Talk Half Listen To Half: Energy-Efficient Neighbor Discovery in Wireless Sensor Networks

Raudel Ravelo Suarez, Amiya Nayak
School of Electrical Engineering & Computer Science
University of Ottawa, Ottawa, Canada

Abstract— Due to the combination of constrained power, low duty cycle, and high mobility, neighbor discovery is one of the most challenging problems in wireless sensor networks. Existing discovery designs can be divided into two types: pairwise-based and group-based. The former schemes suffer from high discovery delay, while the latter ones accelerate the discovery process but increase transmission package size or incur too much energy overhead, far from practical. Guided by the *Talk More Listen Less* (TMLL) principle (published in 2016), in which beacons are not necessarily placed in the wakeup slots, we propose a different version of a group-based protocol, we call *Talk Half Listen Half* (THLH). For the first time, a group-based protocol uses the *Channel Occupancy Rate* (COR), one of the fundamental novel components of the TMLL model, for performance improvements, in the same way, Duty Cycle (DC) was used in previous group-based protocols. The proposed protocol uses low transmission overhead in comparison with previous group-based discoveries.

1 INTRODUCTION

With the proliferation of portable computing platforms and small wireless devices, mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs) have received more and more attentions as a means of data communication among devices regardless of their physical locations [1]. Wireless devices usually rely on portable power sources (batteries) to provide the necessary power. Among all extensive studies on minimizing power consumption, neighbor discovery is one of the fundamental components focusing on communication and access.

Discovery among neighboring nodes or neighbor discovery serves as a prerequisite for both types of networks in order to achieve connectivity [2]. Only after an initial discovery can a node set up communications with others. However, the difficulty arises when trying to achieve the discovery among low-powered devices without any sort of previous knowledge. The latter refers to the assumption of nodes not having any information about their neighboring nodes such as position, power usage or further transmissions. The ultimate goal is to minimize the discovery latency, a term that refers to the time taken for making contact with neighboring nodes.

In order to reduce power consumption, nodes enter into a low-power sleep mode whenever possible. In this state, a node is neither able to transmit not to listen. On the contrary, nodes stay into what is called an active state whenever listening and/or transmitting. As the amount of energy used in the sleep mode is infinitesimal when compared with that of an active state, the former is not taken into account. The power usage is defined by the fraction of time consumed in the active state over the total time. However, it is more common to see it as the percentage of time a node spends in an active state, known as the duty cycle (e.g. 5%). In practical applications, nodes turn the radio on/off from time to time to satisfy a given energy constraint, being the percentage of time that the radio is on, the duty cycle.

Among the extensive research literature on neighbor discovery duty cycle and discovery latency are usually of top concern. Generally, worst-case and average discovery latency have been adopted by most research work. In any case, it is desirable to have a low duty cycle and minimum discovery latency; having low average discovery latency also a point of interest. It is not difficult to observe a trade-off between duty cycle and discovery latency. A lower duty cycle usually leads to 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.

In this paper, we study different neighbor discovery protocols and their approaches to develop better solutions for the neighbor discovery problem. We propose a new group-based protocol for the TMLL model where the COR usage is not fixed. Our protocol uses the approach of mutual assistance while keeping the energy for listening constantly. It uses mutual assistance with a strategy for keeping the limit on the COR usage up to twice as much the initial one.

2. RELATED WORK

2.1 Pairwise-Based Protocols

The main focus of pairwise-based protocols is the discovery between a single pair of nodes. Through the years, there have been several proposals for asynchronous neighbor discovery that can work under both symmetric and asymmetric duty cycles. For example, a family of Birthday protocols [8] is representative of probabilistic designs where there is no latency bound guaranteed for the discovery. Moreover, prime-based protocols as Disco [4] and U-Connect [7], provide a latency bound. Another state-of-the-art deterministic protocol is Searchlight [10] which achieves high efficiency by leveraging constant offset between periodic active slots. The main drawback of Searchlight comes exactly from this approach for handling asymmetry, the requirement of duty cycles to be power-multiples of the smallest one results in only a small number of options available (e.g., at most four for a targeted span of duty cycles ranging from 1% to 10%). On the other hand, there are also redundant discovery opportunities within a latency bound making room for improvement. Hello [11] provides a generic framework that incorporates existing deterministic protocols such as Quorum, Disco, U-Connect, and Searchlight and by exploring its parameter space finds optimal parameters for symmetric and asymmetric duty cycles. In other works such as Blind Date [5] and G-Nihao [6] beacons are not necessarily placed inside the active slots, however, it is the latter who proposes the complete independence of beacons and listening slots, leading to a redefinition of the listen-listen model. Finally, Lightning [19]

