Group-Based Neighbor Discovery in Low-Duty-Cycle Mobile Sensor Networks

Liangyin Chen, Yuanchao Shu, *Member, IEEE*, Yu (Jason) Gu, *Member, IEEE*, Shuo Guo, Tian He, *Senior Member, IEEE*, Fan Zhang, and Jiming Chen, *Senior Member, IEEE*

Abstract—Wireless sensor networks have been used in many mobile applications such as wildlife tracking and participatory urban sensing. Because of the combination of high mobility and low-duty-cycle operations, it is a challenging issue to reduce discovery delay among mobile nodes, so that mobile nodes can establish connection quickly once they are within each other's vicinity. Existing discovery designs are essentially pairwise based, in which discovery is passively achieved when two nodes are prescheduled to wake up at the same time. In contrast, this work reduces discovery delay significantly by proactively referring wake-up schedules among a group of nodes. Since proactive references incur additional overhead, we introduce a novel selective reference mechanism based on spatiotemporal properties of neighborhood and the mobility of nodes. Our quantitative analysis indicates that the discovery delay of our group-based mechanism is significantly smaller than that of the pairwise one. Our testbed experiments using 40 sensor nodes and extensive simulations confirm the theoretical analysis, showing one order of magnitude reduction in discovery delay compared with legacy pairwise methods in dense, uniformly distributed sensor networks with at most 8.8 percent increase in energy consumption.

Index Terms—Wireless sensor networks, low-duty-cycle, proactive discovery, group-based mechanism

1 Introduction

Wireless Sensor Networks have been proposed for use in many challenging applications, such as military surveillance, scientific exploration and structural monitoring. Sustainable deployment of these systems calls for energy-efficient designs. Extensive research has indicated that energy in low-power sensors is consumed mostly by being ready for potential incoming packets, a problem commonly referred to as *idle listening* [1], [2]. For example, the widely used ChipCon CC2420 radio [3] draws 19.7 mA when receiving or *idle listening*, which is actually larger than the 17.4 mA used when transmitting. More importantly, packet transmission time is usually very small (e.g., about 1 millisecond to transmit a TinyOS packet using a CC2420 radio), while the duration of *idle listening* for reception can be orders of magnitude longer. For example, most environmental applications, such as Great Duck Island [4], sample the environment at relatively low rates (on the order of minutes between samples). With a comparable current draw and a 3$\sim$4 orders of magnitude longer duration waiting for reception, *idle listening* [1] is a major energy drain that, if not optimized, accounts for most energy in communication.

Therefore, the most effective energy conservation technique is to reduce duty-cycle by listening briefly and shutting down radios most of the time (e.g., 99 percent or more). Such a simple low-duty-cycle operation makes it difficult for nodes to discover their neighbors within physical vicinity if nodes are deployed randomly (e.g. dropped from an aircraft along its flying route) and listen to the channel in an asynchronous manner. This issue becomes even more challenging when low-duty-cycle operation is combined with the mobility of sensor nodes in many applications such as in ZebraNet [5], data collection [6], [7] and urban sensing [8]. Because mobility invalidates many assumptions implicit in low power static designs [9], such a combination imposes a time constraint on how fast nodes shall finish discovery before they are physically disconnected. Similar scenarios can also be found in friend discovery in mobile social networks (e.g. based on short-range communication like Bluetooth), and inventory and warehouse environment where active RFID tags are attached to goods and the distribution of tags changes dynamically.

Node discovery in low-duty-cycle network has attracted research attention in recent years. Previous work mainly focuses on how to ensure a pair of nodes can wake up simultaneously through a certain type of scheduling algorithms. Notable discovery designs include: stochastic-based protocols [10], [11], quorum-based protocols [12], [13], [14], [15], [16], asynchronous discovery protocols [17], Disco [18], U-Connect [19] and Searchlight [20]. These designs successfully ensure that a pair of nodes finish discovery within a bounded delay. Although literature is encouraging, we believe there is room to improve. Specifically, we notice all existing designs essentially are pairwise based. Discovery is

- L. Chen is with the School of Computer Science, Sichuan University, Chengdu 610064, China. E-mail: chenliangyin@scu.edu.cn.
- Y. Shu, F. Zhang, and J. Chen are with State Key Lab. of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China. E-mail: ycshu@zju.edu.cn, jmchen@iipc.zju.edu.cn.
- Y. Gu is with IBM Watson Health, Austin, TX 78758. E-mail: yugu@us.ibm.com.
- S. Guo is with Arista Networks, San Jose, CA 95134. E-mail: gglemon@gmail.com.
- T. He is with the Department of Computer Science and Engineering, University of Minnesota, 200 Union Street SE, Minneapolis, MN 55455. E-mail: tianhe@cs.umn.edu.

