Panacea: A Low-Latency, Energy-Efficient Neighbor Discovery Protocol for Wireless Sensor Networks

Zhen Cao, Zhaoquan Gu, Yuexuan Wang, Heming Cui
Department of Computer Science, The University of Hong Kong

***Abstract*—Neighbor discovery is essential to construct wireless sensor networks. Most existing studies on neighbor discovery are based on the assumption that either all nodes are within the radio range of each other, or only two nodes compose the network. However, networks are partially-connected in reality: some nodes are within the radio range of each other, while others are not. Low latency and energy efficiency are two common goals, which become even more challenging to achieve at the same time in partially-connected networks. We find the collision caused by simultaneous transmissions is the main obstruction of achieving the two goals. In this paper, by alleviating collisions, we present *Panacea*, the first low-latency, energy-efficient protocol for a partially-connected network. To begin with, we design *Panacea*-WCD for nodes that have collision detection mechanisms and *Panacea*-NCD for nodes that do not have. Then, we prove the discovery latency of *Panacea* is bounded by a low latency $O(nlnn)$, where each node has n neighbors on average. Finally, we implement *Panacea* for evaluations in complicated partially-connected networks. The evaluated results corroborate our analyses: compared with two notable protocols, i.e. PND and Coupon, *Panacea* achieves up to 5.14 times lower latency with the same energy consumption; and saves 30% energy for similar latency with Coupon.**

I. Introduction

Wireless sensor networks have been drawing great attention because of their wide range of applications such as environment observations [1], health-care [2], social networks [3], agriculture monitoring [4], and etc. In general, a large number of sensor nodes are deployed in a distributed manner without any pre-defined network infrastructure, and the nodes are required to construct a network for communication. *Neighbor discovery* [16] is the process of constructing one-hop neighboring connectivity. As an essential step in configuring wireless sensor networks, neighbor discovery has been widely studied in past few years [5]–[15].

Unfortunately, to the best of our knowledge, no existing neighbor discovery algorithms achieves both low latency and energy efficiency in partially-connected networks. First of all, for battery-powered sensors, lifetime prolongation and discovery latency reduction should be jointly considered. Achieving low latency and energy efficiency are two major goals. Second, a wireless sensor network consists of multiple nodes and these nodes are partially-connected, i.e. some nodes are within the radio range of each other while others are not. It becomes more demanding to achieve low latency and energy efficiency for this partially-connected network.

By experiment, we find collisions caused by simultaneous transmissions result in the waste of time and energy. We considered a partially-connected network with 1000 nodes and probability 0.5 for two nodes to be neighbors. Collisions of Hello [8], a deterministic protocol designed for two nodes, happened as frequently as 99.8% of the time; collisions of PND [14], a probabilistic protocol, happened as frequently as 46.5% of the time. That means, because of collisions, existing works waste time and energy, and cannot achieve low latency and energy efficiency for partially-connected networks.

To save energy, sensor nodes switch radio off for most of the time, and on if necessary. The fraction of time a node turns its radio on is denoted as *duty cycle*. Some existing algorithms transmit and listen with certain probabilities that are related to duty cycle [12], [13], [15], but assume the network is fully-connected, i.e. any two nodes are neighbors, which is unrealistic. Some works focus on minimizing *discovery latency* [12], [13], which is defined as the time to discover all neighbors. However, nodes keep radios on all the time (duty cycle is 1) and run out of energy quickly. The protocol in [14] considers both duty cycle and discovery latency, but it cannot guarantee bounded latency. Many deterministic protocols minimize discovery latency with a given duty cycle in [5]–[11]. Nevertheless, these methods only aim at two nodes, and ignore collisions caused by simultaneous transmissions among multiple nodes. In a word, no low-latency, energy-efficient neighbor discovery algorithms exists for a partially-connected network.

