, the first and the following  $(k/r + 1)$ -th to  $(2k/r)$ -th slots ( $S_2$ ) are active slots, and so forth.

Fig. 6 shows a example of  $n = 9, r = 2$ . Originally 5 consecutive active slots are at the beginning of each cycle as in the naïve over-half method. The four active slots are separated into two sequences, each of length 2 slots. These sequences of active slots are spread across two cycles, reducing the duty cycle from over 50% to 33%. Adjusting the value of  $n$  and  $r$  could further reduce the duty cycle.

SearchLight [21] adopts such translation in essence. The parameter  $r$  is set to  $k$  such that there is two active slots in each cycle, one static slot (namely the *anchor* slot) at the start and one “moving” slot (the *probe* slot) searching for

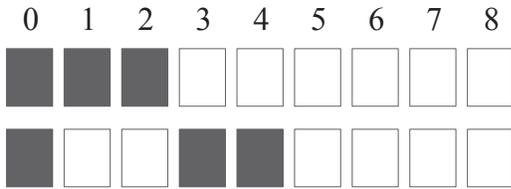


Fig. 6. Spread of active slots into multiple rounds.

the anchor slots of the other node. The authors proved that probing of a full period containing  $r$  cycles guarantees two overlaps whenever the slot boundaries of two nodes are not aligned. This redundancy is eliminated by *striped probing* in which the probe slot moves every other slot and the length of active slots is expanded by a small factor to cope with perfect alignment scenarios. In order to increase the chance that two probe slots meet, the authors exploited *randomized probing* by adding randomness to the patterns of probe slots.

The spread of active slots over multiple cycles reduces duty cycle effectively, but at the cost of extended discovery time since the period length gets expanded. It is easy to show that the worst-case discovery latency after spreading is  $r$  times more than the naive method. This result also demonstrates the trade-off between duty cycle and discovery latency.

### C. Rotation-resistant Intersection

Alternative to the over-half occupation idea, another type of deterministic neighbor discovery is achieved by exploring *combinatorial characteristics* of slots. After all, we would like to figure out under energy constraints, how we can schedule active slots of two nodes such that they intersect regardless of rotations (*i.e.*, time displacement). We assume the number of time slots in each period is  $n$ , and the slot index starts at 0. Let  $\pi$  denote a set of indices of active slots in a period. A  $k$ -rotation of  $\pi$ , denoted by  $\pi^k$ , is obtained by adding  $k$  modulo  $n$  to each element in  $\pi$ . When  $\pi$  and  $\pi^k$  have common elements, the active slots overlap with a time displacement of  $k$  slots. For example when  $n = 9$ ,  $\pi = \{1, 4, 8\}$ , and  $k = 4$ ,  $\pi^k = \{5, 8, 3\}$  and active slots overlap at slot 8 with a time displacement of 4 slots. Therefore, the problem of neighbor discovery can be formulated as follows: Given  $n$ , find a  $\pi$  of minimum size such that  $\pi \cap \pi^k \neq \emptyset$  for all  $0 \leq k < n$ .

Fig. 7 presents an example satisfying this property in which  $n = 9$  and  $\pi = \{0, 1, 3, 6\}$ . For three nodes  $x$ ,  $y$ , and  $z$ , they all adopt the same  $\pi$  for active slot scheduling. The time counter of  $x$  is used as the reference. Node  $x$  discovers  $y$  in slot 3, and  $y$  discovers  $z$  in slot 5, with both time displacement of 2 slots ( $k = 2$ ). Node  $x$  and  $z$  discover each other at slot 10 with initial time displacement of 4 slots ( $k = 4$ ). It is easy to verify that mutual discovery is guaranteed within a period for other time displacement possibilities.

As the intersection is guaranteed irrespective of any rotation, this property is named as *rotation-resistant intersection*. The work of [18] is among the earliest adopting this idea. It follows the notion of *quorum* [35] and thus is usually referred to as the Quorum protocol. The sequence of time slots is separated into groups containing  $m^2$  consecutive slots. In each group,

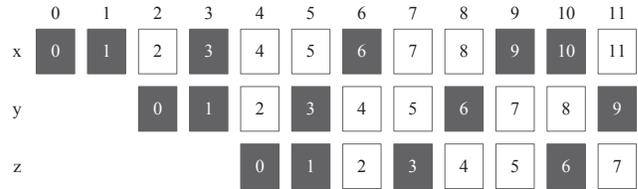


Fig. 7. Rotation-resistant intersection.

