ODM: A Neighbor Discovery Protocol Based on Optimal Discovery Model in WSNs

Hao Wang[1], Liangxiong Wei[1], Ping Yuan[2], Xiaodi Li[3], Qian Luo[4], Xiao Luo[4] and Liangyin Chen[1]

[1] School of Computer Science, Sichuan University,
Chengdu, China
[e-mail: chenliangyin@scu.edu.cn]
[2] School of Mathematics and Information Engineerging,
Chongqing University of Education
Chongqing, China
[e-mail: pypingyuan@163.com]
[3] College of Mechanical Engineering,
Donghua University, Shanghai, China
[e-mail: lixiaodi327@gmail.com]
[4] Second Research Institute,General Administration of Civil
Aviation of China, Chengdu, China
[e-mail: caacsri_luoqian@163.com]
*Corresponding author: Liangyin Chen

Abstract

It is a challenging issue to improve the energy efficiency of neighbor discovery in WSNs. This paper proposes an optimal discovery model (ODM) for the first time. Based on the model, we investigate the influence of the relative size of two unequal active slots on the energy efficiency. ODM provides the energy optimal value of the length of the larger active slot at a given duty cycle. Other than existing methods, the worst-case latency bound of ODM is only one period. This is a subversive conclusion, because almost all other related methods are based on a wake-up schedule that contains several periods. We theoretically deduce that ODM can reduce worst-case discovery latency by 43.89% compared to Searchlight-Trim when their duty cycles are the same. The simulations verify the advantage of ODM.

1. Introduction

Nowadays wireless sensor networks (WSNs) are more and more widely used in many fields such as environmental monitoring [1], tactical military [2][3], disaster relief [4] and so on. Besides, WSN is one of key enablers for the Internet of Things (IoT) paradigm. It brings IoT applications richer capabilities for both sensing and actuation [5]. WSN usually includes thousands of sensor nodes which can obtain physical data such as temperature, humidity, vibration data from their surroundings. Then, these data are transferred to sink nodes or base station directly or via multi-hop communication. In most WSN applications, tiny sensor devices are deployed randomly in inaccessible areas. These sensor devices have limited energy budget, and they work continuously until their energy drains. They not only collect sensing data but also send and receive data packets. An ultimate goal of WSN applications is to increase the lifetime of sensor devices through minimizing energy consumption [6].

Discovering neighboring sensors is an essential and continuous task in WSNs. The neighbor discovery is indispensable not only during the deployment phase of sensor networks but also at any time when the network begins to work, since some new sensor devices may be deployed anytime during the working process of the network. Energy efficiency of neighbor discovery is directly related to the lifetime of sensor nodes which are powered by tiny batteries. In addition, there is no synchronous clock for all sensors so that the difficulty of neighbor discovery increases. Energy-efficient neighbor discovery is very challenging and almost all related studies focus on addressing this issue. By the way, smaller energy consumption and discovery latency mean that the method is more energy-efficient [7].

Current neighbor discovery algorithms fall into two categories, i.e. pair-wise algorithms and group-based ones. Group-based method can be regarded as the middleware over pair-wise method. The more energy-efficient the pair-wise protocols used is, the more energy- efficient the neighbor discovery is when the same group-based middleware is employed [8][9][10]. As a result, our main goal in the paper is to improve the energy-efficiency of pair-wise method. Based on whether successful neighbor discovery can be achieved within a certain time (this time is called worst-case latency bound or worst-case bound), there are two categories of pair-wise protocols as well, i.e. the probabilistic algorithms [11][12] and the deterministic ones [13]-[21]. Since many application scenarios require that the worst-case bound is finite [7], we focus on deterministic algorithms in the paper. Owing to restricted energy budget, many deterministic algorithms are slotted-based and each node performs duty-cycled operations. That is to say, time is divide into many slots and each node sleeps during most slots (which are called sleep slots), while turning awake during a few remaining slots (which are called active slots). Each node has a wake-up schedule which is a string of binary numbers where '1' and '0' represent an active slot and a sleep slot respectively. Suppose each wake-up schedule consists of T slots. The wake-up schedule repeats every T slots. The duty cycle (for short DC) is defined as the ratio of the total active slots length within one wake-up schedule to

the time of one wake-up schedule, i.e. T slots. Deterministic methods improve energy-efficiency by studying how many the value of T is and whether each slot in the T slots is active or not. For any two nodes using the same wake-up schedule, deterministic method guarantees that the two nodes can discover each other within one wake-up schedule. The core idea of deterministic methods is to reduce the T and the total length of active slots in T slots while ensuring nodes discover each other within T slots for any two nodes using the same wake-up schedule.

