Improving Neighbor Discovery by Operating at the Quantum Scale

Shuaizhao Jin*, Xiangyun Meng†, Daniel Lin-Kit Wong◇, Zixiao Wang‡, Ben Leong‡, Yabo Dong*, Dongming Lu*
*Zhejiang University *{jinszhao, dongyb, ldm}@zju.edu.cn
†University of Washington †xiangyun@cs.washington.edu
◇Carnegie Mellon University ◇wonglkd@cmu.edu
‡National University of Singapore ‡{zixiao, benleong}@comp.nus.edu.sg

***Abstract*—Duty-cycling is generally adopted in existing sensor networks to reduce power consumption and these networks depend on neighbor discovery protocols to ensure that nodes wake up and discover each other. For different neighbor discovery protocols, the discovery latency is determined by two factors: the wake-sleep pattern and slot size. To the best of our knowledge, previous works on neighbor discovery have thus far been focused on improving the wake-sleep pattern. In this paper, we investigate the extent to which we can improve discovery latency by reducing the slot size. We found that by reducing the slot size, i.e., reducing the listening time in active slots, the collisions between beacons and synchronization between nodes become more severe, which can lead to discovery failures that are not predicted by existing theoretical models. We show that we can mitigate these effects by reducing the number of beacons and introducing randomization. We propose a new continuous-listening-based neighbor discovery algorithm called *Spotlight*. Our evaluations with a practical sensor testbed suggest that Spotlight can achieve a 50% reduction in discovery latency over existing state-of-the-art neighbor discovery protocols without increasing power consumption in existing sensor networks.**

I. Introduction

Power is an important consideration for low-power mobile sensing devices, especially for the sensors deployed in scenarios where it is difficult to replace batteries. Duty-cycling [2, 6] is commonly employed to save power, but it then requires a neighbor discovery protocol to ensure that nodes can detect each other. Existing neighbor discovery protocols ensure that nodes will wake up within the same time slot to discover each other using different wake-sleep patterns [12, 17, 4, 10, 11, 3, 18, 15].

The key drawback of duty-cycling compared to "always on" operation is that the discovery latency is significantly higher, so the goal is to develop algorithms that have small discovery latencies and also guarantee discovery. Clearly, the discovery latency is determined by two factors: the wake-sleep pattern and slot size. But, to the best of our knowledge, previous works on neighbor discovery have thus far been focused on the former, i.e., developing novel wake-sleep patterns to reduce discovery latency. In this paper, we focus on the latter and investigate the fundamental question: *exactly how far can we go in reducing discovery latency simply by reducing the slot size?*

Naively, we would expect the discovery latency to decrease proportionally to the decrease in slot size. However, our measurement study with a practical testbed showed that when the slot size is reduced to the region where the listening time is comparable to the beacon preamble, the neighbor discovery latency increased drastically, which means that we move into an operating regime where "quantum" effects become non-negligible. We refer to this regime, where the listening time of the active slots is comparable to length of the beacon preamble, as the *quantum* regime. Existing worst-case latency bounds and measures such as the power-latency product were not sufficient to predict how algorithms, such as Searchlight [3] or Nihao [14], would perform at small slot sizes. This demonstrated a gap in existing theoretical models and analysis when it comes to analyzing performance in this regime. In particular, we found in our measurement study that: (i) the collisions between beacons are relatively common, leading to discovery failures that are not predicted by existing theoretical models; (ii) the relative clock skew matters, and the synchronization between nodes can lead to worst-case discovery latencies that are many times worse than those predicted by existing theoretical models.

We show that we can significantly reduce discovery failures from collisions and synchronization by (i) reducing the density of beacons and (ii) introducing randomization in the wake-up time. But we can do even better. We discovered in our detailed analysis of discovery failures that Searchlight's approach of having probe slots [3] was harmful because it introduced new failure modes. We hence propose a new continuous listening and periodic beaconing pattern, called *Spotlight*. We describe a methodology for analyzing it and proving correctness using a construct called a *BL (Beacon-Listen) diagram*. We show that we can achieve optimal discovery latency for our new wake-sleep pattern with an $n \times 2n$ configuration.

To the best of our knowledge, our work is the first to systematically investigate how ultra-small slot sizes affect neighbor discovery latency for sensor networks. In addition, we make several contributions to the understanding of sensor performance in the *quantum* regime, detailed as follows:

- We completed a detailed measurement study of a practical duty-cycling sensor device and showed that existing models of sensor operation ignore practical radio phenomena which have non-negligible effects in the *quantum* regime;

- We investigated the reasons for discovery failures in the *quantum* regime, and showed that (i) reducing the number of beacons and (ii) introducing randomization can effectively mitigate these failures, enabling us to operate at an effective slot size equal to one beacon length;
- We proposed a new continuous-listening-based neighbor discovery algorithm called *Spotlight* and showed with a practical sensor testbed that Spotlight outperforms existing state-of-the-art neighbor discovery algorithms.

We show in our evaluations with a practical sensor testbed that Nihao performed surprisingly well in the *quantum* regime, but even so, Spotlight can achieve a 50% reduction in discovery latency over Nihao given the same duty cycle setting. We have also shown in our testbed experiment that it is actually feasible to operate at the slot size of 1 ms, which is equal to the beacon length.

Traditional sensor applications typically adopt slot sizes between 10 ms [4, 19, 14] to 50 ms [15]. As sensor hardware continues to improve, we can expect to be able to operate at significantly smaller slot sizes. By adopting the techniques we have developed, we anticipate a reduction in neighbor discovery latencies by an order of magnitude with no increase in power consumption. Our simple randomization technique, which can prevent discovery failures arising from synchronization in the *quantum* regime, is generally applicable to existing neighbor discovery protocols, and is thus expected to be a general feature of such algorithms in the future.

The rest of the paper is organized as follows: in §II, we present an overview of existing neighbor discovery protocols. In §III, we describe the results of our measurement study, which reveal the challenges of working in the *quantum* regime, and two techniques to mitigate the observed problems. In §IV, we describe Spotlight, our new continuous-listening-based neighbor discovery algorithm. Finally, we present our evaluation results obtained via a practical sensor testbed in §V, and conclude in §VI.

II. Related Work

Most neighbor discovery protocols are *slotted* protocols and divide time into identically-sized discrete slots [3, 4, 18, 15]. During *active* slots, the nodes wake up to transmit and listen for beacons from potential neighbors. The more recent proposals sought to minimize the theoretical *discovery latency*, i.e., the number of elapsed slots needed to guarantee that two nodes' active slots overlap [3, 14]. We have found that they typically ignored two important factors affecting the actual latency, *slot size* and *practical discovery failures* (failures that can occur when active slots overlap.)