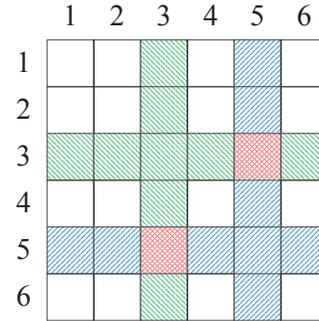


Fig. 8. A Quorum illustration.

we arrange the  $m^2$  slots as an  $m \times m$  matrix in a row-major manner. A node arbitrarily picks one column and one row of entries as active slots (called quorum intervals), while the remaining  $m^2 - 2m + 1$  slots are idle slots. Given two nodes that are perfectly time-aligned, we can see that their quorum intervals always have at least two intersecting slots. This is because a column and a row in a matrix always have an intersection. The case becomes somewhat complicated when the nodes are not time-aligned. Still, the authors proved that the Quorum protocol guarantees at least two overlapped active slots in every  $m^2$  slots, no matter what the time displacement (equivalent to rotation) is.

A Quorum illustration is given in Fig. 8. In the  $6 \times 6$  matrix  $M$ , a node selects the third row and column and the other does the fifth ones. Their selection has common slots in  $M(3, 5)$  and  $M(5, 3)$ , where  $M(i, j)$  stands for the entry in the  $i$ -th row and  $j$ -th column. They discover each other in these two overlapping active slots.

Analysis on the number of active slots needed for rotation-resistant intersection are developed in [36], [35], [37]. A lower bound of  $\Omega(\sqrt{n})$  slots are required for discovery, and  $O(n)$  active slots suffice to guarantee discovery as similar to the match-making problem [35]. Zheng et al. [36] applied optimal block designs using difference sets that are obtained by the Multiplier Theorem [38]. The work of [37] improved on the quorum construction with low power and investigated randomized schedules with high probability of rotation-resistant intersection.

While the Quorum protocol provides a reasonable bound on the worst-case latency, it may perform much worse than the birthday protocols in the average case due to the redundant intersection. Besides, as  $n$  is a global parameter, it only supports symmetric operations, *i.e.*, all nodes must operate at the same duty cycle. Lai et al. [39] improved the Quorum protocol so as to handle asymmetric cases in which two duty cycles are allowed. We argue that this may still be too restrictive. The

work of [36] proved that designing an optimal schedule for the asymmetric case following the block design is reduced to the vertex cover problem, which is NP-complete. Therefore, they are fundamentally limited to symmetric networks.

#### D. Coprime Cycles

In order to overcome the limitation of supporting only symmetric operations while still providing a strict bound on the worst-case latency, researchers have made use of coprime cycles thanks to the Chinese Remainder Theorem [40]. Suppose  $n_1, n_2, \dots, n_k$  are positive integers which are pairwise coprime. The Chinese Remainder Theorem states that for any given sequence of integers  $a_1, a_2, \dots, a_k$ , there exists an integer  $x$  such that the following equations of simultaneous congruence hold:

$$\begin{aligned} x &\equiv a_1 \pmod{n_1}, \\ x &\equiv a_2 \pmod{n_2}, \\ &\vdots \\ x &\equiv a_k \pmod{n_k}. \end{aligned} \quad (5)$$

Furthermore, all solutions  $x$  are congruent modulo the product  $N = n_1 n_2 \cdots n_k$ .

Based on what we learn from the Chinese Remainder Theorem, the idea of coprime cycles is as follows. We can design a periodic scheduling in which a node is active only in the first slot of each period, and the period lengths are pairwise coprime. Then neighbor discovery of two nodes is guaranteed within the product of their respective period lengths. Put it formally, let  $p$  and  $q$  denote the number of slots in each period of two nodes. They are coprime such that their greatest common divisor is 1. Then the maximum discovery latency is bounded by the product  $pq$ .

We make several observations of the applicability of the coprime idea for neighbor discovery in WSNs. First, the prerequisite of the coprime idea is to make sure for any two nodes, their period lengths are pairwise coprime. Without this condition, it is possible that two neighboring nodes would never discover each other. A centralized method to assure this requirement is that a node computes a set of pairwise coprime numbers and allocates them to other nodes. It might be a solution at first glance, but not applicable in real-world applications due to its deficiency in computation, scalability, and adaptivity. A distributed method to relieve the pain is to assign each node a prime number. The coprime property holds in *most* cases as two different prime numbers are coprime in evidence. However, the exceptional scenario that two nodes pick the same prime number (*i.e.*, symmetric duty cycles) should not be overlooked. For example in Fig. 9, node  $x$  and  $y$  both select 3 as their cycle lengths, and they have an initial displacement of 1 slot. Then they would never discover each other.