Manuscript received 12 May 2013; revised 23 Feb. 2015; accepted 23 Aug. 2015. Date of publication 29 Sept. 2015; date of current version 29 June 2016. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2015.2476471

achieved only when two nodes are pre-scheduled to wakeup simultaneously. However, if nodes can share their known schedules with each other, discovery can be achieved in a more proactive and fast manner with small overhead.

This paper presents a *Group-based Discovery* method as a performance add-on to existing pairwise mobile discovery designs. It essentially builds a schedule reference mechanism among nodes to expedite the discovery process. The operation of schedule reference is straightforward. For example, after node B discovers node A using a traditional pairwise method, node B can proactively refer (push) the wakeup-schedules of its known neighbors (such as node C) to node A. Consequently, node A can quickly discover node C indirectly via node B. Clearly, excessive reference operations would introduce high overhead in communication. The challenging issue of our work is to design a *selective* reference mechanism to trade off between reducing discovery delay and overhead.

In summary, our contributions are as follows:

- To the best of our knowledge, all previous work focuses on scheduling designs for pairwise discovery. We investigate how group-based discovery can reduce discovery delay with small overhead, and provide theoretical analysis of group-based discovery delay for mobile low-duty-cycle networks and compared with pairwise one.
- Utilizing spatiotemporal properties of neighborhood, we propose a selective reference mechanism that can avoid unnecessary references while still speeding up overall discovery process.
- We implement and evaluate our design in a physical test-bed consisting of 40 nodes and through large-scale simulations, indicating that our design is effective and suitable for resource constrained sensor nodes with two different mobility patterns. One order of magnitude reduction in discovery delay makes our discovery method a promising technique for mobile sensor network with high density.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 introduces the network model and assumptions. Section 4 introduces the basic design and makes a theoretical analysis of discovery delay. Section 5 introduces an advanced selective reference design. Experimental results using a 40-node testbed and extensive simulation results are presented in Sections 6 and 7, respectively. Section 8 concludes the work.

2 Related Work

Node discovery is nothing new and has a rich literature in both ad hoc and wireless sensor networks. Discovery in always-awake networks mainly focuses on network models with directional antennas [21], [22], [23], while solutions for discovery in duty-cycled networks are highly diverse, especially in mobile environments which impose time constraints on how fast discovery should be finished. Notable ones includes: stochastic-based protocols [10], [11], quorum-based protocols [12], [13], [14], [15], [16], asynchronous discovery protocols [17], ENDP [2], Disco [18], U-connect [19] and Searchlight [20]. In the birthday protocol [10], nodes listen, transmit or sleep in a probabilistic round-robin fashion, which statistically trades off between discovery energy with discovery latency. Due to the stochastic nature of its operation, there is no guarantee on the worst-case discovery latency. Quorum-based protocols [12], [13], [14], [15], [16] address this limitation by ensuring existence of overlapped wake-up durations between pairwise nodes within a bounded time. For instances, in [12], Tseng et al. construct an $m \times m$ grid matrix within contiguous slots. A node arbitrarily picks one column and one row of entries from the matrix to transmit and receive, respectively. Since m is a global parameter, all nodes are required to operate in a symmetric duty cycle setting (i.e., all nodes consume same amount of energy for discovery purpose). To support asymmetric duty-cycle setting, Zheng et al. [13] apply optimal block designs using difference sets to detect neighboring nodes in finite time without requiring slot alignment. Based on their method, the discovery problem in asymmetric duty-cycle setting reduces to an NP-complete minimum vertex cover problem requiring a centralized solution. ENDP is proposed in [2] to reduce the need for network scans by distributing synchronization information from nodes. However, it focuses on how to reduce the node consumption and is applied only to synchronized networks. To achieve this goal, ENDP further introduces an efficient network beacon signaling scheme through beacon transmission rate control.

To provide a distributed solution in an asymmetric design, Disco [18] introduces a neighbor discovery method based on the Chinese Reminder Theorem [24], in which each node selects a pair of primes as period independently based on the requirement of its duty cycle. For example, if node i select T_{i0} and T_{i1} as its working periods, after node i start to work, once its time counter can divide by T_{i0} or T_{i1}, it will wake up, or else be in sleep. U-connect [19] proposes a unified neighbor discovery protocol for symmetric and asymmetric duty cycle settings. Specifically authors show that U-connect is a 1.5-approximation algorithm for the symmetric asynchronous discovery scenario, and the existing protocols such as Quorum and Disco are 2-approximation algorithms. Recently, Searchlight [20], an asynchronous protocol, which can be considered as the generalized version of U-connect is proposed to improve the performance of worst-case discovery latency in 50 percent.

Although neighbor discovery techniques are diverse, all of them focus on pairwise discovery. None of aforementioned work investigate how to increase discovery probability and decrease discovery delay by selectively sharing schedule information among a group of nodes during the discovery process in an asynchronous network. In this work, we introduce a generic reference mechanism on top of current discovery methods. It serves as a performance add-on to existing discovery methods, therefore is complementary to the state-of-the-art.

3 System Models and Assumptions

In this section, we define the network model and assumptions related to group-based discovery design for mobile networks.