Existing protocols can be classified by their adoption of either *probabilistic* or *deterministic* wake-sleep slot schedules to achieve an overlap. Probabilistic protocols such as Birthday [12] can often achieve better average-case latencies, but are unable to provide worst-case bounds. The performance of deterministic protocols can be characterized by their *power-latency product*, $\Lambda = DC \cdot L$ [10], with a lower bound of $\Lambda \approx \sqrt{L}$ for an optimal mutual discovery protocol, where L denotes the worst-case latency in terms of slots [20] and DC denotes the duty cycle.

Disco [4] ($\Lambda = 2\sqrt{L}$) and U-Connect [10] ($\Lambda = 1.5\sqrt{L}$) employ prime-based active slot schedules that guaranteed discovery using the Chinese Remainder Theorem [13]. On the other hand, Quorum ($\Lambda = 2\sqrt{L} - 1$) ensured overlaps by casting each period of n^2 slots as an $n \times n$ square, and designating an arbitrary row and column as active slots [17]. Subsequently, Searchlight [3] ($\Lambda = \sqrt{2L}$) introduced the practice of designating active slots as either stationary *anchor* or moving *probe* slots, with the probe slot iterating over possibilities until it achieved an overlap with another node's anchor slot; this concept was also adopted by BlindDate [18] ($\Lambda = \sqrt{\frac{9}{10}L}$). Sun et al. proposed a common unified framework, called *Hello*, that allowed these protocols (Quorum, Disco, U-Connect, Searchlight) to be compared.

To our knowledge, we are the first to comprehensively study the impact of small slot sizes. Most previous work either picked an arbitrary slot size that worked for their specific hardware platform [4, 3], or used the same slot size as previous work [14]. The slot sizes used include 10 ms [4], 25 ms [4], 50 ms [15] and 2 s [3]. Some reasons cited for not using smaller slot sizes included jitter [7], overflowing slots [15], and slow hardware state transitions [3]. We believe that some of these difficulties reflect the phenomena of slot misalignment exacerbated by clock drift and practical discovery failures, which we describe in §III and §III-B. We have successfully achieved deterministic neighbor discovery at a slot size of 1 ms and 1% duty cycle using our custom sensor platform.

Motivated by the discovery failures observed when working with small time slots, we concluded that it was advantageous to adopt heterogeneous active slots (with different transmit-listen patterns) instead of the traditional homogeneous active slots (with the same transmit-listen pattern for all slots). Most of the previous protocols used the beacon-listen-beacon pattern (a listening period sandwiched by two beacons) proposed in Disco [4]. Our use of heterogeneous slots shares a similar motivation that was recently cited by U-Connect [10] and Nihao [14] ($\Lambda = \alpha\sqrt{L}, \alpha < 1$). U-Connect used small transmit-only slots and combined multiple slots into a single large slot for listening. Similarly, Nihao asserted that slots should be used for transmitting or listening exclusively, and proposed that transmission slots consist of a single beacon followed by sleeping for the rest of the slot. Another strategy that we adopted, like other recent protocols, is to focus on achieving one-sided discovery, since this can be easily extended to mutual discovery via using the information received to predict and send a beacon during the other node's next listening period [14, 9]. We found that at some level one-sided discovery was relatively common once the slot sizes were comparable to the beacon length.

III. What Happens With Small Slots

The neighbor discovery latency L between two sensor nodes with duty cycling is expected to be $L = N \cdot S$, where N is the number of slots needed for neighbor discovery, and S is the

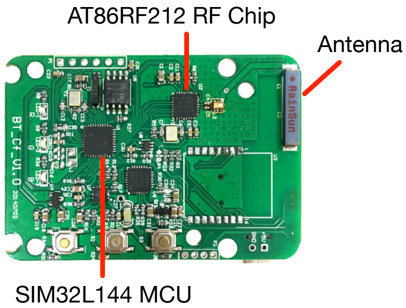


Fig. 1: Custom sensor platform.

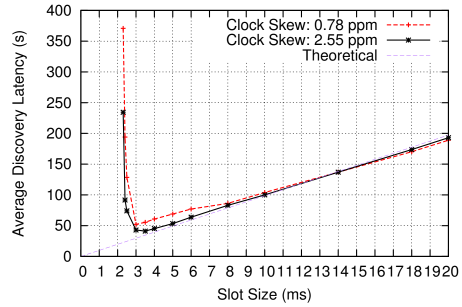


Fig. 2: Average pairwise discovery latency for two pairs of nodes.

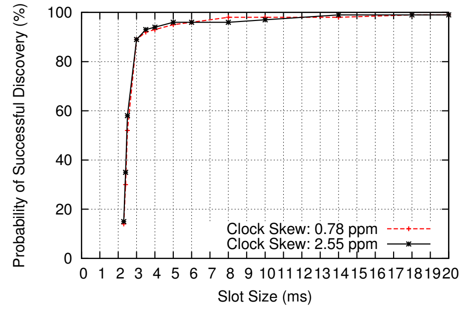


Fig. 3: Probability of successful discovery within the first period.

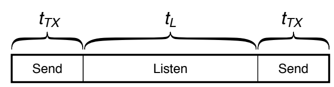
slot size in terms of time. Most existing neighbor discovery protocols focus on reducing N. In contrast, to the best of our knowledge, we are the first to investigate the extent to which we are able to improve latency by reducing the slot size S.

To this end, we use a custom sensor platform (shown in Fig. 1) in our preliminary measurement study. The platform is equipped with a 32-bit ARM Cortex-M3 MCU and a low-power IEEE 802.15.4-compatible radio transceiver AT86RF212. For our study, we used Searchlight [3], which is a state-of-the-art neighbor discovery protocol, at a duty cycle of 1%. While Nihao is a more recent protocol, its theoretical worst-case latency and parametrization suggested that Nihao would perform worse than Searchlight for small slot sizes [14]. In each active slot of Searchlight, two beacons of 11-byte payload are sent. With a data rate of 250 Kbps and the packet structure of a 802.15.4 payload, the beaconing time is 940 μs.

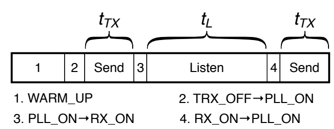
In our measurement study, we focus on one-sided discovery latency, i.e., for a pair of nodes, the discovery process ends whenever one node receives the beacon from the other. In practice, one-sided discovery is sufficient in all the (many) applications where mutual discovery can be reconstructed and used offline. We use two pairs of nodes to show how slot sizes affect the one-sided discovery latency. For each pair of nodes, we randomly set the slot index and slot offset to simulate a real discovery scenario.