Second, the selection of period length corresponds to its duty cycle. Specifically, the duty cycle is equal to the inverse of the period length as it is active once per period. This contributes to the support of asymmetric operations: two nodes having different period lengths work at different duty cycles. On the other hand, a node can select its period length



Fig. 9. Selection of the same prime.

according to its desired duty cycle. For example, 101 may be a choice for the period length if the desired duty cycle is 1%.

Third, the coprime idea can be a good start for periodic scheduling. To further decrease discovery latency, we can extend the only one active slot in each period by stretching it to consecutive active slots, or add multiple separated active slots. Note that again the trade-off between duty cycle and latency plays a role in such adjustment: a lower latency is achieved at the cost of higher duty cycle.

Several work adopts the coprime idea for neighbor discovery [19], [20], [41], with differences in dealing with the case when the round lengths of neighbors are not coprime. Herman et al. [41] proposed a repeat process of random prime selection. A node randomly selects a number from a set of two coprime numbers  $\{z, z+1\}$ , uses it for  $k \cdot z$  rounds, and repeats the process. The maximum discovery latency is thus  $O(z^2)$ . In Disco [19], each node selects a pair of prime numbers  $\{p, q\}$ . The nodes then wake up at multiples of the individual primes. Clearly, their duty cycles are equal to the sum of the reciprocals of respective primes, *i.e.*,  $1/p + 1/q$ . As  $p \neq q$ , discovery is guaranteed even if neighbors pick the same pair of numbers. For example,  $x$  and  $y$  may select the same pair of primes, say 3 and 5. As 3 and 5 are coprime in evidence, they will discover each other within 15 slots. The authors also suggest the use of *balanced* primes (the difference between the pair of primes is minimal) for symmetric discovery, and *unbalanced* primes (where the difference is maximal) for asymmetric discovery. Besides, a refinement using a triple of prime numbers is also discussed in [19]. U-Connect [20] further relaxed the constraint from a pair of primes to a single prime. Besides waking up 1 slot every  $p$  slots, the nodes also wakes up in the first  $\lceil (p+1)/2 \rceil$  slots every  $p^2$  slots (combining with over-half occupation).

Compared with the Quorum protocol, all the coprime schemes support asymmetric operations due to the various selection of prime numbers. In [20], the authors adopted the theoretically optimal neighbor discovery schedules discussed in [36] and analyzed the Quorum protocol, Disco and U-Connect using the power-latency (PL) product as the metric. Their theoretical analysis shows that the Quorum protocol and Disco are both 2-approximation algorithms of the optimal, while U-Connect achieves a 1.5-approximation of the optimal.

#### E. Integrative Analysis of Deterministic Protocols

Up to now, we have covered four design principles and corresponding representative protocols. Despite the difference in underlying principles, we observe some similarities among the deterministic protocols (*i.e.*, Quorum, Disco, U-Connect, and Searchlight). For example, some of their active slots, say those in Disco and anchor slots in Searchlight, exhibit a repetitive pattern. Interestingly, we realize that under symmetric duty

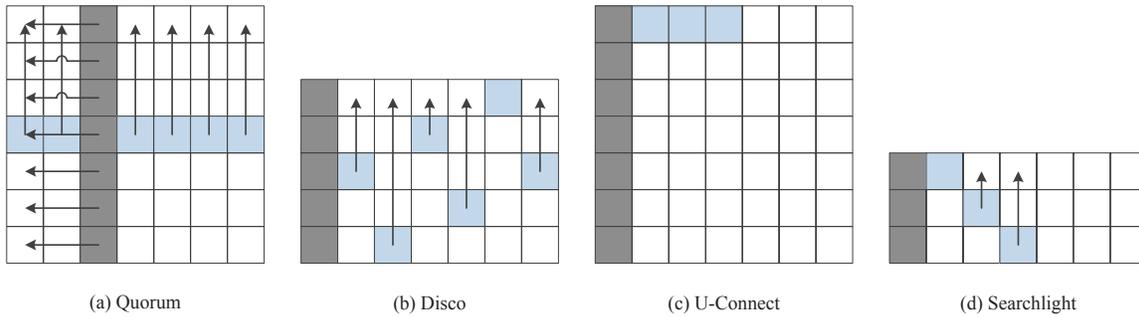


Fig. 10. Deterministic protocols incorporated into a generic framework.

cycles, these deterministic protocols can be incorporated into a generic framework. Specifically, when a period of slots are organized as a matrix, these deterministic protocols can be transformed to the case where the slots in the first column and in the first half of the first row are active slots. This is illustrated in Fig. 10. In the following, we show how the transformation is done and its implication.