The lengths of all slots in earlier algorithms are the same. Later researchers find there exists redundancy in the overlap of two nodes' active slots, which limits improvement of the energy efficiency. They assign different sizes to active slots and sleep ones to eliminate the redundancy partly or fully. As a result, their studies, including Searchlight-Striped [7] and Non-integer [17] (Non-integer can amend a series of protocols such as Disco [15], U-connect [16], Searchlight [7]. Searchlight-Trim is the best one in Non-integer), obtain a great energy-efficiency improvement. Even so, almost all existing neighbor discovery algorithms, including Disco [15], Searchlight family [7], Block-based algorithms [18] [19], are based on a wake-up schedule that contains several periods. We find this way is still not very energy-efficient.

In this paper, we propose an optimal discovery model (ODM) to comprehensively analyze the performance of the algorithms. Based on the model, we investigate the influence of the relative size of two unequal active slots, i.e. anchor slot (A slot) and probe slot (P slot) on the energy efficiency. We fix the duty cycle (for short DC, which is equivalent to the energy consumption [7]) and compare the discovery latency of almost related methods to measure their energy-efficiency. When a value of DC is given, ODM can give the energy-optimal length of A slot (L_A). Besides, the theoretical analysis provides a subversive conclusion: the worst-case bound of ODM is only one period (the length of the period is decided by the value of DC and L_A), which is very different from the existing studies. To the best of our knowledge, there is no existing method whose wake-up schedule contains only one period. This subversive conclusion may highlight new arenas for future researches about neighbor discovery.

2. Related Works

Neighbor discovery is the key of WSNs applications, because it is the first step of network construction. Besides, neighbor discovery protocol must run continuously in each sensor node until their energy drains, because the network topology may change at any time, i.e. many new sensors may be deployed anytime during the work process of the network. However, there is no synchronous clock for all sensors, which increase the difficulty of neighbor discovery at a limited energy budget. To save energy, always remaining the the radio "on" is not practical because most energy consumption is caused by RF model [7]. In general, each node performs duty-cycled operations for the purpose of energy conservation. That is to say, time is divide into many slots and each node sleeps during most slots (which are called sleep slots), while turning awake during a few remaining slots (which are called active slots). The main goal of neighbor discovery is to determine which slots need to be active to improve energy-efficiency meanwhile successful neighbor discovery is guaranteed.

There are many studies related to neighbor discovery in WSNs. These neighbor discovery algorithms fall into two categories, i.e. pair-wise neighbor discovery methods [11]-[21] and group-based ones [8][9][10]. The pair-wise neighbor discovery protocols only focus on whether any two nodes can find each other. Group-based methods are based on pair-wise ones. The difference is that sensor node can carry neighbor table in its sending packets to recommend the wake-up schedule of discovered neighbors to other nodes and accelerate the process of neighbor discovery. Group-based method can be regarded as the middleware over pair-wise method. The more energy-efficient the pair-wise protocols used, the more energy-efficient the neighbor discovery is when the same group-based middleware is employed. So, our main objective in the paper is to improve the energy-efficiency of pair-wise method.

Based on whether successful neighbor discovery can be achieved within a certain time (this time is called worst-case latency bound or worst-case bound), existing pair-wise protocols fall into two categories as well, i.e. the probabilistic algorithms [11][12] and the deterministic ones [13]-[21]. The probabilistic algorithms includes Birthday [11] and Panda [12]. In birthday protocols, sensor node probabilistically decides the state for each slot from transmitting, receiving and sleep independently. A latest probabilistic protocol is Panda [12]. In Panda, sensors initialize in sleep state to conserve energy. To maximize the discovery rate, Panda also follows a probabilistic approach in which sensors sleep for an exponential duration. Following sleep step, sensors wake up and listen for discovery packets from their neighbors for a fixed duration. If a packet is received, the sensor remains listening until it completes reception of this message. If no transmission is heard in the listen state, the node transmits its discovery message. The authors use the renewal theory to analyze the performance of Panda.

Probabilistic protocols have satisfied mean case performance. However, because of the lack of worst-case latency bound, these probabilistic protocols inevitably incur the problem of long tail. That means that the discovery latency may be arbitrarily long. Since many application scenarios requires that the worst-case bound is finite [7], we focus on deterministic algorithms in the paper. If there is no ambiguity, neighbor discovery refers in particular to deterministic algorithms in the remaining of this paper.