In this paper, we propose *Panacea*, the first low-latency, energy-efficient neighbor discovery protocol for partially-connected networks. The contributions are highlighted as follows:

- We propose two algorithms based on whether sensors' hardware mechanisms can identify collisions: *Panacea*-NCD for no collision detection while *Panacea*-WCD for collision detection;
- *Panacea* is suitable for a partially-connected network, not limited to fully-connected networks or two-node scenario;
- *Panacea* achieves low bounded latency and energy efficiency by alleviating collisions;
- *Panacea* is suitable for both synchronous (i.e. all nodes start at the same time) and asynchronous (i.e. all nodes

start at different time slots) networks.

Panacea improves both latency and energy efficiency. On one hand, *Panacea* is bounded by low latency $O(nlnn)$ in both synchronous and asynchronous scenarios, where n is the average number of neighbors. In the simulation, *Panacea*'s discovery latency is up to 5.14 times lower, compared with Coupon [13] and PND [14]. On the other hand, the evaluated results show that *Panacea* saves 30% energy when it achieves similar latency with Coupon [13].

The rest of this paper is organized as follows. We review existing approaches to neighbor discovery in Section II, and introduce the preliminaries in Section III. Then, we propose *Panacea*-NCD and *Panacea*-WCD in Section IV. We analyze their performance in synchronous and asynchronous scenario in Section V and VI respectively. We implement *Panacea*, and illustrate the comparison outcomes in Section VII. Finally, we conclude this paper in Section VIII.

II. Related Work

Up to now, various neighbor discovery algorithms have been proposed in wireless sensor networks. These algorithms mainly fall into two categories, probabilistic and deterministic.

In probabilistic algorithms, nodes transmit, listen or sleep with a certain probability. In [12], Birthday protocols maximized the expected discovery rate with probability theory methods; it was further studied in [13], whose discovery latency was bounded. However, they ignored the significance of saving energy. Following that, similar and nicer probabilistic algorithms with a pre-defined duty cycle were proposed in [14], [15], but they only considered unrealistic fully-connected networks.

In deterministic algorithms, according to a survey [16], three methods were adopted to guarantee discovery between two nodes. First, *over-half occupation* takes advantage of overlapping slots if each node turns on its radio for more than half of a period. It was adopted in many algorithms, such as SearchLight [11]. Second, *quorum system* ensures discovery with intersected quorums in [5], [7], [10]. Among these algorithms, only Hedis [10] supports asymmetric duty cycle. Third, based on Chinese Remainder Theorem [17], *co-prime* guarantees intersection of slots in Disco [6], U-Connect [7], and Todis [10]. However, most of these deterministic algorithms simply considered two-node scenario, and failed to address collisions among multiple nodes.

To the best of our knowledge, no existing work considers both latency and energy consumption for partially-connected networks. Hence, it is important and necessary to design a low-latency, energy-efficient neighbor discovery protocol for partially-connected networks.

III. Preliminaries

In this section, we introduce our system model and assumptions for neighbor discovery.

A. Sensor Node Model

Assuming that there are N sensor nodes in total and each node u_i has a unique identifier (ID) i. Time is assumed to be divided into slots of equal length t_0, which is sufficient to finish communications. All nodes can communicate through a fixed channel by exchanging messages. In each time slot, node u_i chooses one of the three states $\{Transmit, Listen, Sleep\}$, where $Transmit$ means u_i broadcasts (sends a packets) on the channel; $Listen$ means u_i receives packets from neighbors; $Sleep$ means u_i turns its radio off and does nothing.

For simplicity, we assume state transitions do not consume any time or energy, and only nodes that are transmitting or listening consume their battery power. A node's *duty cycle* is defined as the fraction of time that the node is transmitting or listening, and a lower duty cycle implies a longer lifetime. In our model, we assume each node u_i has a pre-defined duty cycle θ_i. Besides, we assume the start time of node u_i is t_i^s, and u_i sleeps until t_i^s.

B. Network Model

Most works [12], [13] only consider cliques, where any two nodes are neighbors, but this assumption is unrealistic. We consider a partially-connected network and use *neighboring matrix* $M_{N\times N}$ to represent the neighboring relations. If u_i is a neighbor of u_j, we set M_{ij}, M_{ji} to be 1 (we assume the neighboring relation is undirected and u_j is also a neighbor of u_i), else the value of the entrance is 0.