1) *Quorum*: In Quorum, a node picks one column and one row of entries as active slots in an  $m \times m$  array of consecutive slots. We observe that it makes no difference if each node picks the first column and the first row. Equivalently, it incurs no loss in terms of discovery latency if the selected row/column are shifted to the first row/column (see Fig. 10(a)). As a result, Quorum becomes a redundant variant of the framework with additional active slots in the second half of the first row.

2) *Disco*: A node using Disco for neighbor discovery wakes up at multiples of individual prime  $p$  or  $q$  ( $p \neq q$ ). Consider a cycle of  $pq$  slots that are organized into a  $q \times p$  matrix. Clearly, the first column are all active slots. Since  $p$  and  $q$  are coprime, it is known that no two of the integers  $q, 2q, \dots, (p-1)q$  are congruent modulo  $p$  [42]. Therefore, there is exactly one active slot in each remaining column. If those remaining active slots are shifted to the first row (see Fig. 10(b)), we can prove using a similar proof technique as in [21] that the worst-case latency under symmetric duty cycles remains unchanged (a period). Therefore, Disco amounts to a redundant variant of the framework.

3) *U-Connect*: It can be easily observed (see Fig. 10(c)) that U-Connect is a direct derivant from the framework with a  $p \times p$  square matrix, where  $p$  is prime.

4) *Searchlight*: In Searchlight, a nodes wakes up at anchor slots that come every  $t$  slots, and at probe slots that traverse from position 1 to  $\lfloor t/2 \rfloor$  across  $\lfloor t/2 \rfloor$  cycles. It is easy to show that the worst-case latency stays the same if probe slots are shifted to the first row, using a similar technique as before (see Fig. 10(d)). Essentially, Searchlight is also a particular variant of the framework.

To summarize, these four representative deterministic protocols are all interconnected through the generic framework. Though this framework applies only under symmetric duty cycles, it has the potential to be used to explore other candidate protocols with different combinations of parameters (*e.g.*, number of rows and columns) and to further determine the optimal configuration. At present this issue is still under investigation.

## IV. COMPARISON AND EVALUATION

In order to strengthen the understanding of the aforementioned NDPs, we present a qualitative comparison and a quantitative evaluation study in this section. For the comparison, we employ multiple criteria of interest to make it comprehensive. For the evaluation, we focus on their energy efficiency and measure the cumulative distribution of discovery at given duty cycles, either symmetric or asymmetric.

### A. Comparison Study

We include the Birthday protocol, Searchlight, Quorum, Disco, and U-Connect for the comparison study. The comparison is based on the following criteria:

- **Probabilistic or deterministic (P/D)**. All NDPs can be dichotomized into probabilistic and deterministic approaches. Basically deterministic approaches are able to give a predictable discovery latency, while the discovery using probabilistic approaches can only be estimated with probabilities. For applications that require a strict upper bound on discovery latency, deterministic approaches are preferable.
- **Supporting of asymmetric operations (Asymm.)**. For many applications, nodes need to adjust their duty cycles according to their tasks, energy budget, connectivity, *etc.* Therefore, a neighbor discovery protocol that supports asymmetric operations are highly desired.
- **Average latency (Avg lat.)**. At a given duty cycle (say 5%), latency is the key to evaluate energy efficiency of NDPs. During the initial deployment, nodes need to discover their neighbors as soon as possible for data exchange. As discovery latency is a random variable, its mean value acts as a good indicator.
- **Maximum latency (Max lat.)**. While a low average latency is desirable, the maximum latency at a duty cycle of  $d$  may be of greater interest to many applications that need to ensure discovery within a given amount of time. Readers can refer to [21] for detailed analysis.

The comparison is conducted under the same duty cycle  $d$ . The result is summarized in Table I. We group the protocols according to their design principles, *i.e.*, randomness, over-half occupation (OHO), rotation-resistant intersection (RRI), and coprime cycles. To reflect the underlying design principle of Searchlight, we leaves its optimization techniques out of account and consider only its original implementation, as those techniques can also be well applied to other protocols.