978-1-7281-1340-1/19/$31.00 ©2019 IEEE

applies mutual assistance to achieve two-way discovery, a concept described as: after one-way discovery, the bi-directional discovery can be achieved easily by the node receiving the packet proactively replying in the sender´s next available listening slot.

2.2 Group-Based Protocols

The collaborative approach together and the use of extra information (shared while transmitting) is what set the main difference between group-based and pairwise protocols.

Many of these protocols are set on top of an underlying pairwise-based protocol. Because of their nature, the performance varies depending on the network properties and the specific protocol's approach. Generally, better results are obtained with a higher number of nodes [14] where more information is shared among them while pairwise scenarios are the most difficult to improve. There are many different approaches (e.g. [15], [16], 18]) on how to manipulate the information acquired.

3. PRELIMINARIES

3.1 Time-Slotted Model (Listen-Listen Model)

A time-slotted model (also known as Listen-Listen model) is usually adopted for analysis simplicity. Continuous time is separated into the 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 (as shown in Fig. 1). In an active slot, a node transmits and listen, in an idle slot it does nothing (stays in a power saving mode).

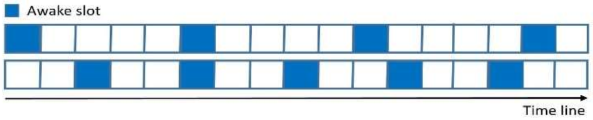

Figure 1: Aligned slots for X and Y with different schedules

3.2 Asynchronism

As nodes work independently and do not set up a global time reference, they may have to discover each other with a displacement among slot boundaries known as asynchronism. As a result, one node may not be able to discover the other even if their active slots overlap. As we can see in Fig. 2, when node x is active for the second time, node *y*'s state changes from idle to active in the middle of x's active slot. Hence, node *x* can successfully receive the beacon from node *y* and recognize *y* as its neighbor but *y* is unaware of *x's* presence as the beacon from *x* arrives at *y* when *y* is idle.

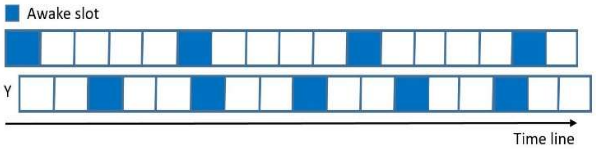

Figure 2: Slot un-alignment example

3.3 Talk-Listen (TMLL) Model

Most neighbor discovery protocols have been designed under the traditional model seen in Section 3.1. In this case, whenever a node is active, it transmits and listens within the same slot. Over the years, authors have come up many others with slight changes; for example, Blind Date [4] proposes a solution where beacons are not necessarily placed in the wakeup slot but next to it, and Lightning [19] defines a new type of slot where there is a beacon part and an idle part. However, it was in Talk More Listen Less: Energy-Efficient Neighbor Discovery in Wireless Sensor Networks [5] where nodes are allowed to beacon anywhere outside from the wake-up slots. The idea comes from the fact that a short beacon is more energy-efficient than an active slot, hence, idle-listening can significantly be reduced by decreasing wake-up slots and increasing the number of beacons sent. We can see this approach as a separation of the active slot into two: an active listening slot and an active transmitting slot. In this way, a node can decide whether to listen or to transmit (may be at the same time, see Figure 3: Talk-Listen model example).