We plot the neighbor discovery latency L against different slot sizes S for two pairs of nodes with relative clock skews of 0.78 ppm (parts per million) and 2.55 ppm respectively, and compare them to Searchlight's theoretical worst-case discovery latency under 1% duty cycle [8] in Fig. 2. While the neighbor discovery latency L is expected to decrease proportionally to the reduction in S, our testbed experiments show that L increases drastically when the slot size is reduced beyond a certain point. Essentially, when operating at the *quantum scale*, where the slot size is comparable to the beacon length, the performance of Searchlight changes drastically. We can also see that the average discovery latency is dependent on the relative clock skew, which is surprising.

In Fig. 3, we plot the probability of successful discovery within the first period, which is the theoretical worst-case discovery latency. Our results suggest that the poor performance at small slot sizes is primarily caused by the dramatic increase in *discovery failures* in this regime. In the following sections, we characterize these discovery failures and describe techniques to mitigate them.



(a) Ideal.



(b) Practical.

Fig. 4: Comparison of an ideal and a practical active slot.

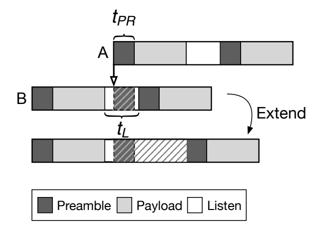


Fig. 5: If B has received A's preamble, B extends its listening time if necessary in order to receive A's payload.

A. Modeling A Real Active Slot

Existing deterministic neighbor discovery protocols [4, 3, 18, 15] generally adopt a beacon-listen-beacon configuration for the active slot shown in Fig. 4(a). In other words, for each ideal active slot, a beacon is transmitted at the start and end, and listening is done in between. This arrangement seems logical since it enables a pair of nodes to hear each other's beacon and achieve mutual discovery when their active slots overlap.

In practice, radio devices experience extra delays such as warm-up and state-transition delays [16, 5]. Based on the analysis of our own hardware platform [1], an active slot generally has 4 delays: a warm-up delay (Component 1 in Fig. 4(b)) and 3 state-transition delays (Components 2, 3 and 4 in Fig. 4(b)). While these delays are negligible at large slot sizes, they become more significant in the quantum regime.

Delays. The *warm-up* delay [1] is the time taken by the internal oscillator and voltage regulator to reach a stable state and is typically on the order of milliseconds [5]. For our hardware platform, the average warm-up delay is 1.3 ms. And the average state-transition delays for components 2, 3 and 4 are 200 μs, 25 μs and 25 μs respectively.

Sending & Listening Time. Sending time is relatively stable and depends on the total beacon length and the adopted data rate. The beacon length consists of user payload and the headers of MAC and physical layers. If we denote l, l_{MAC}, and l_{PHY} as the length in bits of the user payload, MAC and

PHY headers respectively, the time required to send a beacon can be calculated as:

$$t_{\text{TX}}(l) = \frac{l + l_{\text{MAC}} + l_{\text{PHY}}}{\text{PhyRate}} \tag{1}$$

where t_{TX} shall be referred to as the *beacon length*. Unlike the sending time, there may be variations in the listening time. In practice, after detecting a valid synchronization header (SHR), the transceiver automatically enters the `BUSY_RX` state, during which we cannot change its state. Thus, it is possible to set a smaller listening time and to automatically extend active slots once a valid preamble is received, as illustrated in Fig. 5. If we denote the time for receiving the payload as t_{RX} (inclusive of the possible idle period before receiving the preamble) and the time for receiving a preamble as t_{PR}, then the total listening time $t_{\text{L}} = t_{\text{RX}} + t_{\text{PR}}$.

Model. Based on the above analysis, given a beacon length l, the practical slot size S is given by:

$$S = 2t_{\text{TX}}(l) + t_{\text{D}} + \underbrace{t_{\text{RX}} + t_{\text{PR}}}_{t_{\text{L}}} \tag{2}$$

where t_{D} is the sum of warm-up and state-transition delays. It turns out that we can reduce the slot size by shifting the warm-up process to the preceding slot(s) [3] along with the first state-transition delay (Component 2 in Fig. 4(b)). In other words, we compensate for this delay by making a node wake up slightly earlier. Since the last two state-transition delays are negligible, we can simplify the practical slot size model to:

$$S = 2t_{\text{TX}}(l) + t_{\text{L}} \tag{3}$$

B. Practical Discovery Failures

While existing neighbor discovery protocols assume that discovery will always be successful whenever the active slots of two sensor nodes overlap [4, 10, 3], as illustrated in Fig. 6, there are actually 3 possible cases when two active slots overlap: *Discovery Failure*, *One-sided Discovery*, and *Mutual Discovery*. One-sided discovery, where only one node hears the beacon from the other node, is uncommon for large slot sizes and is thus typically ignored. In the quantum regime, one-sided discovery is relatively common because the listening time is comparable to the beacon length. In practice, one-sided discovery is sufficient for all the (many) applications where mutual discovery can be reconstructed and used offline, which is primarily an engineering issue. Thus, we treat one-sided discovery and mutual discovery the same.

Discovery failures occur when beacons overlap and two sensor nodes have to wait for the next overlapping active slots to try to discover each other again. This suggests that the actual neighbor discovery latency L_{actual} is as follows:

$$L_{\text{actual}} = N \cdot S + \Delta T \tag{4}$$

where ΔT is the additional latency caused by discovery failures. $\Delta T = 0$ when there is no discovery failure. The expected neighbor discovery latency $\bar{L}_{\text{actual}}$ can be expressed as:

$$\bar{L}_{\text{actual}} = N \cdot S + P_{\text{fail}} \Delta T \tag{5}$$

Quantifying P_{fail}. By analyzing the discovery failure modes, we found that P_{fail} was dependent on the relative slot offset of the two active slots, the relative sizes of the beacon length (t_{TX}), and the listening time (t_{L}). Fig. 7 shows the detailed analysis of the relative slot offset with the corresponding outcome of discovery. The results for the two possible cases: (i) $t_{\text{L}} \leq t_{\text{TX}}$ and (ii) $t_{\text{L}} > t_{\text{TX}}$, are summarized in Table I.

TABLE I: Duration of different discovery cases.