The earliest deterministic neighbor discovery protocols are quorum-based ones [13][14], where time is divided into m^2 contiguous slots. These m^2 slots are arranged as a two-dimensional m × m array and each sensor node can arbitrarily select one row and one column of the slots as active slots. This pattern ensures that no matter which row and column are chosen, one of the active slots of any sensor node can overlap one active slot of the other node within m^2 slots. Another kind of earlier protocols is prime-based deterministic protocols, including Disco [15] and U-connect [16]. In Disco, each node chooses a pair of prime numbers The sensor nodes then awake when the sequence number of one slot can be divide exactly by either of the two prime numbers. If one sensor chooses primes p_1, p_2 and another sensor chooses p_3, p_4, the worst-case latency between these two nodes will be $\min\{(p_1 \cdot p_3), (p_1 \cdot p_4), (p_2 \cdot p_3), (p_2 \cdot p_4)\}$, where $p_1 \neq p_2$, $p_3 \neq p_4$. U-Connect uses only one prime. Each node not only wakes up one slot every p slots, but also wakes up (p+1)/2 slots every p^2 slots. The worst-case latency for U-Connect is p^2. Although quorum-based and prime-based protocols have good worst-case performance, they are worse than the probabilistic protocols, i.e. Birthday, in the average case.

In response to that, Searchlight [7] was proposed. Searchlight employ two active slots, i.e. anchor slot (A slot) and probe slot (P slot) in one period. A slot is fixed in the first slot in each period. Since the length of any period is the same, the offset between the A slot of any node and the A slot of the other node is remain unchanged. Then, P slot is introduced to probe the A slot the other sensor node. P slot is from 2-th to $\left\lfloor \dfrac{t}{2} \right\rfloor$-th slot, where t is the length of one period. P slot is at 2-th slot in first period and move to next slot in next period if the position is not more than $\left\lfloor \dfrac{t}{2} \right\rfloor$. When P slot is at $\left\lfloor \dfrac{t}{2} \right\rfloor$-th slot, it moves to 2-th slot again at next period. The worst-case bound of Searchlight is $t\left\lfloor \dfrac{t}{2} \right\rfloor$. It outperforms previous deterministic protocols much and is the best energy-efficient deterministic one which use same *active* slots.

Researchers afterward find that assigning different sizes to *active* slots and sleep ones brings a great energy-efficiency improvement. Those studies include Searchlight-Striped [7] and Non-integer [17] protocols. Compared with Searchlight, for Striped-Searchlight each active slot overflows by δ, where δ is so small that only one packet can be sent, and it only probes every *even* slot from 1 to $\left\lfloor \dfrac{t}{2} \right\rfloor$. Through this improvement, Searchlight- Striped increases a small amount of DC but reduces the size of wake-up schedule to half [20]. Non-integer trims the active slots to 0.5+δ. This improvement can be applied to quorum-based, prime-based and Searchlight.

Although Striped-Searchlight and Non-integer obtain much promotion in energy efficiency, they still employ equal-size active slots, thus is not very energy-efficiency. Another latest studies is Lightning [20]. Unequal-size active slots is used and the energy- efficiency of Lightning is much better than the methods stated above. The reason why the previous studies employ equal-sized active slots is that the length of active slots is decided by RF module's shift time from sleep state to active state. But some recent studies indicate that some RF modules include another state named as idle state besides sleep and active state. The shift time from idle state to active state is very short. Employing active, sleep as well as idle state makes they employ two kinds of active slots with different active lengths [20].

Although unequal-size active slots is employed in Lightning, it fails to analyze the relative size of the different-size *active* slots. Our study indicates that the relative size of the different-size active slots have much influence on the performance of neighbor discovery protocols. In this paper, we propose an optimal discovery model (ODM) to comprehensively analyze the performance of the algorithms. Based on the model, we investigate the influence of the relative size of two unequal active slots, i.e. A slot and P slot on energy efficiency. Besides, the theoretical analysis draws a subversive conclusion: the worst-case bound of ODM is only one period (the length of the period is decided by the value of DC and L_A), which is very different from the existing studies. This subversive conclusion may highlight new arenas for future researches about neighbor discovery.

There are other related methods, i.e. block-based methods [18][19] and Nihao [21]. In block-based methods, block design concept can be applied to find a neighbor discovery schedule. Nihao defines dedicated listen and transmitting slots. In each listen slot, the radio listens to the channel during the whole slot length. In each transmitting slot, one beacon is sent at the beginning of the slot, and the node goes back to the sleep mode afterwards.

3. Our Design

We design a more general model which is used to investigate the optimal value of L_A and n that make the worst-case bounds minimum. So, the model is called optimal discovery model (ODM). As a prerequisite, the definition of L slot is given below.

Definition1. Logical slot (L slot): Time is divided into contiguous L slots. The length of each L slot is τ. We assume the value of τ is mδ, where δ is a small amount time which is sufficient to send or receive one packet. Note that the concept L slot is used to help us understand that the number of L slots in each period is t whether L_A changes or not.