For simplicity, to initialize $M_{N\times N}$, we assume node u_i is a neighbor of node u_j with probability p_n. The assumption makes the model more realistic than fully-connected networks and it could simplify the analysis of neighbor discovery protocols. Let $n = p_nN$, which denotes the average number of neighbors. We assume n is large for complex partially-connected networks.

C. Collision Model

When two or more neighbors of node u_i transmit in the same time slot, collisions occur on the channel and u_i cannot decode the message correctly. Actually, node u_i can discover a neighbor u_j successfully if and only if u_j is the only transmitting neighbor of u_i. In some cases, collisions can be detected by hardware mechanisms [13], [14]. A listening node can distinguish whether collisions occur or no neighbors is transmitting, apart from successful discovery. This collision detection (CD) mechanism enables the listening node to notify its transmitting neighbors of the transmission outcomes, and hence they take measures to reduce the collisions.

Neighbor discovery is not bidirectional, which means u_i discovering u_j is not equal to u_j discovering u_i. Let discovery latency $L(i,j)$ be the duration from when u_i starts to when u_i discovers u_j:

$$L(i,j) = T - t_i^s. \tag{1}$$

TABLE I
NOTATIONS FOR NEIGHBOR DISCOVERY

Notation	Description
u_i	Sensor node u_i with ID i
t_i^s	Sensor node u_i starts neighbor discovery at time t_i^s
p_n	A node is the neighbor of another with probability p_n
N	The number of nodes in the network is N
n	The average number of neighbors is n, $n = p_n N$
t_0	The length of a time slot is t_0
$L(i,j)$	The discovery latency that node u_i discovers node u_j
$L(i)$	The discovery latency that node u_i discovers all neighbors
p_i^t	Node u_i transmits in a time slot with probability p_i^t
p_i^l	Node u_i listens in a time slot with probability p_i^l
p_i^s	Node u_i sleeps in a time slot with probability p_i^s
θ	The pre-defined duty cycle is θ
W	The time slot spent by a node discovering all neighbors is W
M	Neighboring matrix, $M_{ij} = 1$ means u_i and u_j are neighbors

Then, *discovery latency* of node u_i (denoted as $L(i)$) is the time to discover all neighbors:

$$L(i) = \max_{M_{ij}=1} L(i,j). \tag{2}$$

The sensor nodes could start at the same or different time slots, which are called *synchronous* and *asynchronous* respectively. The notations are listed in Table I.

In this paper, we study low-latency, energy-efficient neighbor discovery to reduce collisions for both synchronous and asynchronous networks, with and without a collision detection mechanism.

IV. PANACEA: ENERGY-EFFICIENT NEIGHBOR DISCOVERY PROTOCOL

In this section, we describe *Panacea*, a low-latency, energy-efficient neighbor discovery protocol. To begin with, we propose *Panacea*-NCD for the scenario that the sensor nodes have no collision detection mechanism. Then, we present *Panacea*-WCD if the nodes have collision detection mechanisms.

A. Panacea-NCD: No Collision Detection

Panacea-NCD is a randomized neighbor discovery algorithm without collision detection. We describe the algorithm for node u_i as follows:

- In time slot t, u_i transmits with probability p_i^t, listens with probability p_i^l, and sleeps with probability p_i^s;
- If u_i chooses state *Transmit*, it transmits a message containing its source ID on the channel;
- If u_i chooses state *Listen*, it listens on the channel and decodes the source ID of received message if it receives a message successfully;
- If u_i chooses state *Sleep*, it does nothing.

It is obvious that $p_i^t + p_i^l + p_i^s = 1$. Suppose the duty cycle of node u_i is θ_i, which denotes the fraction of time slots that u_i is transmitting or listening, hence

$$\theta_i = p_i^t + p_i^l. \tag{3}$$

Considering nodes are comparatively well-distributed in the network, we suppose the probabilities of each node in each state are the same for simplicity. That is, $\forall i \in [1,N], p_i^t = p_t$, $p_i^l = p_l$, $p_i^s = p_s$, and $\theta_i = \theta$. To minimize the probability of collisions' occurrence, we derive an approximation

$$p_t = \frac{1}{n}. \tag{4}$$

This transmission probability effectively alleviates collisions and helps to achieve low latency. We show how to derive this value in Section V.