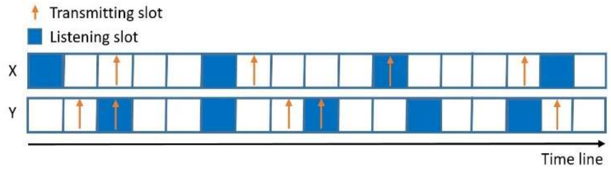

Figure 3: Talk-Listen model example

The discovery schedule of a node m in Talk-Listen model is defined as two binary functions: $\psi L(m,t)$ and $\psi B(m,t)$, representing the schedule of wakeup slots and beacons at time t, respectively [7].

$$\psi L(m,t) = \begin{cases} 1, \text{ listen for a slot} \\ 0, \text{ sleep} \end{cases}$$

$$\psi B(m,t) = \begin{cases} 1, \text{ send a beacon and back to sleep} \\ 0, \text{ sleep} \end{cases}$$

In this way, neighbor discovery can be defined with $\psi L(m,t)$ and $\psi B(m,t)$. If node m1, m2 can directly communicate with each other, a unidirectional neighbor discovery meaning m1 discovers m2 is defined as:

$$\exists t|\ \psi L(m1,t) = \psi B(m2,t) = 1$$

Applying the analog reasoning, a bidirectional neighbor discovery meaning m1 and m2 discover each other is defined as:

$$iff\colon \exists t1,t2|\ \psi L(m1,t1) = \psi B(m2,t1) = 1 \ and\ \psi B(m1,t2) = \psi L(m2,t2) = 1.$$

Since the concern in applications that involve neighbor discovery is energy-efficiency and discovery rate, the key metrics considered by existing discovery protocols are duty-cycle (DC) and worst-case discovery latency (L). When representing them in this model, the duty-cycle DC of a given periodic discovery schedule with period T is:

$$DC = \frac{1}{T}(\sum_{t=0}^{T-1} \psi L(m,t) + \alpha((\sum_{t=0}^{T-1} \psi B(m,t) - N_c))$$

where Nc represents the number of common active slots that satisfy:

$$\psi L(m,t) = 1 \text{ and } \psi B(m,t) = 1$$

Subtracting Nc is to avoid double counting when wakeup and beacon occur in the same slot. α is the proportion between the time for transmitting a beacon and the duration of a wake-up slot. None of the existing discovery protocols has considered α, since each beacon is placed inside the wakeup slot. However, α cannot be ignored when beacons are separated with wakeup slot in the Talk-Listen model. Although the beacon is a short packet that can be broadcasted in less than 1ms with an IEEE 802.15.4 compatible radio, it will dominate the duty-cycle especially when the TMLL principle is aggressively adopted [7].

3.4 Channel Occupancy Rate (COR)

COR measures the degree a discovery protocol occupies the channel or in other words, the fraction of the time that the channel is occupied. In a discovery schedule cycle, COR is defined as:

$$COR = \alpha * \frac{N_B}{T}$$

where $N_B = \sum_{t=0}^{T-1} \psi B(m,t)$ and since in each slot exists at most one beacon in Talk-Listen model, a simplified η is used to represent the COR:

$$\eta = \frac{COR}{\alpha} = \frac{N_B}{T}$$

We list the main points that drive the development of our work below:

- The G-Nihao protocol is a pure pairwise-based protocol, and it does not make use of any group-acquired information in such scenarios.

- Analysis of how G-Nihao achieves the first contact between a pair of nodes highlighted a significant percentage with high latencies.

Finally, even when G-Nihao improved all previously known neighbor discovery protocols, we believe that there still are many previous ideas that can be used, modified and adapted for its improvement, as well as new ones that may come up along this process.

4. TALK HALF LISTEN-2-HALF (THL2H)

The TMLL model brings new possibilities to all protocols (either pairwise or group-based) designed within these rules. Now, nodes can use a given energy budget in two different ways: to listen more and/or to transmit more. Taking into account that the energy needed for beaconing is considered to be a fraction of that for listening, the use of a given budget may yield better results spent on transmissions than on listening.

In a group scenario, this led to the possible idea for nodes to variable increase the COR (i.e. send more transmissions per period, shown in Section 3.4) when necessary while keeping the same energy used for listening. On one hand, as previous group-based protocols have been based on the Listen-Listen model, they increase both the energy used for listening and transmissions as they cannot split them. On the other hand, pairwise scenarios may be ideal for not to increase the COR usage as each node only have to focus on a single neighboring node.

4.1 General Idea

When analyzing how discoveries between two nodes occur in G-Nihao, we focused on two main points:

1. The latency of the one-way discovery, in other words, the latency of the first node in listening to the other or latency of the first contact.

2. The latency between the one-way discovery and the two-way discovery which completes the discovery process (both nodes listened to each other) in a pairwise scenario.