3.1 A Simple Version

To help understand the essence of the method, a simple version is presented. Each node has

two kinds of active slots in period t, $i.e.$ A slot and P slot, whose lengths are different. There is one A slot in each period whose location is in the first slot. For any two neighboring nodes x and y, let $\varphi(x, y)$ be the phase offset from the A slot of x (denoted by A_x) to the A slot of y (denoted by A_y), as shown in **Fig. 1(a)** and **Fig. 1(b)**. $\varphi(x, y)$ is a random value between one period for different node-pairs because nodes are asynchronous. When $\varphi(x, y)$ is not more than
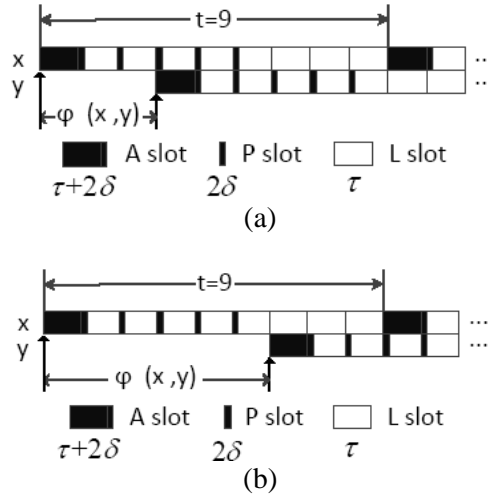
Fig. 1. The offset between A slots of any two neighbors nodes

τ, A_x and A_y will overlap, which is called A-A overlap. For all other offsets, A_x and A_y will never overlap or the overlap length is less than 2δ. So, P slot is employed to search for the A slot of the node's neighbors. To help understand, all P slots can be assigned in one period. We use $M(c)$ to denote one node is in A slot, P slot or sleep state at c-th δ time in the period. The value of c is from 0 to mt - 1, because there are t L slots in one period. $M(c)$ is:

$$M(c) = \begin{cases} A, \ 0 \le c < m+2 \\ P, \ (c\%m = 0 \ or \ c\%m = 1) \ and \ 2m \le c < \left\lfloor \dfrac{t}{2} \right\rfloor m + 2m \\ sleep, \ otherwise \end{cases} \quad (1)$$

where "$\%$" denotes the modulus operator. **Fig. 2** helps you easily follow the expression (1). If $\tau < \varphi(x, y) < \left\lfloor \dfrac{t}{2} \right\rfloor \tau + \tau$, P_x (x's P slot) will overlap A_y in one period (see **Fig. 1(a)**). If $\left\lfloor \dfrac{t}{2} \right\rfloor \tau + \tau < \varphi(x, y) < t\tau$, P_y (y's P slot) will overlap A_x in one period (see **Fig. 1(b)**). Herein, the term "overlap" means the overlap between two nodes' active slots is not less than 2δ. In short, there is at least one A-A or A-P overlap within one period. In Searchlight, the P slot is one L slot. But this arrangement is not energy-efficient. So, we trim each P slot to 2δ and overflow 2δ for each A slot (see **Fig. 2**). Are all P slots located in one period is energy-efficient? We can assign all the $\left\lfloor \dfrac{t}{2} \right\rfloor$ slots to several periods, $i.e.$ n periods (these n

periods is called one hyper-period) and the energy optimal value of n (n_{opt}) can be deduced. Note that, each period always has one A slot no matter how many the value of n is. The essence of our method is: when the condition below is met, it is sure that there is at least one A-A or A-P overlap within n periods. The condition is:

　　1) If the n periods is considered as a matrix M and all rows of the matrix are projected to one row, the range from 2τ to $\left\lfloor \dfrac{t}{2} \right\rfloor \tau + \tau$ in projected row will contain $\left\lfloor \dfrac{t}{2} \right\rfloor$ P slots;

　　2) And the offset between the starts of each two continuous P slots is not less than the length of $L_A - 2\delta$.
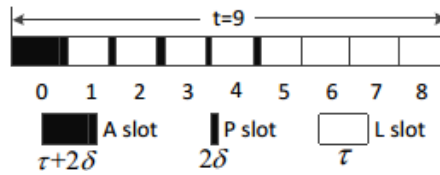
Fig. 2. Each node wakes or sleeps according to same wake-up schedule. This is a simple schedule. Other schedules including the optimal one are presented in the following subsections.

Finally, two questions are raised: 1) Can the energy efficiency be improved by changing L_A at a given DC. 2) Can we obtain n_{opt}? To answer the questions, the optimal version is presented next. Then, the theoretical analysis about n_{opt} and L_A is detailed.