TABLE I  
COMPARISON STUDY OF NEIGHBOR DISCOVERY PROTOCOLS

Protocols		P/D	Asymm.	Avg lat.	Max lat.
Randomness	Birthday	P	Yes	Low	N/A
OHO	SearchLight	D	Yes	Low	$2/d^2$
RRI	Quorum	D	No	High	$4/d^2$
Coprime	Disco	D	Yes	High	$4/d^2$
	U-Connect	D	Yes	Medium	$9/4d^2$

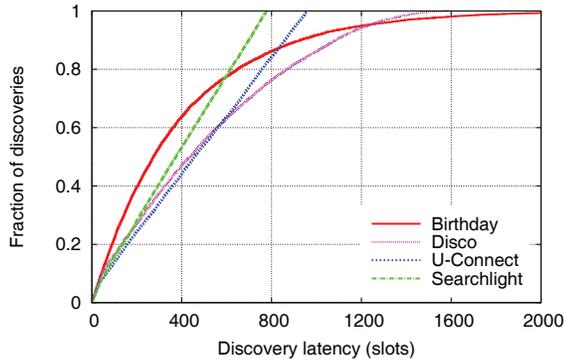


Fig. 11. CDF of discovery latency at 5% duty cycle.

## B. Evaluation

In order to measure energy efficiency, both duty cycle and discovery latency should be included for analysis. However, due to the trade-off between them, we consider only discovery latency of different protocols at given duty cycles. Specifically, we are interested in the CDF (Cumulative Distribution Function) of discovery latency under symmetric and asymmetric duty cycles. Note that Quorum does not support asymmetric duty cycles, and is thus not included. Disco uses two different sets of parameters for symmetric and asymmetric duty cycles, and we follow this convention in the evaluation. All the protocols are evaluated through a state-based simulation: nodes To reflect the overall distribution of discovery latency, we ran them 10000 times and collected discovery latency with random initial clock readings. In order to be independent of specific hardware platforms, we use the number of slots to measure latency.

1) *Discovery Latency under Symmetric Duty Cycles:* We evaluate first the performance of different protocols under symmetric duty cycles. The evaluation was conducted with a typical duty cycle of 5%. Fig. 11 shows the CDF of discovery latency. The Birthday protocol achieved the fastest discovery most of the time, but suffered from the unpredictable large latency as shown by the long tail. Disco was inferior to the Birthday protocol for over 95% of the time but it gave a latency bound of 1600 slots. U-Connect further shortened the worst-case latency to 960 slots. Finally, Searchlight improved on U-Connect and guaranteed a worst-case latency of 800 slots. In short, we infer that Searchlight is most preferable in terms of either maximum or average latency under symmetric duty cycles.

2) *Discovery Latency under Asymmetric Duty Cycles:* We now consider the performance of the protocols under asymmetric duty cycles. For implementation convenience, we adopted two duty cycles, 5% and 1%. Fig. 12 shows the

TABLE II  
PARAMETER SETTINGS

Protocols	5% (Symm.)	(5%, 1%) (Asymm.)
Birthday	0.05	0.05, 0.01
Disco	{37, 43}	{23, 157}, {101, 9973}
U-Connect	31	31, 151
Searchlight	40	40, 200

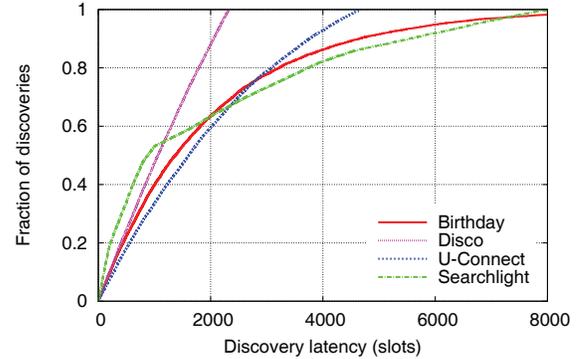


Fig. 12. CDF of discovery latency at the duty cycles of 5% and 1%.

result. Similarly, the Birthday protocol suffered from the unpredictable large latency. Different from the good performance under symmetric duty cycles, Searchlight incurred large latency comparable to the Birthday protocol. The performance of U-Connect was consistent in both cases, limiting the maximum latency to 4680 slots. Finally, Disco further dominated U-Connect, giving the maximum latency of 2322 slots. The message here is that Disco exhibits the best performance under asymmetric duty cycles.

3) *Average Latency for Both Cases:* Though the above illustrations show the maximum latency of the protocols, how they perform in terms of average latency is not quite clear. Therefore, we calculated the average latency of each protocol for both cases, and presented the results in Fig. 13. Note that despite the unbounded maximum latency, the Birthday protocol achieved fast discovery under the symmetric duty cycle on average, even 20% faster than Disco and 13% than U-Connect. While under the asymmetric duty cycle, Disco outperformed the others by around 40%. Combining this with the prior results, we find that there is no all-round approach towards energy-efficient neighbor discovery for both cases.