4.2 Model

Taking G-Nihao´s schedule as a starting point, we divide the listening row from G-Nihao into two listening halves. The first $m/2$ slots are kept untouched but the others are placed after $T/2 = m * n/2$ slots as seen in Fig. 5. Then, we add a fraction of a slot (δ) after each half, big enough to be able to listen to a transmission. In this way, the fixed transmissions are kept untouched at the beginning of each row.

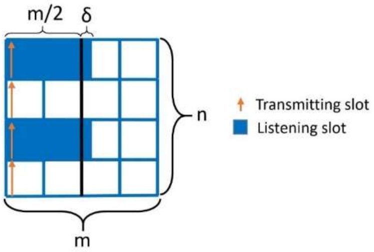

Figure 5: Matrix representation of the THL2H Fixed schedule for m = n = 4

From a node's point of view, all transmissions from any neighboring node with a relative offset greater than $\frac{m}{2} + \delta$ congruent m are missed. However, as the sum of the offsets is equal to m, then a node either listens to its neighbor or it is listened to by its neighbor. In this way, the one-way discovery is guaranteed between any pair of neighboring nodes.

Notice how the second half of every row is not covered by any listening slot (but for the small portion in δ in Fig. 6). As the fixed transmissions are equally distributed, any neighboring node whose offset falls within the second half of any row cannot be listened. As this accounts for 50% the total number of possible offsets, it is easy to see that only one node in every pair is able to make contact with its neighbor.

As an example, Fig. 6 shows how node A is able to listen to B's transmissions and B cannot. However, by moving B's offset one slot further, we fall into the exact opposite situation. It also shows how a node either listens to a fixed transmission or it is listened back making a one-way discovery between every pair of nodes always possible via their fixed schedules.

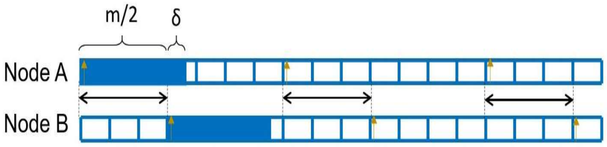

Figure 6: Relative Offset between two nodes

The addition of δ is only for handling the case when both nodes have the exact same offset equals to $m/2 \; mod \; m$. Notice how if we remove it, both nodes would be just missing each other's transmissions. Another possible solution is adding a complete extra slot for listening but as we only need a fraction of a slot big enough as to listen to a single transmission, we proposed the former. We should say that for sufficiently large m , the use of an extra complete slot becomes infinitesimal when compared with the total power usage in a period (T). Finally, the expected power usage per period T is comparable with that of G-Nihao.

The presented schedule focuses on speeding up the one-way discovery. In order to analyze it, we recorded all one-way discovery latencies for all possible encounters between a pair of nodes and compared it with that of G-Nihao. The results are shown in Fig. 7 for a couple of THL2H and a couple of G-Nihao nodes, both running at 10% duty cycle and a COR of $\eta = \frac{1}{10}$. We can see how the worst-case latency of our protocol is bounded by $\frac{T}{2}$ resulting in a lower average one-way discovery latency that settle the basis for the overall improvement we are looking for.

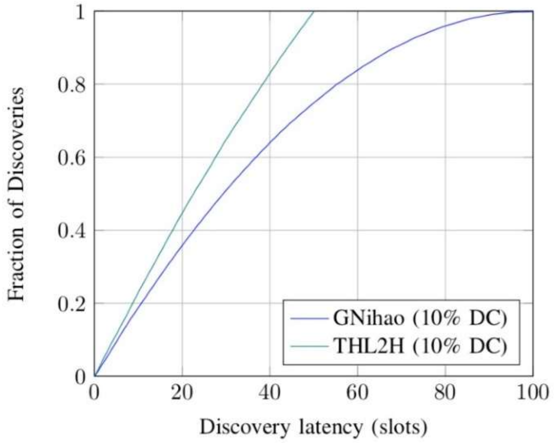

Figure 7: Cumulative latencies of the first contact between two nodes

Below, we present the formal definition of the presented schedule in the Talk-Listen model:

$$\psi L(g,t) = \begin{cases} 1, \text{ if } [t]T < \frac{m}{2} + 1 \text{ or } \frac{T}{2} \leq [t]T < \frac{T}{2} + \frac{m}{2} + 1 \\ 0\text{. otherwise} \end{cases}$$

$$\psi B(g,t) = \begin{cases} 1, \; [\mathrm{t}]T = \mathrm{mi}, \mathrm{i} = 0,1,\ldots,\mathrm{n}-1 \\ 0, \text{ otherwise} \end{cases}$$

Here we use $[t]T$ notation to denote $t \, mod \, T$, which is the slot index in a schedule cycle.