Condition	t_{fail}	$t_{\text{one-sided}}$	t_{mutual}
$t_{\text{L}} \leq t_{\text{TX}}$	$2t_{\text{TX}} - t_{\text{L}} + 2t_{\text{PR}}$	$t_{\text{L}} - t_{\text{PR}}$	$t_{\text{L}} - t_{\text{PR}}$
$t_{\text{L}} > t_{\text{TX}}$	$t_{\text{TX}} + 2t_{\text{PR}}$	$t_{\text{TX}} - t_{\text{PR}}$	$t_{\text{L}} - t_{\text{PR}}$

If we assume that the relative slot offset of two nodes is uniformly distributed in $[0, S)$, we can derive the following probabilities for the discovery failure:

$$P_{\text{fail}} = \begin{cases} \dfrac{2t_{\text{TX}} - t_{\text{L}} + 2t_{\text{PR}}}{2t_{\text{TX}} + t_{\text{L}}}, & t_{\text{L}} \leq t_{\text{TX}} \\[2ex] \dfrac{t_{\text{TX}} + 2t_{\text{PR}}}{2t_{\text{TX}} + t_{\text{L}}}, & t_{\text{L}} > t_{\text{TX}} \end{cases} \tag{6}$$

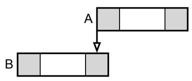
Eq. (6) suggests that the probability of discovery failure increases when slot size decreases, which is consistent with the observations made in our measurement study.

C. Mitigating Collisions & Synchronization

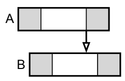
Our key findings in the previous sections are that there are two key challenges when operating in the quantum regime: (i) collisions between the beacons have a non-negligible effect when the beacon length is comparable to the listening time; and (ii) synchronization can cause ΔT to become very large when the relative clock skew between a pair of nodes is small.

Reducing Beacon Density. To reduce collisions, a natural approach is to reduce the number of beacons. We notice that the active slot pattern for Searchlight consists of an anchor slot and a probe slot, where a beacon-listen-beacon pattern is employed. To reduce the density of beacons, we converted the probe slot into a listening slot, i.e., a node will only listen and not beacon during the probe slot. Next, we note that if we preserve the beacon-listen-beacon sequence for the anchor slot, the minimum slot size is twice the beacon length. By reducing the number of beacons for the anchor slot to just one, we can not only reduce the density of beacons, but can also further reduce the slot size. Others have shown that sending one beacon is sufficient for mutual discovery [14]. We call our new version of Searchlight, *Anchor Beacon Probe Listen (ABPL)*. The search pattern of ABPL is illustrated in Fig. 8.

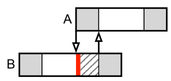
Introducing Randomization to Mitigate Synchronization. In the rare occasion that two nodes have synchronized offsets, ΔT can become very large. Consequently, if relative clock skew is low, it can take a long time for the nodes to desynchronize. To address this, instead of following a rigid wake-sleep pattern, each sensor node jitters its wake-up time by a small amount to decouple the change in slot offset per



(a) Discovery Failure.

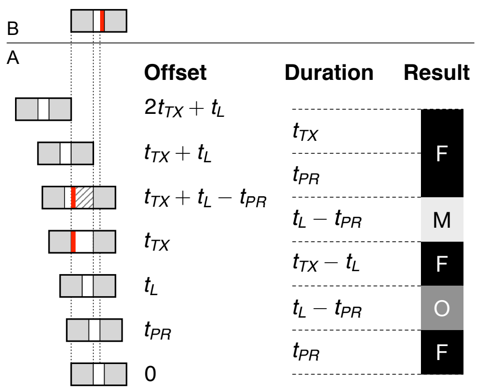


(b) One-sided Discovery.

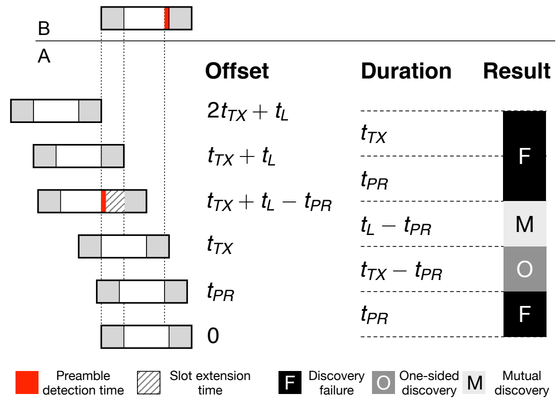


(c) Mutual Discovery.

Fig. 6: Three possible discovery cases when two nodes' active slots overlap.



(a) Discovery cases with $t_L \leq t_{TX}$



(b) Discovery cases with $t_L > t_{TX}$

Fig. 7: Discovery results with different slot size and relative slot offset.

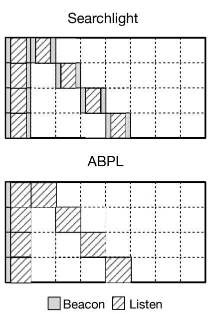


Fig. 8: Searchlight vs. ABPL.

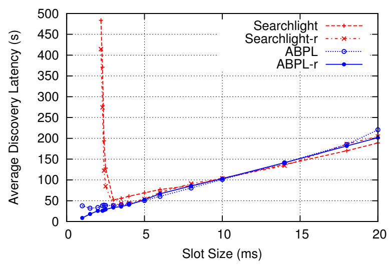


Fig. 9: Improvement from ABPL strategy and randomization technique.

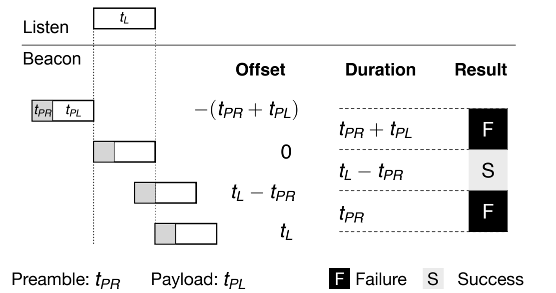


Fig. 10: Discovery failure when preamble falls outside listening period. Note that slot extension still applies.

period from clock skew. The amount of randomization introduced is proportional to the length of the beacon preamble. With randomization, we no longer have to wait for the slot offset to slowly drift to a discoverable region when a pair of nodes happen to be at a particular slot offset that causes discovery failure. We can see the effectiveness of ABPL and randomization in Fig. 9. The results show that randomization is able to reduce the worst-case (long tail) latency significantly for both Searchlight and ABPL.