B. Panacea-WCD: With Collision Detection

If nodes could identify collisions, we present *Panacea*-WCD, a novel random algorithm that achieves low latency for a given duty cycle. With CD, the transmitting node(s) could be notified whether transmissions are successful by their listening neighbors. We describe *Panacea*-WCD as follows:

Each time slot is divided into two sub-slots. Nodes execute transmission or reception in the first sub-slot, and notify and maintain *DiscoveredList* in the second sub-slot. Initially, each node u_i sets $k_i = 0$, and let α be a constant.

1) In the first sub-slot, node u_i transmits a message containing its source node ID with probability $p_i^t = \frac{1}{n+\alpha k}$, listens with probability $p_i^l = \theta - p_i^t$, and sleeps with probability $p_i^s = 1 - \theta$;
2) In the second sub-slot, if
   - u_i is in $Listen$ state in the first sub-slot: if u_i receives a message successfully, u_i decodes and records the source ID in the message. If the ID does not belong to u_i's *DiscoveredList*, u_i adds the ID to *DiscoveredList*, and deterministically transmits a message on the channel (a bit is OK) in the second sub-slot. Else, it does nothing.
   - u_i is in $Transmit$ state in the first sub-slot: if u_i detects energy (a message or a collision by multiple messages), u_i is notified the successful transmission, and sets $k := k+1$. Else, u_i regard its transmission to be unsuccessful in the first sub-slot.
   - u_i is in $Sleep$ state in the first sub-slot: u_i does nothing.

The core of *Panacea*-WCD is that once u_i discovers a new node u_j out of its *DiscoveredList*, u_i adds u_j's ID to its *DiscoveredList* and notifies u_j the successful discovery. In this successful scenario, u_i's feedback in the second sub-slot can be 1-bit. Thus, the second sub-slot is much smaller than the first one, and only introduces a small overhead.

V. ANALYSIS OF SYNCHRONOUS SCENARIO

In this section, we analyze the performance of both *Panacea*-NCD and *Panacea*-WCD for synchronous scenario, where all nodes start neighbor discovery at the same time. We derive the discovery latency is inversely proportional to the duty cycle.

A. Analysis of Panacea-NCD

1) Latency Analysis: According to *Panacea*-NCD, the probability that node u_i discovers a neighbor successfully in a given slot is:

$$p_{suc} = p_t(1-p_t)^{n-1}(\theta - p_t). \quad (5)$$

By maximizing p_{suc}, we alleviate collisions at utmost. Thus, we derive the transmission probability with the derivative

$$p_t = \frac{\theta n + 2 - \sqrt{4 + (\theta n)^2 - 4\theta}}{2(1+n)} \approx \frac{1}{n}.$$

Substituting into the formulation, the probability that a node discovers a neighbor successfully in a given slot is

$$p_{suc} \approx \frac{\theta}{en}. \quad (6)$$

We next show maximizing p_{suc} helps to achieve the lowest expected latency. Let W be a random variable that denotes the time a node spends discovering all neighbors. Considering any node, we define the time spent in discovering a new neighbor after it discovered $j-1$ neighbors to be W_j, which follows Geometric distribution with parameter $p_{suc}(j)$: $p_{suc}(j) = (n-j+1)p_{suc}$. Hence, the expectation of W_j is computed as:

$$E[W_j] = \frac{1}{p_{suc}(j)} = \frac{1}{(n-j+1)p_{suc}}.$$

As $W = W_1 + W_2 + ... + W_n$, the expectation of W is

$$E[W] = \sum_{j=1}^{n} E[W_j] = \frac{1}{p_{suc}} H_n$$

where H_n is the n-th Harmonic number, i.e., $H_n = lnn + \Theta(1)$. By maximizing p_{suc}, the lowest expected discovery latency becomes:

$$E[W] \approx \frac{ne}{\theta}(lnn + \Theta(1)) = \Theta(nlnn). \quad (7)$$

2) Upper Bound of Latency: We show the discovery latency is not likely to be much larger than its expectation.