3.2 The Optimal Version

Compared to the simple version above, the difference of the optimal version is that L_A is changeable. Suppose L_A is $(k + 2)\delta$, where k is an integer and no less than 1. There is only one A slot in one row and the location of A slot is at the beginning of each row. When all rows are projected into one row, to ensure the P slots are uniformly distributed in the first half of the projected row except A slot, the two conditions below should be satisfied:

　　1) The offset between the starts of each two continuous P slots are $L_A - 2\delta = k\delta$;

　　2) The number of P slots in the projected row, *i.e.* N, is $\left\lceil \dfrac{\left\lfloor \dfrac{t}{2} \right\rfloor m\delta}{L_A - 2\delta} \right\rceil = \left\lceil \dfrac{\left\lfloor \dfrac{t}{2} \right\rfloor m}{k} \right\rceil$, where $m\delta = \tau$.

The two conditions guarantee successful neighbor discovery within one matrix (or say one hyper-period). For simplicity, we evenly assign all the P slots in one matrix to all n rows. So, there are $\left\lceil \dfrac{\left\lceil \dfrac{t}{2} \right\rceil m}{k} \right\rceil / n$ P slots in one row. Sometimes $\left\lceil \dfrac{\left\lceil \dfrac{t}{2} \right\rceil m}{k} \right\rceil$ can't be divisible by n. The remaining P slot(s) are added to the final period. The general expression of $M\ (c)$ in (1) is:

$$M(c) = \begin{cases} A, \ 0 \le c < k+2 \\ P, \ (c\%k = 0 \ or \ c\%k = 1) \ and \ 2k \le c < \left\lfloor \dfrac{t}{2} \right\rfloor m + 2k \\ sleep, \ otherwise \end{cases} \qquad (2)$$

Fig. 3(a) and **Fig. 3(b)** are two examples. In **Fig. 3(a)**, $n = 2$, $L_A = 12\delta$. In **Fig. 3(b)**, $L_A = 7\delta$. There are 4 and 8 P slots in T respectively. The optimal values of k and n will be obtained in next section.

(a)

(b)
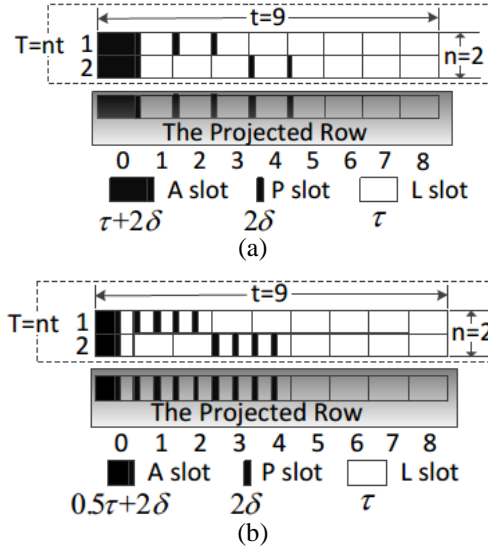
Fig. 3. Two examples of ODM

Lemma 1 There is at least one A-A or A-P overlap whose length is not less than 2δ within one hyper-period (the proof of the lemma is omitted since it is similar to Searchlight and is stated above).

In ODM, A slot employs the same beaconing strategy as Disco [3] to ensure bidirectional discovery in A-A overlap. One beaconing packet is sent both at the beginning and the end of A slot. P slot is required to send one beaconing packet at the first δ and switch to receiving mode in the second δ. When one node receives one beaconing packet, it is required to send one beaconing packet to make its receivers discover it. So, bi-direction discovery in A-P overlap is achieved.

4. Theoretical Analysis

We fix the DC and see the variation of the worst-case bound when the parameters n and k changes for ODM. The minimum value of worst-case bound is deduced in the analysis below. The worst-case bounds of other methods are also given. DC (whose value is d) is defined as the total active slots length within a certain time divided by this total time. In one hyper-period $T = nt$, there are two kinds of active slots, A slot and P slot. As stated in the subsection above, there are n A slots whose length are $(k + 2)\delta$, $\left\lceil \dfrac{\left\lfloor \dfrac{t}{2} \right\rfloor m\delta}{k\delta} \right\rceil$ P slots whose length are 2δ. So, the

DC of ODM is:

$$d = \frac{n(k+2)\delta + \left\lceil \frac{2\left\lfloor \frac{t}{2} \right\rfloor m\delta}{k} \right\rceil}{ntm\delta} \approx \frac{k+2}{mt} + \frac{1}{kn} \tag{3}$$

where $d > \dfrac{1}{kn}$. Then, $t = \dfrac{k+2}{m(d - \frac{1}{kn})}$. The worst-case bound

$$w(n,k) = nt = \frac{n(k+2)}{m(d - \frac{1}{kn})} \tag{4}$$

We take partial derivative of $w(n, k)$ with respect to n. When $\dfrac{\partial w}{\partial n} = 0$, we find the

worst-case bound obtains the minimum value. The corresponding n_{opt} is $\dfrac{2}{dk}$. When n_{opt} is

taken into equation (4), $w(k) = \dfrac{4}{md^2}(1 + \dfrac{2}{k})$.