IV. Focused Listening with Spotlight

In ABPL, anchor slots still adhere to the beacon-listen pattern. The beacon length is fixed, so as we decrease the slot size, the listening period during the anchor slot is reduced correspondingly. Once the listening period is shorter than the beacon length, the probability that the anchor slot will successfully "catch" a beacon becomes extremely low, and in most cases, the overlaps are the beacons from anchor slots and the listening period of probe slots.

In the limit, we are essentially left with two kinds of active slots: beacon (B) and listen (L). We refer to them as *heterogeneous slots* with different beacon-listen activities, in contrast to the conventional homogeneous active slots that use only the beacon-listen-beacon strategy [4].

A. Probing Considered Harmful

Like Searchlight, ABPL distributes the listening time in a period over a series of probe slots. While Searchlight introduced this probing pattern to reduce average latency, we found that this probing pattern also introduced discovery failures in the quantum regime similar to those discussed in §III-B. The key observation is that the effective listening time of a listening slot is smaller than its length because it would be unable to decode a packet if the preamble did not fall in its entirety within the listening slot. This failure mode is illustrated in Fig. 10, and is analogous to the modes described earlier in Fig. 7. We observe from Fig. 10 that with a listening slot of length t_{L}, the effective listening time is only $t_{\mathrm{L}} - t_{\mathrm{PR}}$. Thus, the probability of discovery failure is:

$$P_{\mathrm{fail}} = \frac{t_{\mathrm{PR}}}{t_{\mathrm{L}}} \tag{7}$$

When operating in the quantum regime, the preamble occupies a substantial portion of a slot and this causes failures that have a significant impact on latency. For example, on our hardware platform with a beacon length of 1 ms and a preamble length of 0.2 ms, the effective failure probability is ~20%.

Our key insight is that we can eliminate most occurrences of this failure mode by combining all probe slots into one long listening slot for each period. In particular, by combining n probe slots into a single continuous listening slot, the failure probability becomes:

$$P_{\mathrm{fail}} = \frac{t_{\mathrm{PR}}}{n \cdot t_{\mathrm{L}}} \tag{8}$$

In contrast to Searchlight, which spreads out its probe slots, we call our scheme *Spotlight*, since our listening is focused at a fixed spot and we have a long listening interval during each period. Comparing Eqs. (7) and (8), we find that Spotlight has a much lower failure probability than Searchlight and ABPL. For example, compared to ABPL with 1% duty cycle, Spotlight reduces the failure rate from 20% to 0.2%, a two order of magnitude reduction. Spotlight also adopts the randomization technique used in ABPL-r. Since failure is caused by missing the preamble, we set the range of randomization to be the length of one preamble.

B. Achieving Discovery Guarantees

To analyze Spotlight, we consider a class of pattern matrices $\mathbb{M}(m, n, a, b)$, that we call *BL (Beacon-Listen) diagrams*. Each matrix represents a pattern of beacon, listening and sleep slots over a period with $m \cdot n$ slots. As illustrated in Fig. 11(a), the first a slots in the first column are all beacon slots, and the next b slots in the first row adjacent to the first beacon slot are listening slots. An alternative form, where the top left slot is a listen slot, is shown in Fig. 11(b). Spotlight is represented by Variant 1 of the BL diagram, with $\mathbb{M}(m, 2m, m, m)$, where m can be derived from the desired duty cycle setting.

Clearly, not all configurations of m, n, a, b guarantee discovery. To ensure discovery happens within one period, the BL diagram must satisfy two conditions:

1) Either $a = m$ (Variant 1) or $b = n$ (Variant 2), i.e., either beacon slots must fill up the entire column, or the listening slots must fill up the entire row.
2) If $a = m$, then $b \geq \lfloor \frac{n}{2} \rfloor$ (Variant 1). Otherwise, if $b = n$, then $a \geq \lfloor \frac{m}{2} \rfloor$ (Variant 2). In other words, with one filled row or column, the remaining row or column has to be longer than half the maximum length.

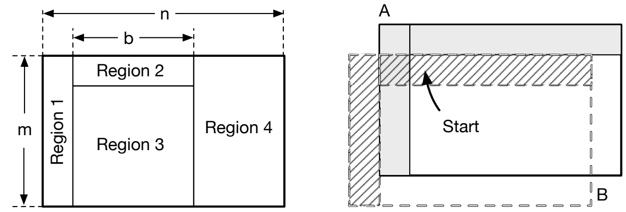
Proof of Condition 1: For Variant 1, b is always smaller than n. Suppose $a < m$. If we let slot $(0, 0)$ of node B overlap with slot $(m - 1, 0)$ of node A, then there will not exist a beacon-listen overlap between A and B (see Fig. 12(a)) for any value of b. Therefore, $a = m$.

Similarly, for Variant 2, a is always smaller than m. Suppose $b < n$. If we let slot $(0, 0)$ of node B overlap with slot $(0, 1)$ of node A, then there will not exist a beacon-listen overlap between A and B (see Fig. 12(b)). Therefore, $b = n$.

Proof of Condition 2: By symmetry, it suffices to consider Variant 1, i.e., the case $a = m$. Suppose $b < \lfloor n/2 \rfloor$. There will then be $b' = n - b - 1 > b$ empty columns at the right side of the matrix. If we let slot $(0, 0)$ of node B overlap with slot $(0, b + 1)$ of node A, then there will not exist a beacon-listen overlap between A and B (see Fig. 13). Therefore, $b \geq \lfloor n/2 \rfloor$.

Claim: If Conditions 1 and 2 hold, discovery is guaranteed within one period.

Proof: Consider two nodes A and B with the same BL diagrams. Without loss of generality, we consider the case $a = m$ and let $B(0, 0)$ (a beacon slot) overlap with every slot $A(i, j), (0 \leq i < m, 0 \leq j < n)$. We divide A's slots into four regions (see Fig. 14(a)), and consider the discovery outcome within each region exhaustively as follows:



(a) Enumeration of all possible overlap scenarios.

(b) Worst-case overlap scenario when $a = m$ and $b = n - 1$. Latency for any valid b given this offset is $m \cdot n - 1$ slots.

Fig. 14: Possible overlap cases

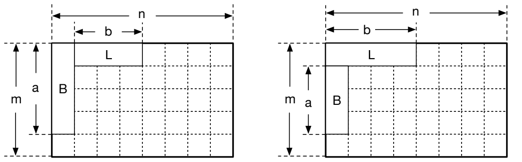
- **Region 1**. All slots in Region 1 are beacon slots. As a result, beacon-beacon overlap always occurs, and discovery will fail. This will be repeated for all the beacon slots of A and B. In other words, Region 1 is an unavoidable "dead zone", as seen in other deterministic protocols. Spotlight incorporates randomization to avoid being trapped in this zone for a prolonged period.
- **Region 2**. All slots in Region 2 are listening slots, resulting in beacon-listen overlap. Discovery is successful with no delay.
- **Region 3**. Since Condition 1 is satisfied, there will exist at least one beacon in B that overlaps with a listening slot in Region 2 of A. Discovery will happen within one period.
- **Region 4**. Since Condition 2 is satisfied, the width of Region 4 is smaller than the width of Region 2. Therefore, one of B's subsequent listening slots $B(0, i), 0 < i \leq b$ will overlap with A's beacon slots. Discovery will happen within one period.