If W_i is given, the value of W_j will not be affected for $i < j$. That is, for $i \neq j$, W_i and W_j are independent and they satisfy $P(W_j = w_j | W_i = w_i) = P(W_j = w_j)$. Since W_j follows Geometric distribution, and $Var[W_j] = \frac{1-p_j}{p_j^2}$, the variance of W is

$$Var[W] = \sum_{j=1}^{n} Var[W_j] \leq \frac{\pi^2}{6p_{suc}^2} - \frac{H_n}{p_{suc}}.$$

With *Chebyshev's inequality*, the probability that the discovery time is 2 times larger than the expectation is

$$P[W \geq 2E[W]] \leq \frac{Var[W]}{E[W]^2} \leq \frac{\pi^2}{6H_n^2} - \frac{p_{suc}}{H_n}.$$

For large n, $P[W \geq 2E[W]]$ is close to 0. That is, the time for a node to find all neighbors is very likely to be smaller than 2 times of expected latency. Therefore,

$$W = O(nlnn). \quad (8)$$

B. Analysis of Panacea-WCD

1) Latency Analysis: In this part, we analyze the discovery latency of *Panacea*-WCD. Similar to *Panacea*-NCD, we suppose the latency to find a new neighbor after discovered $j-1$ neighbors is W_j.

To simplify this model, we consider a node u_i discovering neighbors at an average level, where the number of discovered neighbors follows normal distribution. That is, some nodes that discover neighbors faster and slower tend to compensate the number of discovered neighbors of each other. In other words, we have a global transmission probability $p_t(j)$ in W_j, which is

$$p_t(j) = \frac{1}{n + \alpha(j-1)}. \quad (9)$$

In a given time slot, the probability that u_i discovers a new neighbor after it discovered $j-1$ neighbors is

$$p_{suc}(j) = p_t(j)(1-p_t(j))^{n-1}(\theta - p_t(j))(n-j+1). \quad (10)$$

As W_j follows Geometric distribution with parameter $p_{suc}(j)$, for α close to 1, $\frac{1}{n+\alpha(j-1)} \ll \theta$, the latency expectation that u_i discovers all neighbors within $W = \sum_{i=1}^{n} W_i$ is:

$$E[W] = \sum_{j=1}^{n} E[W_j] = \sum_{j=1}^{n} \frac{1}{p_{suc}(j)}.$$

Because

$$\sum_{j=1}^{n} \frac{n + \alpha(j-1)}{n-j+1} = (\alpha+1)nH_n - \alpha n$$

where H_n is n-th *harmonic number*, we can obtain:

$$\frac{n}{\theta}[(\alpha+1)lnn + \Theta(1)] \leq E[W] \leq \frac{en}{\theta}[(\alpha+1)lnn + \Theta(1)].$$

Therefore, we derive

$$E[W] = \Theta(nlnn). \quad (11)$$

2) Upper Bound of Latency: We show that the latency is not likely to be much larger than its expectation.

For $i \neq j$, W_i and W_j are independent by definition. As $Var[W_j] = \frac{1-p_{suc}(j)}{p_{suc}^2(j)}$ for Geometric distribution with parameter $p_{suc}(j)$, we obtain:

$$Var[W] = \sum_{j=1}^{n} Var[W_j] = \sum_{j=1}^{n} \frac{1}{p_{suc}^2(j)} - \sum_{j=1}^{n} \frac{1}{p_{suc}(j)}.$$

We know that

$$\sum_{j=1}^{n} \frac{1}{p_{suc}^2(j)} \leq \frac{e^2}{\theta^2}[\alpha^2 n - 2\alpha n(1+\alpha)H_n + \frac{\pi^2}{6}(1+\alpha^2)n^2].$$

According to *Chebyshevs inequality*, the probability that the discovery time is 2 times larger than the expectation is:

$$P[W \geq 2E[W]] \leq \frac{e^2\pi^2(1+\alpha^2)/6}{[(1+\alpha)H_n - \alpha]^2} - \frac{\theta/(en)}{[(\alpha+1)H_n - \alpha]}.$$

It is close to 0 when n is large. Hence, the latency is not likely to be 2 times larger than the expectation. Therefore,

$$W = O(nlnn). \tag{12}$$

In a word, *Panacea*-NCD and *Panacea*-WCD both are bounded by $O(nlnn)$.