The duty cycle is given by:

$$DC = \frac{m + \alpha n}{mn}$$

However, the previous schedule only guarantees a one-way discovery. Hence, a strategy for achieving two-way discovery is still needed.

4.2.1 Beaconing on-demand

With the fixed schedule presented, there is such a small chance for a pair of nodes to listen to each other, that we will assume the one-way discovery is achieved by only one of them. Hence, this node has to send an extra transmission for the other to listen to it and complete the discovery process. Instead of sending the reply at the first available listening slot, our solution consists in placing it at one of the slots within the middle column (Fig. 8). The only question is, is there always a neighbor listening slot at that column? The answer is yes and the proof is rather simple. If a node listened to a fixed transmission, the transmitter is located with a relative offset smaller or equal to $\frac{m}{2} + \delta$. As fixed transmissions are all located in the first slots of every row, the first listening slot is located at the exact same offset (relative to m) as fixed transmissions are. As the listening slots span for half a row, then the middle slot of that row must be always covered by at least one of them. Finally, the worst-case latency of THL2H is given by the latencies worst-case of the one-way discovery equals to $\frac{T}{2}$ plus the worst-case latency to reply. In this case, the worst-case placement for the reply, would be $\frac{n}{2}$ rows after the first contact, for a total of $\frac{T}{2}$ slots. Hence, the worst-case latency in the symmetric environment is $L = nm$.

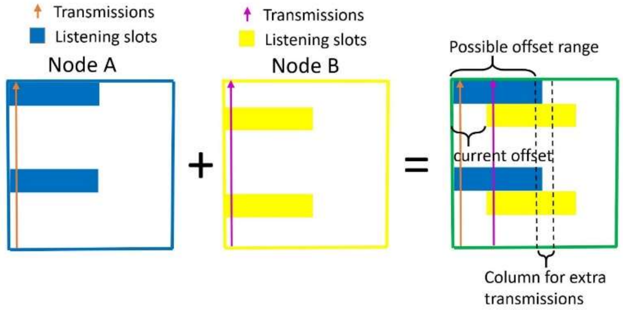

Figure 8: Time intersection between A and B with a given offset

The main advantage of this approach comes in group environments where, all the replies are placed within a column, limiting the possible number of extra transmission per period T to a maximum of n (the number of initial transmissions in the fixed schedule). Therefore, doubling the COR usage in the worst-case. Moreover, as every time a new neighbor makes contact and the protocol schedules the reply, the probability of following neighbors to listen to one of the already placed replies increases. This is due to the fact that every extra transmission covers all possible neighbors with all the $m/2$ different offsets within the same row. As there are n different rows in a period T, the probability of an upcoming neighbor to listen to one of k replies is $\frac{k}{n}$.

The more replies are scheduled, the closer to a G-Nihao schedule with half the initial COR it gets. The transition goes from an initial COR of $\eta = \frac{1}{m}$ to a maximum of $\eta = \frac{1}{(m/2)} = \frac{2}{m}$. Hence, the impact of the maximum COR usage on the duty cycle is very little for small values of α (e.g. $\frac{1}{n}$).

Hence, the duty cycle varies (depending on the COR usage) in the range:

$$\frac{m + \alpha n}{mn} \leq DC \leq \frac{m + 2\alpha n}{mn}$$

4.2.2 Asymmetry

Since heterogeneous devices are likely to have diverse energy budgets, it is reasonable that they operate with different duty-cycles. That is why supporting asymmetric scenarios is a desirable property to have on a protocol. Some protocols only focus on symmetric scenarios (Quorum [13]) while others handle it with restrictions (Searchlight [11]). In the case of G-Nihao, it supports the asymmetric case by simply adjusting the n value according to the required duty-cycle different nodes have. It guarantees discovery with different n_1 and n_2 as long as the m remains constant with a worst-case latency of $m * max\{n1, n2\}$ [7].