We have shown that beacon-listen overlap is guaranteed no matter how A and B's slots overlap. As A and B fit on the same BL diagram, the position of the overlap between A, B is the same in each period. Thus, given the current beacon-listen overlap, the next beacon-listen overlap will happen within $m \cdot n$ slots, i.e., discovery is guaranteed within one period.

C. Worst-case Discovery Latency

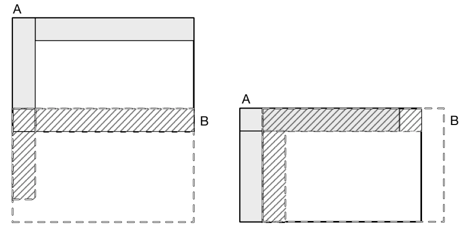
Spotlight. Since Spotlight adopts the Variant 1 configuration, we have $a = m$. From Condition 2, we have $b \geq \lfloor n/2 \rfloor$. A smaller b translates to a lower duty cycle and less power used. However, counter-intuitively, choosing a larger $b > n/2$ does not improve worst-case latency. Consider the worst-case slot offset and starting position where $A(1, 1)$ overlaps with $B(0, 2)$, as illustrated in Fig. 14(b). The discovery would take $mn - 1$ slots for $\lfloor n/2 \rfloor \leq b < n - 1$ and $mn - 2$ slots for $b = n - 1$.

Consequently, for a fixed set of constants m and n, setting $a = m$ and $b = \lfloor n/2 \rfloor$ would minimize energy consumption, while ensuring discovery within one period. It thus makes sense to first decide on the desired duty cycle C and then set m and n accordingly to minimize the worst-case discovery latency. While beaconing typically requires more energy than listening, the energy consumption for beaconing in modern



(a) Variant 1: $(0, 0)$ is a beacon slot. (b) Variant 2: $(0, 0)$ is a listening slot.

Fig. 11: Two variants of BL diagrams.



(a) Variant 1. (b) Variant 2.

Fig. 12: Construction to prove Condition 1.

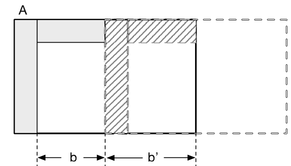


Fig. 13: Construction to prove Condition 2.

sensor platforms is only slightly more [1, 10] ($\leq 50\%$). For simplicity, we assume that energy consumption is the same, allowing us to express C as:

$$C = \frac{a+b}{mn} \tag{9}$$

(If so desired, the difference in energy consumption can be accounted for by adding an extra constant term to this analysis.)

Since $a = m$ and $b = \lfloor n/2 \rfloor$, we may assume without loss of generality that n is an even integer and substitute $a = m, b = n/2$ into Eq. (9) to obtain:

$$n = \frac{2m}{2Cm - 1} \tag{10}$$

Since the worst-case latency is one period (i.e., mn slots), we have:

$$L = mn = \frac{2m^2}{2Cm - 1} \tag{11}$$

From Eq. (11), L has one global minimum at $m = 1/C$. Assuming $1/C$ is an integer, we can substitute $m = 1/C$ into Eq. (10) to derive $n = 2/C$ and thus $n = 2m$. Hence, Spotlight's pattern achieves the best worst-case latency for a specified duty cycle.

Spotlight-Transposed (Spotlight-T). It turns out that for Variant 2, i.e., the case of $b = n$, we can repeat the same analysis and obtain an analogous conclusion that the minimum latency is achieved when $m = 2n$. Visually, we can transform Spotlight to this configuration, by transposing Spotlight's BL diagram, followed by swapping the beacon and listening slots. Hence, we call this alternative *Spotlight-Transposed* (or Spotlight-T). The worst-case discovery latency for Spotlight and Spotlight-T are equivalent, but we shall see in §V-C that the average latencies for Spotlight-T are slightly worse. The key difference is that the beacons for Spotlight are uniformly distributed over time, while those for Spotlight-T are not distributed uniformly. Intuitively, it is not surprising that the former achieves better average-case latencies. We leave the detailed analysis as future work.

Comparison with Nihao. We noticed that Nihao, when used with the minimum slot size (i.e., $\alpha = 1$), can also be described by our BL diagram, with a parametrization of $\mathbb{M}(m, n, m, n-1)$, where m and n are determined by the duty cycle. We illustrate the patterns for Spotlight, Spotlight-T and Balanced Nihao in Fig. 15. Since we proved that

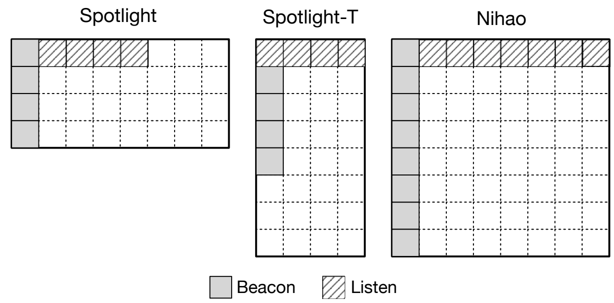


Fig. 15: Comparing Spotlight, Spotlight-T and Balanced Nihao at a duty cycle of 25%.

TABLE II: Parameters for evaluated algorithms.

Algorithm	Duty Cycle	Period (slots)	Active slots
ABPL-r	1%	20,000	100BL+100L
B-Nihao	1%	$10,000(1+\alpha)^2$	1BL+199B+199L
Searchlight	1%	20,000	200BLB
Spotlight	1%	20,000	100B+100L
Spotlight-T	1%	20,000	100B+100L
ABPL-r	5%	800	20BL+20L
B-Nihao	5%	$400(1+\alpha)^2$	1BL+19B+19L
Searchlight	5%	800	40BLB
Spotlight	5%	800	20B+20L

Spotlight and Spotlight-T have optimal worst-case latencies among all possible BL diagram configurations, it holds that Nihao's pattern is suboptimal and is bettered by Spotlight and Spotlight-T. In particular, for the same duty cycle, Balanced Nihao's worst-case latency is approximately twice that of Spotlight and Spotlight-T. Besides having a better beaconing pattern, a key advantage of Spotlight/Spotlight-T over Nihao is the use of randomization to avoid beacon synchronization.