VI. Analysis of Asynchronous Scenario

Both *Panacea*-NCD and *Panacea*-WCD are suitable for asynchronous scenario, where nodes start the algorithm at different time slots. Under this circumstance, we assume each node u_i has a start time t_i^s and assume the maximum start time offset between any two nodes in the network is $\delta = \max(t_i^s - t_j^s)$. Denote the latency that u_i finds a new neighbor after it discovered $j-1$ neighbors as W_j', and the latency that u_i finds all neighbors as W', we analyze the algorithms as follows.

A. Analysis of Panacea-NCD

It is obvious that the start time offset δ will not affect the latency after two nodes both initiate their neighbor discovery process. That is,

$$E[W_j'] = E[W_j] + \delta. \tag{13}$$

Hence, the expected latency for a node discovering all its neighbors is

$$E[W'] \approx \frac{e}{\theta} nlnn + [\Theta(1) + \delta]n = \Theta(nlnn). \tag{14}$$

The expected latency in asynchronous scenario is δn slots larger than that in synchronous scenario. Besides, the probability that the latency is 2 times larger than the expectation is still close to 0 in the asynchronous scenario.

B. Analysis of Panacea-WCD

For *Panacea*-WCD, the latency after both two nodes start neighbor discovery will not be influenced by the start time offset δ. Therefore, we also have Eqn. (13) of latency expectation to discover a new neighbor after discovered $j-1$ neighbors. The expected latency for a node discovering all its neighbors is

$$\begin{aligned} &\frac{(\alpha+1)nH_n}{\theta} + (\delta - \alpha/\theta)n \leq E[W] \\ &\leq \frac{e}{\theta}[(\alpha+1)nH_n] + (\delta - \alpha e/\theta)n. \end{aligned} \tag{15}$$

We derive

$$E[W'] = \Theta(nlnn).$$

The expected latency in asynchronous scenario is δn slots larger than that in synchronous scenario, and bounded by $\Theta(nlnn)$. The probability that the discovery latency is 2 times larger than the expectation is still close to 0. Hence, the discovery latency is still bounded by $O(nlnn)$.

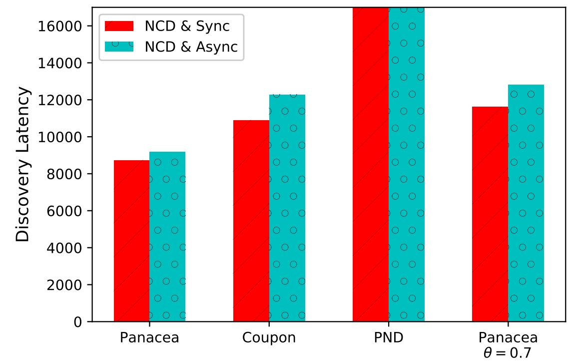


Fig. 1. Panacea-NCD has lowest discovery latency without CD.

VII. Evaluation

We implemented *Panacea* in C++ and evaluated the algorithms in a cluster of 9 servers, each equipped with an Intel Xeon 2.6GHz CPU with 24 hyper-threading cores, 64GB memory and 1T SSD. The basic settings in the evaluation are: $N = 1000$, $t_0 = 20ms$, $p_n = 0.5$, $\delta = 1000$ slots. That is, the network is composed of 1000 nodes, 0.5 density, and maximum start time offset 1000 slots, where each slot is $20ms$. These settings make the network more complicated and realistic than that in [5]–[15].