## V. FUTURE DIRECTIONS

Although a number of approaches to neighbor discovery have been proposed, several issues still remain open for future research. We point out and discuss some possible topics in this section.

- **Mining of wake-up patterns.** Traditional NDPs, including those discussed in this survey, concentrate on scheduling of active slots without referring to the schedules of other nodes. We argue that this might not be efficient enough for mutual discovery. In fact, nodes can add their scheduling information to beacon messages such that their neighbors are able to learn the wake-up patterns, leading to a faster discovery in future or a greater energy

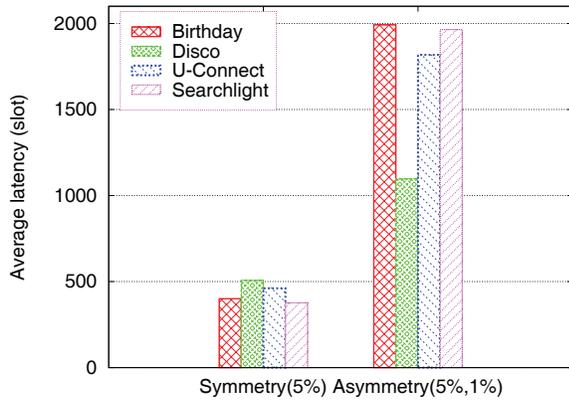


Fig. 13. Average latency for the symmetric and asymmetric duty cycles.

conservation. Take the coprime schemes for example. A node can inform its neighbors of the selection of prime number(s). Its neighbors then learn its wake-up patterns (*i.e.*, when it will be active). After the initial discovery, they can adjust their own scheduling for future discovery based on application requirements. A big challenge of adopting this idea is how to ensure discovery efficiently as the scheduling is adjusted, different from the one they broadcasted before. For example, if a node simply add active slots for faster discovery, its duty cycle would be high if it has a great number of neighbors. Acc [43] is a recent work leveraging temporal diversity to decide which slots is added as active slots. Further improvement and analysis is left open.

- **Collaborative neighbor discovery.** Besides mining wake-up patterns from beacon messages, another possibility to boost neighbor discovery is to leverage cooperation among nodes. For example, in addition to direct discovery between neighbors, we may also consider *indirect* neighbor discovery where two nodes discover each other via a third node. The intuition behind this is that two nodes with the same neighbor(s) are neighbors with a high probability. Their common neighbors may act as coordinators to accelerate the indirect discovery. However, as neighborhood relation is not transitive, such indirect discovery may not be valid. Acc [43] assessed spatial similarity of two nodes using the ratio between the number of common known neighbors and one's own neighbors. Further investigation of indirect discovery and other collaborative ways are needed.
- **Neighbor discovery in multihop WSNs.** We hold an assumption of one-hop neighbors in this paper. However, it is possible to define and employ two-hop neighbors, where two nodes are not within the communication range of each other but have at least one common one-hop neighbor. Such neighbors have the potential to benefit a number of network services, such as routing [44], connectivity [45], and localization [46], [47]. At present, no work has focused on multihop neighbor discovery. Research challenges, such as communications between two-hop neighbors and load balancing among common one-hop neighbors, remain open for study. We expect

novel research work to fill this gap.

- **Flock discovery.** Most of existing neighbor discovery protocols focus on discovery of two nodes and apply the two-node mechanism to the whole network. However in many applications, such as habitat monitoring, asset tracking, and search and rescue, discovery of a group of nodes attracts greater interest. The former case with individual discovery can be regarded as micro-discovery, while the latter with flock discovery as macro-discovery. Due to its flock nature, existing protocols may not be efficient. Coordination among flock nodes might be adopted as a necessary mechanism.
- **Neighbor discovery in mobile computing.** Compared with applications of WSNs, applications of mobile computing are more related to everyday's life and become extremely popular in recent years. With the rapid development of smart devices such as mobile phones and tablets, these devices not only act as the tools they are designed to be, but also powerful mobile stations for sensing and communicating where neighbor discovery is the first step. But different from the strict energy constraint in WSNs, mobile computing applications put a loose requirement on energy consumption and prefer faster discovery. A typical example is human traffic statistics. The authors of eDiscovery [48] propose an efficient device discovery protocol as the first step to bootstrapping opportunistic communication for smartphones. We believe a lot of research work will be conducted in this direction.

## VI. CONCLUSION

In this survey, we collect the ideas prevalent in the research literature on neighbor discovery in both mobile ad hoc networks and wireless sensor networks. In general, neighbor discovery protocols can be roughly classified based on their underlying principles: randomness, over-half occupation, rotation-resistant intersection, and coprime cycles. We present and compare several representative protocols under these four principles, and evaluate them under symmetric and asymmetric duty cycles. We further point out several future directions in this field. As a fundamental process in both communication and power management, neighbor discovery will remain hot in research community and further research will deepen our understanding of discovery mechanisms and ad hoc communications.