V. Performance Evaluation

In this section, we present our evaluation results comparing Spotlight to existing state-of-the-art neighbor discovery protocols using a practical testbed with custom sensor hardware.

A. Testbed and Experiment Setup

Testbed Design. Out testbed is capable of measuring the states of up to 20 sensor nodes simultaneously. Sensor nodes are hardwired to a central controller for probing and all sensor events are generated by pulling sensors' IO pins up and down to minimize latency. To further ensure that our measurements are accurate, the controller is equipped with an Oven-Controlled Crystal Oscillator (OCXO) with 1 ppb temperature stability to serve as a precision clock.

We implemented Spotlight and other reference protocols on our custom sensor hardware (shown in Fig. 1). The sensor nodes do not run any sensor operating system and are programmed directly in C. This is an advantage for our evaluation since it has been shown that sensor operating systems such as TinyOS often introduce additional jitters [4, 19]. The beacon contains a 11-byte payload which includes necessary information for discovery such as node ID, slot index, number of periods and running rounds. The total length of the underlying physical message is 28 bytes (4 B of preamble + 1 B of SFD + 1 B of PHR + 9 B of MHR + 11 B of MAC payload + 2 B of FCS). At the data rate of 250 Kbps, it takes 0.94 ms to transmit one beacon.

Experimental Setup. We evaluated the protocols at duty cycles of 1% and 5% since they are commonly used in practice [4]. The parameters for different neighbor discovery protocols are shown in Table II. For each protocol, we randomly picked the initial slot index and starting delay (in the range of one slot size) for the 20 nodes. Once a discovery was successful between a particular pair of nodes[1], the current discovery process ends, and a new experiment is started. The experiments were repeated 100 times for each protocol.

B. Determining optimal amount of jitter

As stated in §III-C, varying the wake up time by adding a small amount of jitter helps to alleviate slot synchronization. The goal is to adjust the relative offset so as to achieve discovery in the next period; however, before discovery a node has no knowledge of the current relative offset.

To investigate the impact of the degree of jitter, we conducted a measurement study on our testbed. Fig. 16 shows that the randomization scheme can drastically reduce the long tail of discovery latencies. Thus, for our following experiments, we set the amount of jitter to [0, 200) μs.

C. Comparison to the State-of-the-Art

In Fig. 17, we compare Spotlight with the two state-of-the-art algorithms Nihao and Searchlight at 1% duty cycle. As expected, Searchlight implodes once the slot size falls below 3 ms (3$\times$ beacon length). We observe that except Searchlight, all the other algorithms evaluated could operate at the minimum slot size of 1 ms (i.e., one beacon length). This suggests that the beacon-listen-beacon pattern is not suitable for operating at the quantum scale due to high additional latencies caused by discovery failures ($P_{\text{fail}}\Delta T$).

Even though Nihao has a similar worst-case latency and power-product latency Λ as other protocols, it performed surprisingly well at small slot sizes relative to other protocols. In a similar vein, Spotlight has the same $\Lambda = \sqrt{2L}$ as Searchlight but performs much better. This can be explained by how $\Lambda = DC \cdot L$ considers worst-case latency $N \cdot S$ but not the $P_{\text{fail}}\Delta T$ term in Eq. (5). This demonstrates the presence of gaps in existing theoretical models and analysis when it comes to predicting performance in the quantum regime and taking into account practical discovery failures.

It is plausible that Generic Nihao can be tuned to achieve somewhat better performance than the Balanced Nihao that we evaluated, but theoretically, it is not possible for Nihao to match Spotlight in worst-case performance as shown in §IV-C. We also note that Spotlight-T performs slightly worse than Spotlight in the average case and that Spotlight achieves approximately 50% lower discovery latency relative to Balanced Nihao.

In Figs. 18 and 19, we plot the cumulative distributions for the discovery latencies at slot sizes of 2.5 ms and 1 ms, respectively. Unlike Searchlight and ABPL-r, Spotlight and B-Nihao do not exceed their theoretical worst-case discovery latencies of one period, which are 20 s (20,000 slots) and 40 s (40,000 slots) respectively, as shown in Table II (given 1 ms slot size and $\alpha = 1$ for Nihao.)

D. Performance at Higher Duty Cycle

In Fig. 20, we plot the discovery latency against slot size at 5% duty cycle. We did not plot Searchlight in Fig. 20, because its discovery latencies were in the 14 s to 416 s range and much worse than the algorithms presented in the figure.

One key observation is that while Nihao adopts heterogeneous slots and contiguous listening to reduce the probability of discovery failure, at 5% duty cycle we can observe unusually high latency variations at small slot sizes, indicating that discovery failure has substantial effect on the practical discovery latency. This is because at 5% duty cycle, the length of one period becomes much shorter (more than an order of magnitude difference between 1% and 5% duty cycle), so that the equivalent clock drift in one period becomes much smaller. As a result, latency becomes more vulnerable to clock skew.

To verify our hypothesis, we evaluated a modified version of Nihao with slot randomization ("Nihao-r"). For comparison, we evaluated a version of Spotlight that did not incorporate randomization ("Spotlight-nr"). Nihao-r seemed to achieve good performance while Spotlight-nr performed poorly. This suggests that our randomization technique is critical in allowing Spotlight to achieve low latencies, and that it can be generally applied to existing neighbor discovery protocols.

We can see from Fig. 21 that the cumulative distributions for the randomized and non-randomized versions of the various algorithms are very similar in shape. The key difference is that the randomized versions are able to achieve theoretical worst-case latencies with a higher probability, while that of non-randomized algorithms can exceed the theoretical worst-case latency by up to two orders of magnitude.

VI. Conclusion

Our investigation of the impact of clock skew and beacon collisions when operating at the quantum scale suggests that there was a gap in the existing theoretical models for modeling worst-case and average-case discovery latency. We present and evaluate *Spotlight*, which incorporates randomization and a

[1] We randomly picked a pair of nodes, which had a 0.78 ppm relative clock skew.

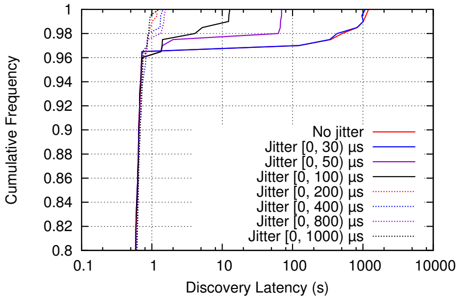


Fig. 16: Discovery latency CDF for 1 ms slot size of Spotlight with varying jitter at 5% duty cycle.