Our protocol handles asymmetry in the same way G-Nihao does: keeping m equal on all nodes. No matter the relative offsets between a pair of nodes, it always guarantees the first contact before half a node's period in the worst case. Hence, the reply adds up a number of slots equals to the other node's half. Finally, the worst-case latency of the THL2H for the asymmetric case is:

$$L = m * (max\left\{\frac{n_1}{2}, \frac{n_2}{2}\right\} + min\left\{\frac{n_1}{2}, \frac{n_2}{2}\right\})$$

5. EVALUATION & DISCUSSION

We are mainly interested in the cumulative distribution function (CDF) of discovery latency, duty cycle, and COR usage. In order to avoid tying the results to particular hardware implementation platforms, the number of slots is used as the metric of the latency. The protocol chosen to compare with was G-Nihao as is the only slotted protocol that takes into account COR usage. The simulation parameters are shown in Table 1.

In a real case scenario, the number of neighboring devices changes over time: some leave and new get within contact. So, the perception of how sparse/crowded the network is, comes along with a time variable. In other words, a scenario where the expected number of new (undiscovered) nodes to show up is low. So, we use the average number of new neighbors per slot as, defined as:

$$\beta = \frac{K}{X}$$

where X is the time (in na umber of slots) a node have been running in a given environment and K the number of new neighboring devices it experienced. Notice that this metric is not about the number of neighbors a given node had (in a lifetime). It only focuses on how fast a node experienced the arrival of new neighbors. For example, for a network with $\beta = 0.1$, we can say that any node experiences one new neighbor every 10 slots.

5.1 *Simulation scenarios and platform*

We divide our analysis into two different scenarios:

- The pairwise scenario, where a pair of nodes try to make contact with each other as fast as possible. As usual, nodes are put within a range from the start of the simulation and without any previous information about each other.

- The group scenario, where a number of nodes are randomly inserted in a constraint area and let discover each other for a given amount of time. In this more complex scenario, nodes may enter the area at any time after the start of the simulation, as well as leave it.

Table 1: Simulation Parameters

Parameters	**Value**
No. of Nodes	50 - 100
Deployment Area (m)	100 x 100
Communication Range (m)	20 - 100
Speed (m/s)	1
Time (seg.)	40
Slot size (ms)	10
α / Slot size ratio	0.05
Duty cycle	5%
COR (η)	1/20

5.2 *Symmetric Discovery*

In the pairwise scenario, THL2H presented only sends one extra transmission. Hence, we consider the COR usage equal (ignoring the round error) to that of G-Nihao. The simulations were done on a pair of nodes running at 5% DC and COR of $\eta = 1/20$.

Fig. 9 shows the CDF of discovery. We can see how THL2H is significantly faster than G-Nihao overall. The worst-case latency of THL2H is 390 which is 2.5% faster than the 400 slots of G-Nihao. The result registered for the avg. case latency is considerably better. THL2H presents 143 slots achieving a 28.5% improvement over G-Nihao's 200 slots.

5.2.1 *Group simulations*

Moving on to a scenario with a group of nodes, we keep the nodes at 5% DC and initial COR usage of $\eta = 1/20$. We set a maximum β in order to keep the COR usage below twice the initial one and we obtained $\beta = 0.1$. Then, we ran simulations on several networks starting at $\beta = 0.01$ and increasing it with a step of 0.01 up to $\beta = 0.1$. This gave us networks where nodes faced an average of one new neighbor every 100 slots up to a maximum of one every 10 slots.

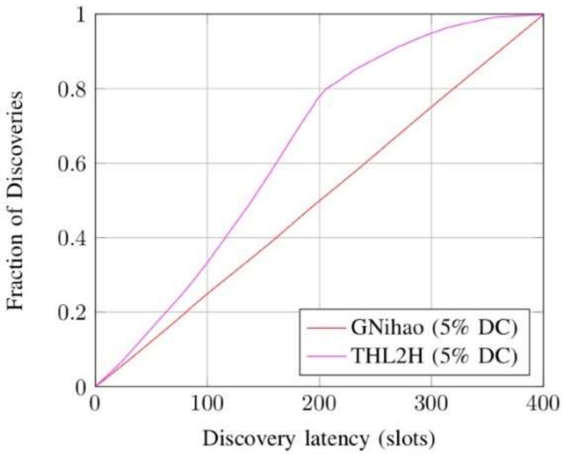

Figure 9: Fraction of Discoveries by latency

For every network, we recorded the actual COR usage and compared it with that calculated as the worst-case. Fig. 10 contains the results for THL2H. The protocol's COR usage was up to a 33% smaller than that of the worst case scenario but most importantly, the simulations clearly show how this improvement gradually increases/decreases as β increases/decreases. A result that is in perfect accordance with our analysis about how unlikely the worst case COR usage becomes the more extra transmissions are sent.