## ACKNOWLEDGMENT

This work is supported in part by the NSFC Major Program 61190110, NSFC under grant 61171067 and 61133016, National Basic Research Program of China (973) under grant No. 2012CB316200, and the NSFC Distinguished Young Scholars Program under Grant 61125202.

## REFERENCES

- [1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *Mobile ad hoc networking*. John Wiley & Sons, 2004.
- [2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, 2002.

- [3] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, 2008.
- [4] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "Trafficview: traffic data dissemination using car-to-car communication," *ACM SIGMOBILE Mobile Computing Commun. Review*, vol. 8, no. 3, pp. 6–19, 2004.
- [5] S. Dornbush and A. Joshi, "Streetsmart traffic: discovering and disseminating automobile congestion using VANET's," in *Proc. IEEE Veh. Technol. Conf.*, 2007, pp. 11–15.
- [6] Q. Huang and R. Miller, "The design of reliable protocols for wireless traffic signal systems," *Technical Report WUCS-02-45*, 2003.
- [7] N. Nafi and J. Khan, "A VANET based intelligent road traffic signalling system," in *Proc. IEEE Telecommunication Netw. and Applications Conf.*, 2012, pp. 1–6.
- [8] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. ACM WSNA*, 2002, pp. 88–97.
- [9] Y. Liu, X. Mao, Y. He, K. Liu, W. Gong, and J. Wang, "CitySee: not only a wireless sensor network," *IEEE Netw.*, vol. 27, no. 5, pp. 42–47, 2013.
- [10] M. Li, Z. Yang, and Y. Liu, "Sea depth measurement with restricted floating sensors," *ACM Trans. Embedded Computing Systems*, vol. 13, no. 1, pp. 1–21, 2013.
- [11] J.-H. Huang, S. Amjad, and S. Mishra, "Cenwits: a sensor-based loosely coupled search and rescue system using witnesses," in *Proc. ACM SenSys*, 2005, pp. 180–191.
- [12] B. Kusý, I. Amundson, J. Sallai, P. Völgyesi, A. Lédeczi, and X. Koutsoukos, "RF doppler shift-based mobile sensor tracking and navigation," *ACM Trans. on Sensor Netw.*, vol. 7, no. 1, pp. 1:1–1:32, 2010.
- [13] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application," in *Proc. ACM SenSys*, 2008, pp. 337–350.
- [14] J. He, S. Ji, Y. Pan, and Y. Li, "Reliable and energy efficient target coverage for wireless sensor networks," *Tsinghua Science and Technol.*, vol. 16, no. 5, pp. 464–474, 2011.
- [15] S. Yau and F. Karim, "An energy-efficient object discovery protocol for context-sensitive middleware for ubiquitous computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 14, no. 11, pp. 1074–1085, 2003.
- [16] P. Dutta, D. Culler, and S. Shenker, "Procrastination might lead to a longer and more useful life," in *Proc. ACM HotNets-VI*, 2007.
- [17] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. ACM MobiHoc*, 2001, pp. 137–145.
- [18] 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.
- [19] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. ACM SenSys*, 2008, pp. 71–84.
- [20] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-Connect: a low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proc. ACM/IEEE IPSN*, 2010, pp. 350–361.
- [21] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: won't you be my neighbor?" in *Proc. ACM MobiCom*, 2012, pp. 185–196.
- [22] K. Sreekanth and S. Sarfaraj, "A survey on neighbor discovery in asynchronous wireless sensor network," *Int. J. of Advanced Research in Computer Science and Software Engineering*, vol. 2, no. 9, pp. 265–269, 2012.
- [23] S. E. Roslin, C. Gomathy, and P. Bhuvaneshwari, "A survey on neighborhood dependant topology control in wireless sensor networks," *Int. J. of Computer Science & Commun.*, vol. 1, no. 1, pp. 185–188, 2010.
- [24] V. Galluzzi and T. Herman, "Survey: discovery in wireless sensor networks," *Int. J. of Distrib. Sensor Netw.*, 2012.
- [25] R. Cohen and B. Kapchits, "Continuous neighbor discovery in asynchronous sensor networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 1, pp. 69–79, 2011.
- [26] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.
- [27] T. Liu, C. M. Sadler, P. Zhang, and M. Martonosi, "Implementing software on resource-constrained mobile sensors: experiences with Impala and ZebraNet," in *Proc. ACM MobiSys*, 2004, pp. 256–269.