(a)

(b)
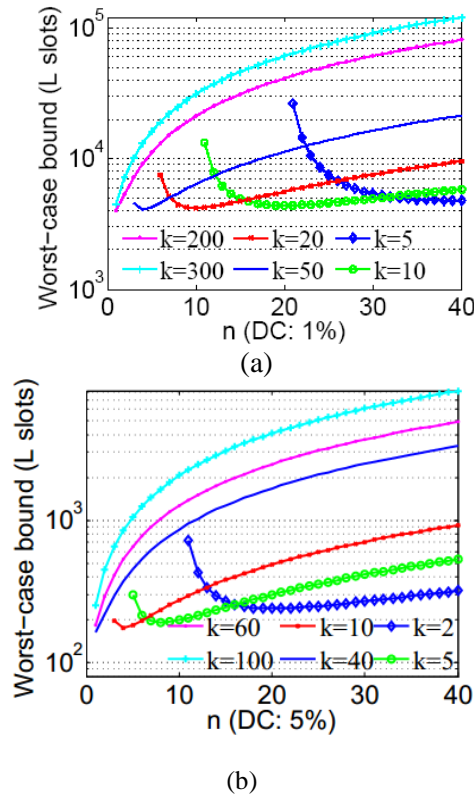
Fig. 4. Worst-case bounds in theory

Then, the optimal value of k is $2/d$. It is because $n = 2/(kd) \geq 1$. So, k must be not less than $2/d$. When $k = 2/d$, n_{opt} is 1. Accordingly, the minimum value of $w(n, k)$ is $\dfrac{4}{md^2}(1+d)$. The curves of worst-case bounds are given by **Fig. 4(a)** and **Fig. 4(b)** when the DC is fixed to 1% and 5% respectively. We can see n_{opt} is always 1. When d, n and k are known, t can be calculated. Then, according to expression (2), the locations of A slot and all P slots in one period is obtained. For comparison, the theoretical worst-case bounds for different protocols at same DC are given by **Table 1**. Since Searchlight-Trim is the best protocol in Non-integer [17], we only care Searchlight-Trim. We can see our method reduce the worst-case bound up to 43.89% in theory compared with Searchlight-Trim.

Table 1. Worst-case bounds for different protocols at same DC

Protocol	Worst bounds	Worst bounds denoted by d
Disco	t^2	$\dfrac{4}{d^2}$
Searchlight-Striped	$\left\lceil \dfrac{\left\lfloor \dfrac{t}{2} \right\rfloor}{2} \right\rceil t \approx \dfrac{t^2}{4}$	$\dfrac{(1+\delta)^2}{d^2}$
Searchlight-Trim	$\left\lfloor \dfrac{t}{2} \right\rfloor t \approx \dfrac{t^2}{2}$	$\dfrac{2(0.5+\delta)^2}{d^2}$
ODM	$nt,\, n_{opt} = 1$	$\dfrac{4}{md^2}(1+d)$

5. Evaluation

The primary goal of our evaluations is to show that whether our algorithm performs better than the other algorithms. The evaluation environment is described as follows: there is a 200m × 200m square region which is divided into 40 × 40 same size grids. We put 200 nodes on the vertexes of the grids randomly. The communication range of any two nodes is within 50m to 100m at random. Firstly, in order to evaluate the worst-case bounds in the static case, we keep all nodes static and evaluate how long discovery neighbor will take with different protocols. Secondly, for evaluating the performance of all protocols in dynamic environment, we let nodes move along the grid side according to a certain speed in the process of the simulation. Their moving directions are reset randomly when they move to the vertexes of the grids but remain unchanged at other time. The start time of each node for all simulations is randomized within the first period.