We evaluated discovery latency of *Panacea*, Coupon [13], Aloha-like [15], and PND [14] in partially-connected network in both synchronous and asynchronous scenarios. Since Coupon [13] and PND [14] do not consider duty cycle, we evaluated the tradeoff of duty cycle and discovery latency among *Panacea*, Aloha-like [15] and Hello [8]. This tradeoff could be quantified by *power-latency product* in [7], which is the product of average power consumption (i.e., duty cycle) with discovery latency.

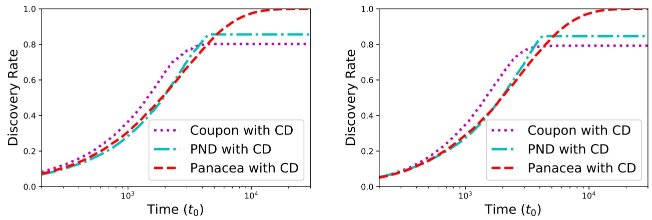
A. Comparison of Discovery Latency

If the nodes have no collision detection mechanism and they do not save energy (set duty cycle as 1), Fig. 1 shows that *Panacea*-NCD has the smallest latency for both synchronous and asynchronous scenarios compared with Coupon and PND. When *Panacea*-NCD has similar performance as Coupon, it could save 30% energy with duty cycle 0.7 (the last one of the figure).

In synchronous scenario, *Panacea*-NCD reaches discovery latency 8737 slots, which is close to the expected latency $\frac{en}{\theta}lnn = 8446$ slots; while Coupon minimizes discovery latency specifically in a fully-connected network, and reasonably has a 24.8% larger latency in partially-connected networks, let alone PND, whose latency is 5.14 times larger.

B. Comparison of Discovery Rate

With collision detection mechanisms, when duty cycle is set to 0.5, and α is set to 1, Fig. 2 shows the discovery rate (the percentage of discovered neighbors) increases, as time increases. In the synchronous scenario, *Panacea*-WCD



(a) Synchronous Scenario (b) Asynchronous Scenario

Fig. 2. Panacea-WCD achieves higher discovery rate with CD.

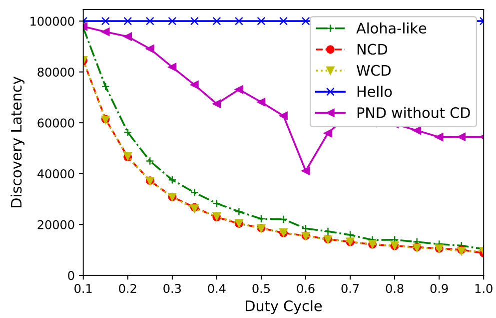
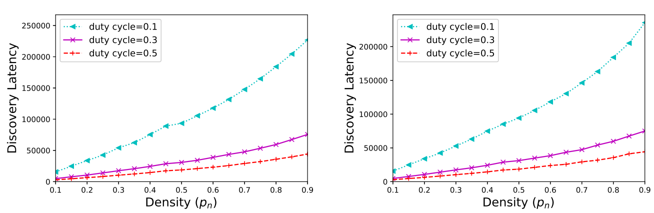


Fig. 3. Panacea has better tradeoff between duty cycle and latency.



(a) No CD (b) With CD

Fig. 4. Discovery latency increases when the network becomes denser.

could reach 100% discovery rate, while Coupon 79.2% and PND 84.6%. That's because Coupon and PND simply regard the case of no collision as a success, ignoring that some neighbors of transmitting nodes fail to discover in *Sleep* state.

C. Tradeoff: Discovery Latency v.s. Duty Cycle

Fig. 3 demonstrates that *Panacea* has a better tradeoff between discovery latency and duty cycle. From the figure, discovery latency is proportional to $\frac{1}{\theta}$ for *Panacea*, which corroborates our analyses. PND's latency is higher, because its collision rate was as high as 46.5%; Hello's latency is the highest, because collisions happen so frequently, leading the collision rate to 99.8%. The *power-latency product* of *Panacea* is 9213, while that of Aloha-like is 20.3% larger and PND is 2.69 times larger than *Panacea*.