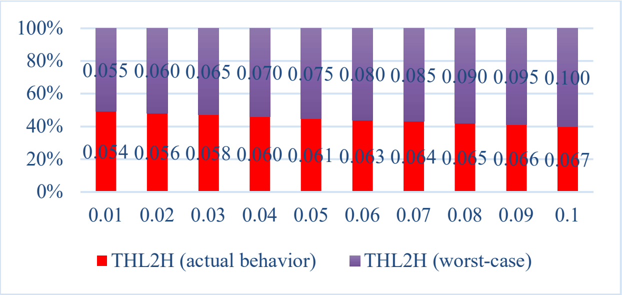

Figure 10: THL2H COR usage worst-case vs simulation

When running simulations on networks with $\beta = 1.5$ (a 50% higher than the previous maximum), the average COR usages obtained were $\eta = \frac{1}{14.14} = 0.070$ for THL2H. The result clearly shows how THL2H performs much better (in terms of COR usage).

5.3 Asymmetric Discovery

The performance on the asymmetric scenario is again compared with that of G-Nihao. The results are shown in Fig. 11 for the CDF of discovery. Simulations were carried out on a pair of nodes running at duty cycles of 1% and 5% with a COR of $\eta = 1/20$. We can see that our protocol is significantly faster than G-Nihao for this scenario as well. The worst-case latency of THL2H is 1190 slots. This value is a 40% improvement over the 2000 slots of G-Nihao. The average case latency recorded was 433, around a 28% improvement over that of G-Nihao with 600.

Finally, we are interested in the COR usage increment needed by G-Nihao in order to match a similar performance of both our protocols. When gradually increasing G-Nihao's COR we found that THL2H worst-case latency of 1190 slots is equivalent to that of G-Niaho with a COR of $\eta = 1/12$, a 40% increment of the COR usage. The average case latency is closer to that of G-Nihao with a COR of $\eta = 1/14$, a 30% increment of the COR usage. Overall, we can say THL2H performed better in this scenario (asymmetric). Result that shows a 50% increment of G-Nihao's COR usage to be able to match its performance.

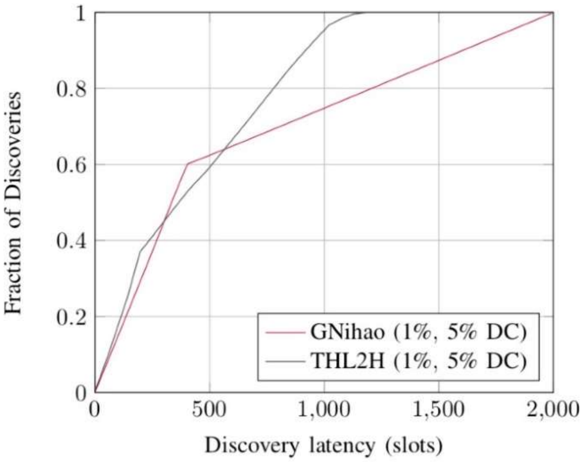

Figure 11: Fraction of Discoveries by latency

6 CONCLUSIONS

In this paper, we have introduced a new protocol for the Talk-Listen model: THL2H. From a pairwise point of view, the proposed protocol uses the mutual assistance approach which basically focuses on achieving one-way discovery to then complete the 2-way discovery by sending one extra transmission.

Our protocol achieves the one-way discovery up to 50% faster than the worst-case scenario of G-Nihao for the symmetric scenario. However, in order to complete the two-way discovery, extra transmissions are sent on-demand. With THL2H, we achieved the same worst-case discovery latency than the G-Nihao protocol but with lower average case latency. The difference in performance we obtained were given by the way the extra transmissions are managed. The THL2H protocol is designed to maintain a COR usage below twice as much the initial one while the second version may increase it deliberately. In the case of the performance of the asymmetric scenario, the results were a significant improvement over that of G-Nihao as THL2H performed better in terms of worst-case and average-case latency. Moreover, we also found that the smaller the ratio of asymmetry the more improvements over G-Nihao's worst case latency.