When the worst-case bounds are evaluated, we log all times of every node discovering its neighbors. If all nodes discover all their neighbors, we end the simulation. We draw the cumulative distribution function (CDF) of discovery latency for all protocols operating at a given DC, such as 2%. **Fig. 5** indicates that ODM achieves smaller latency all along, compared with the others. In particular, ODM reduces the time of discovery latency more than 43.64% compared with Searchlight-Trim.
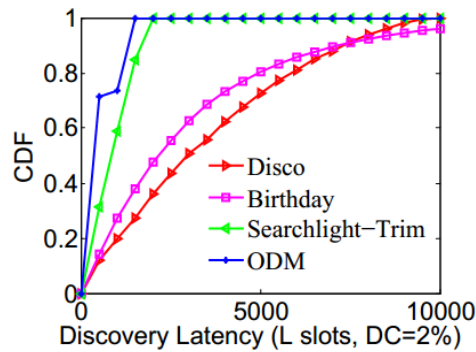
Fig. 5. CDF

We employ average discovery latency(ADL) to evaluate all algorithms in dynamic case. **Fig. 6(a)** indicates that the ADLs all go smaller and smaller for these four algorithms with the value of DCs changing from 1% to %5. When the DC is 1%, ODM lowers ADL approximately 48.29% compared to Searchlight-Trim. **Fig. 6(b)** shows the impact of speed on ADL. It indicates that ODM is so robust that nearly avoiding the influence of speed.
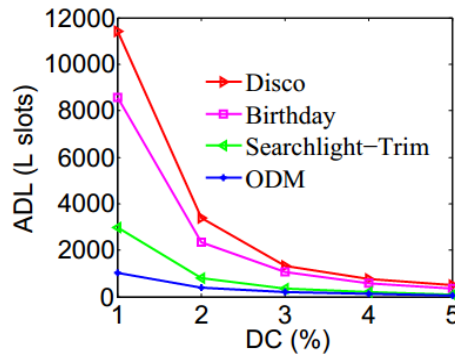
(a)
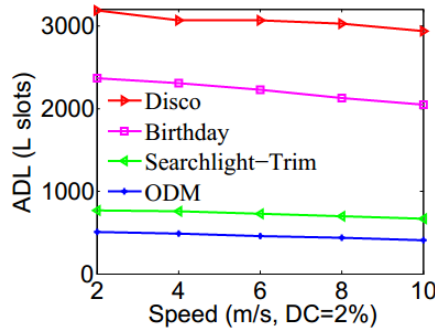
(b)

Fig. 6. The curves of ADL in dynamic scenarios

6. Conclusion

With the challenge that the energy of nodes is usually restricted, the problem of asynchronous neighbors discovery about how to make energy more efficient is a primary concern for WSNs applications. In our research, aiming at energy saving, ODM is proposed. We theoretically deduce that ODM can reduce worst-case bound by approximately 43.89% in comparison to Searchlight-Trim. The simulation results verified the advantages of ODM.

Acknowledgement

References

[1] I. Bisio and M. Marchese, "Efficient Satellite-Based Sensor Networks for Information Retrieval," *IEEE Systems Journal*, vol. 2, no. 4, pp. 464-475, Dec. 2008. Article (CrossRef Link).

[2] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications: Remote large-scale environments," in *Proc. of IEEE Military Communications Conference MILCOM*, pp. 1-7, Oct. 2009. Article (CrossRef Link).

[3] K. C. Go, J. H. Kim, J. R Cha, B. G. Jo and K. K. Kim, "Challenges and Solutions for Routing in Converged Satellite and Terrestrial Networks," in *Proc. of IEEE Military Communications Conference MILCOM*, pp. 1838-1843, Nov. 2013. Article (CrossRef Link).

[4] B. McCarthy, S. Varakliotis, C. Edwards, and U. Roedig, "Deploying wireless sensor networking technology in a rescue team context," in *Proc. of RealWorld Wireless Sensor Networks - 4th International Workshop*, pp. 37-48, Jan. 2010. Article (CrossRef Link).

[5] M. T. Lazarescu, "Design of a WSN Platform for Long-Term Environmental Monitoring for IoT Applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45-54, Mar. 2013. Article (CrossRef Link).

[6] W. Lee, S. Choi, N. Kim, J.H. Youn, and D. Moore, "Block Combination Selection Scheme for Neighbor Discovery Protocol," in *Proc. of Fifth International Conference on Communication Systems and Network Technologies*, pp. 143-147, 2015. Article (CrossRef Link) .

[7] M. Bakht, M. Trower, and R. Kravets, "Searchlight: Won't you be my neighbor?'," in *Proc. of Mobicom'12 Proceedings of the 18th annual international conference on Mobile computing and networking*, 185-196, 2012. Article (CrossRef Link).

[8] D. Zhang, T. He, Y. Liu, et al, "Acc:generic on-demand accelerations for neighbor discovery in mobile applications[C]," in *Proc. of ACM Conference on Embedded Network Sensor Systems*, ACM, 169-182, 2012. Article (CrossRef Link).

[9] 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[J]," *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, pp. 1996-2009, 2016. Article (CrossRef Link).

[10] D. Zhang, T. He, F. Ye, R. K. Ganti, and H. Lei, "EQS: Neighbor Discovery and Rendezvous Maintenance with Extended Quorum System for Mobile Sensing Applications[C]," in *Proc. of IEEE International Conference on Distributed Computing Systems*, pp. 72-81, 2012. Article (CrossRef Link).