D. Network Density

When the network becomes denser, Fig. 4 illustrates the discovery latency increases for scenarios without/with CD. This is because collisions take place more frequently in denser networks. Higher duty cycle contributes to lower latency, since discovery latency is proportional to $\frac{1}{\theta}$.

VIII. Conclusion

In this paper, we have presented *Panacea*, the first low-latency, energy-efficient neighbor discovery protocol for partially-connected wireless sensor networks, where some nodes are within radio range of each other, while others are not. First, we present two novel algorithms, *Panacea-NCD* for nodes without collision detection and *Panacea-WCD* for nodes with collision detection. Then, we prove the low bounded latency of our algorithms, and show the energy efficiency in both synchronous and asynchronous scenarios. This is because we alleviate collisions' occurrence at utmost. Furthermore, we verify our theoretical analyses by simulation. Still, *Panacea* needs to know an average number of neighbors beforehand, and we are to relax a priori knowledge in the future.

References

[1] Y. Liu, X. Mao, Y. He, K. Liu, W. Gong, J. Wang. CitySee: Not only a wireless sensor network. In *IEEE Network*, 27(5), pp.42-47, 2013.
[2] H. Alemdar, and C. Ersoy. Wireless sensor networks for healthcare: A survey. In Computer Networks, 54(15), pp.2688-2710, 2010.
[3] Motani, M., V. Srinivasan, and P.S. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In ACM *MobiCom*, 2005.
[4] Y. Wang, Y. Wang, X. Qi, L. Xu, J. Chen, and G. Wang. L3SN: A Level-Based, Large-Scale, Longevous Sensor Network System for Agriculture Information Monitoring. In *Wireless Sensor Networks*, 2(09), p.655, 2010.
[5] J.-R. Jiang, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai. Quorum-based Asynchronous Power-Saving Protocols for IEEE 802.11 Ad Hoc Networks, *Mobile Networks and Applications*, 10(1-2): 169-181, 2005.
[6] P. Dutta, and D. Culler. Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications. In *SenSys*, 2008.
[7] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar. U-Connect: A Lower Latency Energy-Efficient Asynchronous Neighbor Discovery Protocol. In *IPSN*, 2010.
[8] W. Sun, Z. Yang, X. Zhang, and Y. Liu. Hello: A Generic Flexible Protocol for Neighbor Discovery. In *INFOCOM*, 2014.
[9] Y. Qiu, S. Li, X. Xu, and Z. Li. Talk More Listen Less: Energy-Efficient Neighbor Discovery in Wireless Sensor Networks. In *INFOCOM*, 2016.
[10] L. Chen, R. Fan, K. Bian, M. Gerla, T. Wang, and X. Li. On heterogeneous neighbor discovery in wireless sensor networks. In *INFOCOM*, 2015.
[11] M. Bakht, M. Trower, and R.H. Kravets. Searchlight: Won't you be my neighbor?. In *MobiCom* (pp. 185-196), 2012.
[12] M.J. McGlynn, and S.A. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *MobiHoc* 2001.
[13] S. Vasudevan, D. Towsley, D. Goeckel, and R. Khalili. Neighbor Discovery in Wireless Networks and the Coupon Collector's Problem. In *MobiCom*, 2009.
[14] T. Song, H. Park, and S. Pack. A probabilistic neighbor discovery algorithm in wireless ad hoc networks. In *VTC*, IEEE 79th, 2014.
[15] L. You, Z. Yuan, P. Yang, and G. Chen. ALOHA-like neighbor discovery in low-duty-cycle wireless sensor networks. In *WCNC*, 2011.
[16] W.Sun, Z. Yang, X. Zhang, and Y. Liu. Energy-efficient neighbor discovery in mobile ad hoc and wireless sensor networks: A survey. in *Communications Surveys and Tutorials*, 16(3), pp.1448-1459, 2014.
[17] M. B. Nathanson. Elementary Methods in Number Theory. Vol. 195, Springer, 2000.