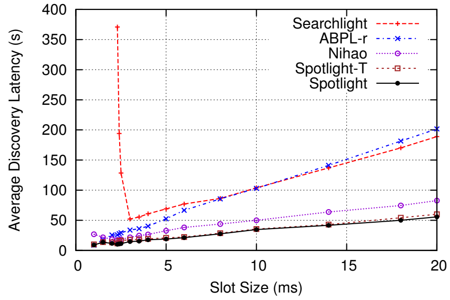


Fig. 17: Average neighbor discovery latency at 1% duty cycle.

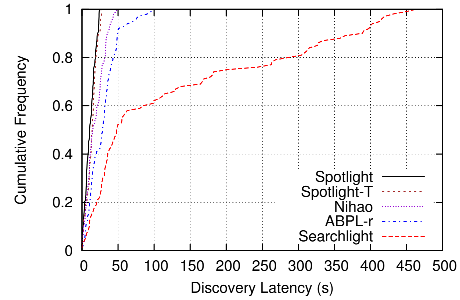


Fig. 18: Discovery Latency CDF for 2.5 ms slot size at 1% duty cycle.

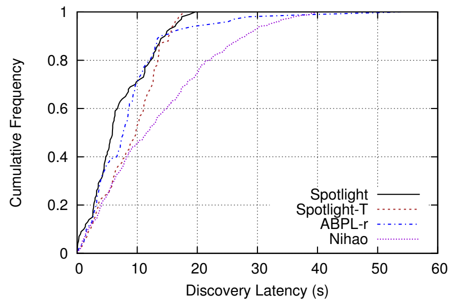


Fig. 19: Discovery Latency CDF for 1 ms slot size at 1% duty cycle.

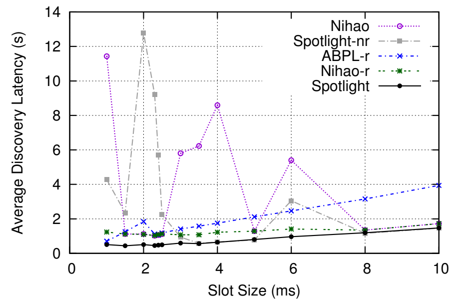


Fig. 20: Average neighbor discovery latency at 5% duty cycle.

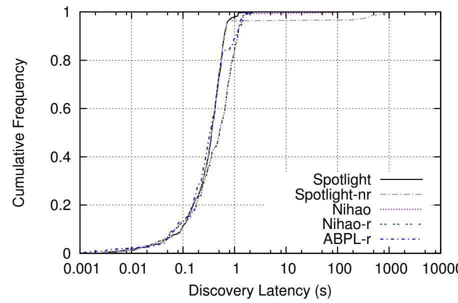


Fig. 21: Discovery latency CDF for 1 ms slot size at 5% duty cycle.

provably optimal beaconing strategy. While slot offset randomization may be a simple idea, we show that it is an important and generally applicable mechanism in the quantum regime for higher duty cycles. To the best of our knowledge, we are the first to demonstrate neighbor discovery at a slot size of 1 ms in a practical sensor network. Our approach will lead to significantly lower discovery latencies as sensor hardware continues to improve and support smaller beacons.

Acknowledgements

This work was carried out at the SeSaMe Center, supported by Singapore NRF under the IRC@SG Funding Initiative, the CETNET NGI Technology Innovation Project (No.NGII20160114) and the Cultural Conservation Technology Project in Zhejiang Province (No. 2017007).

References

[1] Manual of at86rf212, http://www.atmel.com/images/doc8168.pdf, 2010.

[2] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3):537–568, 2009.

[3] M. Bakht, M. Trower, and R. H. Kravets. Searchlight: Won't you be my neighbor? In *Proceedings of MobiCom '12*, Aug. 2012.

[4] P. Dutta and D. Culler. Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In *Proceedings of SenSys '08*, Nov. 2008.

[5] V. Galluzzi and T. Herman. Survey: Discovery in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2012.

[6] K. Han, J. Luo, Y. Liu, and A. V. Vasilakos. Algorithm design for data communications in duty-cycled wireless sensor networks: A survey. *IEEE Communications Magazine*, 51(7):107–113, 2013.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. *ACM SIGOPS Operating Systems Review*, 34(5):93–104, 2000.

[8] S. Jin, Z. Wang, W. K. Leong, B. Leong, Y. Dong, and D. Lu. Improving neighbor discovery with slot index synchronization. In *Proceeding of MASS '15*, Oct. 2015.

[9] C. Julien, C. Liu, A. L. Murphy, and G. P. Picco. Blend: Practical continuous neighbor discovery for bluetooth low energy. In *Proceedings of IPSN '17*. ACM, 2017.

[10] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar. U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol. In *Proceedings of IPSN '10*, Apr. 2010.

[11] S. Lai, B. Ravindran, and H. Cho. Heterogenous quorum-based wake-up scheduling in wireless sensor networks. *IEEE Trans. Comput.*, 59(11):1562–1575, 2010.

[12] M. J. McGlynn and S. A. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *Proceedings of MobiHoc '01*, Oct. 2001.

[13] I. Niven, H. S. Zuckerman, and H. L. Montgomery. *An Introduction to The Theory of Numbers*. John Wiley & Sons, 2008.

[14] Y. Qiu, S. Li, X. Xu, and Z. Li. Talk more listen less: Energy-efficient neighbor discovery in wireless sensor networks. In *Proceedings of INFOCOM '16*, 2016.

[15] W. Sun, Z. Yang, K. Wang, and Y. Liu. Hello: A generic flexible protocol for neighbor discovery. In *Proceedings of INFOCOM '14*, April 2014.

[16] W. Sun, Z. Yang, X. Zhang, and Y. Liu. Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey. *Communications Surveys and Tutorials, IEEE*, 16(3):1448–1459, 2014.

[17] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh. Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks. *Computer Networks*, 43(3):317–337, 2003.

[18] K. Wang, X. Mao, and Y. Liu. BlindDate: A neighbor discovery protocol. In *Proceedings of ICPP '13*, Oct 2013.

[19] 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 SenSys '12*, Nov. 2012.

[20] R. Zheng, J. C. Hou, and L. Sha. Asynchronous wakeup for ad hoc networks. In *Proceedings of MobiHoc '03*, 2003.