[11] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. of ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc'01)*, 2001, pp. 137–145. Article (CrossRef Link).

[12] R. Margolies, G. Grebla, T. Chen, D. Rubenstein, and G. Zussman, "Panda: Neighbor Discovery on a Power Harvesting Budget[J]," *IEEE Journal on Selected Areas in Communications*, 2016. Article (CrossRef Link).

[13] S. Lai, B. Ravindran, and H. Cho, "Heterogenous quorum-based wake-up scheduling in wireless sensor networks," *IEEE Transactions on Computers*, vol. 99, no. PrePrints, 2010. Article (CrossRef Link).

[14] 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*, 2002. Article (CrossRef Link).

[15] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. of the SenSys 2008*, pp. 71-84, 2008. Article (CrossRef Link).

[16] A. Kandhalu, K. Lakshmanan and R. Rajkumar, "U-connect: A lowlatency energy-efficient asynchronous neighbor discovery protocol," in *Proc. of Int. Conf. Inf. Process. Sens. Netw. (IPSN'10)*, pp. 350–361, 2010. Article (CrossRef Link).

[17] S. Chen, A. Russell, R. Jin, Y. Qin, B. Wang, and S. Vasudevan, "Asynchronous neighbor discovery on duty-cycled mobile devices: Integer and non-integer schedules," in *Proc. of MobiHoc'15 Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 47-56, 2015. Article (CrossRef Link).

[18] W. Lee, I. H. Youn, T. Song, N. Kim, and J. H. Youn, "Prime Block Design for Asynchronous Wake-Up Schedules in Wireless Sensor Networks," *IEEE Communications Letters*, pp. 1437-1440, 2016. Article (CrossRef Link).

[19] R. Zheng, J. C. Hou and L. Sha, "Optimal Block Design for Asynchronous Wake-Up Schedules and Its Applications in Multihop Wireless Networks[J]," *IEEE Transactions on Mobile Computing*, vol. 5, no. 9, pp. 1228-1241, 2006. Article (CrossRef Link).

[20] 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[J]," *IEEE Communications Letters*, vol. 20, no. 5, 2016. Article (CrossRef Link).

[21] Y. Qiu, S. Li, X. Xu, and Z. Li, "Talk more listen less: Energy-efficient neighbor discovery in wireless sensor networks," in *Proc. of IEEE Conference on Computer Communications (INFOCOM)*, 2016. Article (CrossRef Link).

Hao Wang received the Bachelor's degree in Computer Science from Sichuan University, Chendu, China in 2015. And he is currently working towards Graduate at School of Computer in Sichuan University, Chengdu, China. His research focus on the Indoor Locationing and Wireless Sensor Network.

Liangxiong Wei is working for his Ph.D in School of Computer, Sichuan University, Chengdu, China. His research insterests include Wireless Sensor Network, Embedded Systems, Computer Network.

Ping Yuan received the Bachelor's degree in 1991, in software engineering, Southwest Normal University, Chongqing, China. She is an associate professor, Chongqing University of Education Chongqing, China. Her research field is computer network application.

Xiaodi Li expected to receive the bachelor's degree of Mechatronics Engineering from Donhua University, Shanghai, China in June 2018. Her research interests include Machine Learning, Natural Language Processing and Artificial Intelligence.

Qian Luo received the PhD degree in Sichuan University, Chengdu, China. He is currently working in the Second Research Institute of Civil Aviation Administration of China (CAAC). He serves as an expert in the field of information technology and has accomplished several key national projects on aviation area. He is a researcher and has published many academic papers.

Xiao Luo is a committee member of the eleventh all-China Youth Federation and a standing director of China Association for Science and Technology. He is currently working in the Second Research Institute of Civil Aviation Administration of China (CAAC) and serves as a director. His research interests include electronic information of civil aviation airport, logistics and aviation security. As a technical leader and manager of aviation, he has won many national honors and published many academic papers.
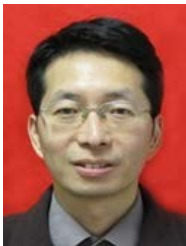
Liangyin Chen received the PhD degree in the School of Computer Science from Sichuan University in 2008. He is currently a professor in the School of Computer Science at Sichuan University. He was a visiting researcher at the University of Minnesota from August 2009 to September 2010 under the guidance of Professor Tian He. He is the author and co-author of more than 30 papers, many of which were published in premier network journals and conferences. He received the award of SCU-Lixin Tang distinguished teacher in 2012. His research interests include wireless sensor network, embedded systems, computer network, distributed systems, big data analytics, industrial Internet and IoT.