REFERENCES

[1] A. Nayak and I. Stojmenovic, Wireless Sensor and Actuator Networks, John Wiley & Sons, 2010.

[2] Z. Gu, Y. Wang, Q.S. Hua and F.C.M. Lau. "Neighbor Discovery in Wireless Sensor Networks", In: Rendezvous in Distributed Systems, Springer, 2017.

[3] W. Sun, Z. Yang, X. Zhang and Y. Liu, "Energy-Efficient Neighbor Discovery in Mobile Ad Hoc and Wireless Sensor Networks: A Survey," IEEE Communications Surveys & Tutorials, vol. 16(3), pp. 1448 - 1459, 2014.

[4] B. Sundararaman, U. Buy and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," Ad Hoc Networks, vol. 3, pp. 281–323, 2005.

[5] P. Dutta and D. Culler, "Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications," in 6th ACM Conference on Embedded Network Sensor Systems, 2008.

[6] K. Wang, X. Mao and Y. Liu, "Blind Date: A Neighbor Discovery Protocol," in 42nd Int. Conference on Parallel Processing, 2013.

[7] Q. Ying, L. Shining, X. Xiangsen and L. Zhigang, "Talk More Listen Less: Energy-Efficient Neighbor Discovery in Wireless Sensor Networks," in 35th Annual IEEE Int. Conference on Computer Communications (INFOCOM), 2016.

[8] S. Lai, B. Ravindran and H. Cho, "Heterogenous quorum-based wakeup scheduling in wireless sensor networks," IEEE Trans. Comput., vol. 59, p. 1562 –1575, 2010.

[9] M. J. McGlynn and S. A. Borbash, "Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks," in 2nd ACM Int. Symposium on Mobile Ad Hoc Networking & Computing, 2001.

[10] R. Zheng, J. Hou and L. C. Sha, "Asynchronous Wakeup for Ad Hoc Networks," in 4th ACM Int. Symposium on Mobile Ad Hoc Networking & Computing, 2003.

[11] I. Anderson, Combinatorial designs and tournaments, Oxford University Press, 1998.

[12] M. Bakht, M. Trower and R. Kravets, "Searchlight: Helping Mobile Devices find their Neighbors," in 3rd ACM SOSP Workshop on Networking, Systems, and Applications on Mobile Handhelds, 2011.

[13] T. Yu-Chee, H. Chih-Shun and H. Ten-Yueng, "Power-Saving Protocols for IEEE 802.11-Based Multi-Hop Ad Hoc Networks," in 21st Annual Joint Conference of the IEEE Computer and Communications Societies , 2002.

[14] 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 10th ACM Conference on Embedded Network Sensor Systems, 2012.

[15] L. Chen, Y. Shu, Y. Gu, S. Guo, T. He, F. Zhang and J. Chen, "Group-based Neighbor Discovery in Low-duty-cycle Mobile Sensor Networks," IEEE Trans. on Mobile Computing, vol. 15(8), pp. 1996-2009, 2014.

[16] S. B. Manir, "Collective Neighbour Discovery in Wireless Sensor Network," International Journal of Computer Applications, vol. 131(11), 2015.

[17] Q. Niu, W. Bao and S. Xia, "An Improved Group-Based Neighbor Discovery Algorithm for Mobile Sensor Networks," International Journal of Distributed Sensor Networks, 2014.

[18] L. Wei, W. Sun, H. Chen, P. Yuan, F. Yin, Q. Luo, Y. Chen and L. Chen. "A Fast Neighbor Discovery Algorithm in WSNs", Sensors (Basel), 18(10):3319, Oct. 2018.

[19] L.Wei, B.Zhou, X.Ma, D.Chen, J.Zhang, J.Peng, Q.Luo, L.Sun, D.Li, and L. Chen. "Lightning: A High-efficient Neighbor Discovery Protocol for Low Duty Cycle WSNs". IEEE Communications Letters, vol. 20, no. 5, pp. 966-969, 2016.