- [28] H. Jun, M. H. Ammar, M. D. Corner, and E. W. Zegura, "Hierarchical power management in disruption tolerant networks using traffic-aware optimization," *Elsevier Computer Commun.*, vol. 32, no. 16, pp. 1710–1723, 2009.
- [29] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM SenSys*, 2004, pp. 95–107.
- [30] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM*, 2002, pp. 1567 – 1576.
- [31] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves, "Energy-efficient collision-free medium access control for wireless sensor networks," in *Proc. ACM SenSys*, 2003, pp. 181–192.
- [32] Wikipedia, "Birthday paradox," [http://en.wikipedia.org/wiki/Birthday\\_problem](http://en.wikipedia.org/wiki/Birthday_problem).
- [33] S. Fang, S. Berber, and A. Swain, "Analysis of neighbor discovery protocols for energy distribution estimations in wireless sensor networks," in *Proc. IEEE ICC*, 2008, pp. 4386–4390.
- [34] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili, "Neighbor discovery in wireless networks and the coupon collector's problem," in *Proc. ACM MobiCom*, 2009, pp. 181–192.
- [35] S. Mullender and P. Vitányi, "Distributed match-making," *Algorithmica*, vol. 3, no. 1, pp. 367–391, 1988.
- [36] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proc. ACM MobiHoc*, 2003, pp. 35–45.
- [37] M. Bradonjić, E. Kohler, and R. Ostrovsky, "Near-optimal radio use for wireless network synchronization," *Theoretical Computer Science*, vol. 453, pp. 14–28, 2012.
- [38] I. Anderson, *Combinatorial designs and tournaments*. Oxford University Press, 1998.
- [39] S. Lai, B. Ravindran, and H. Cho, "Heterogenous quorum-based wakeup scheduling in wireless sensor networks," *IEEE Trans. Comput.*, vol. 59, no. 11, pp. 1562–1575, 2010.
- [40] I. Niven, H. Zuckerman, and H. Montgomery, *An introduction to the theory of numbers*. John Wiley & Sons, 2008.
- [41] T. Herman, S. Pemmaraju, L. Pilard, and M. Mjelde, "Temporal partition in sensor networks," in *Proc. Springer SSS*, 2007, pp. 325–339.
- [42] K. H. Rosen, *Discrete mathematics and its applications*. McGraw-Hill, 2012.
- [43] D. Zhang, T. He, Y. Liu, Y. Gu, F. Ye, R. K. Ganti, and H. Lei, "Acc: generic on-demand accelerations for neighbor discovery in mobile applications," in *Proceedings of ACM SenSys*, 2012, pp. 169–182.
- [44] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," *Wireless Commun.*, vol. 11, no. 6, pp. 6–28, 2004.
- [45] A. Ghosh and S. K. Das, "Coverage and connectivity issues in wireless sensor networks: a survey," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303 – 334, 2008.
- [46] Z. Yang and Y. Liu, "Quality of trilateration: confidence-based iterative localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 631–640, 2010.
- [47] Y. Liu, Z. Yang, X. Wang, and L. Jian, "Location, localization, and localizability," *J. of Computer Science and Technol.*, vol. 25, no. 2, pp. 274–297, 2010.
- [48] B. Han and A. Srinivasan, "eDiscovery: energy efficient device discovery for mobile opportunistic communications," in *Proc. IEEE ICNP*, 2012, pp. 1–10.



**Wei Sun** received the B.E. degree in computer science from University of Science and Technology of China, Hefei, China, in 2011. He is currently a Ph.D. candidate in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include wireless ad hoc/sensor networks and mobile computing.



**Zheng Yang** received the B.E. degree in computer science from Tsinghua University, Beijing, China, in 2006 and the Ph.D. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 2010. He is currently an Assistant Professor with the School of Software, TNList, Tsinghua University. His research interests include wireless ad hoc/sensor networks and mobile computing.



**Yunhao Liu** received the B.S. degree in automation from Tsinghua University, China, in 1995, the M.S. and Ph.D. degrees in computer science and engineering from Michigan State University, in 2003 and 2004, respectively. He is now an EMC chair professor at Tsinghua University. His research interests include wireless sensor networks, peer-to-peer computing, and pervasive computing.



**Xinglin Zhang** received the B.E. degree in the School of Software, Sun Yat-Sen University, Guangdong, China, in 2010. He is now a Ph.D. student in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology. His research interests include wireless ad-hoc/sensor networks, mobile computing, and crowdsourcing.