We focus on a network with n mobile nodes running under a low-duty-cycle mode, i.e., a node remains dormant most of time and becomes active only briefly (e.g., less than 5 percent) to sense and communicate. When a node is in the active state, it can receive packets transmitted from neighboring nodes. When a node is in the dormant state, it turns off all function modules except a timer for the purpose of waking itself up. In other words, a node can wake up to transmit a packet at any time, but can receive packets only when it is in its active state.

The working schedule of a mobile node denotes the active-dormant behaviors of the mobile node over its lifetime. It consists of a set of *active instances*, during which a node can receive packets. Each active instance m at node i can be represented by a tuple (t^i_m, d^i_m), where t^i_m denotes the starting time of the active instance and d^i_m denotes the corresponding duration of the active instance m. Since many sensor node working schedules are periodic [18], it is often sufficient to represent an infinite sequence of active instances, using repeated subsequences with a period time T_i.

Let Γ_i be the working schedule of node i and the number of active instances within a period be M, we can have $\Gamma_i = \{(t^i_1, d^i_1), (t^i_2, d^i_2), \ldots, (t^i_M, d^i_M)\}$. According to its working schedule, a node continuously transits its state between active and dormant state. Therefore, the duty-cycle of node i is $\frac{\sum_{m=1}^{M} d^i_m}{T_i}$.

To simplify our description, in the rest of the paper we assume all active instances have the same durations (τ). When a node is said to be active at time t, it has an active instance that starts at time t with duration of τ. We note that this definition of working schedule can actually accommodate active instances with varying durations. Essentially, if we let τ be the finest granularity of time durations, we can represent any node schedule with the fixed τ. In addition, we assume nodes are uniformly distributed with identical disk-shape communication range. Although collisions and packet loss are common in wireless communication, they are not considered in the design part since they are orthogonal to the main idea of our group-based discovery method. In Section 7, we will extensively analyze the impact of the both factors through simulations.

3.1 Neighbor Discovery Model

For two neighboring mobile nodes i and j to discover each other, they need to be within each other's communication range and their active instances shall at least partially overlap in time. Here we denote the basic time unit (slot size) as time instance and note that strict alignment of time instances is not required. Formally, let the duration that two nodes i and j are within each other's communication range be $[t, t + \Delta t]$, and the working schedule of node i and node j during this Δt time be $\Gamma_i^{\Delta t}$ and $\Gamma_j^{\Delta t}$ respectively, then these two nodes can discover each other if $\Gamma_i^{\Delta t} \cap \Gamma_j^{\Delta t} \neq \emptyset$. The discovery times are the elements in the set $\Gamma_i^{\Delta t} \cap \Gamma_j^{\Delta t}$. For example, if node i and node j are within each other's communication range during time [100, 200], and the active instances of node i and node j during this time interval are $\{135, 178\}$ and $\{116, 178\}$, respectively. Then node i and node j will be able to discover each other during this encounter at time 178. We note similar to legacy pairwise methods (e.g., Quorum [12], [13], [14], [15], [16], Disco [18], U-connect [19] and Searchlight [20]), by sending a discovery message at both beginning and end of an active instance, we can ensure discovery in the presence of clock skew/drift without the assumptions of time synchronization and aligned time instances.

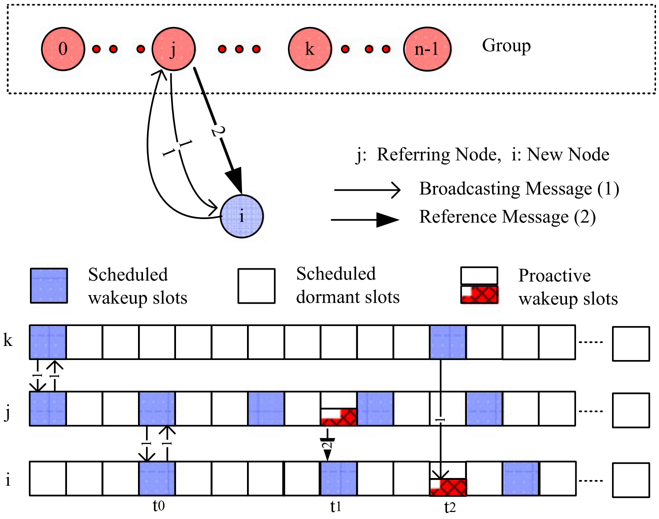


Fig. 1. Design of group-based discovery.

Note that for ease of understanding, we assume symmetric communication range of nodes and analyze mutual discovery solely. However our design can be directly extended to asymmetric discovery. In Section 7, we extensively evaluate the discovery performance with asymmetric communication capability of nodes.

4 Basic Group-Based Discovery Design

In this section, we introduce the basic design of group-based discovery and quantitatively compare our group-based discovery design with legacy pairwise node discovery approaches. Since legacy pairwise designs have effectively handled mobility in the network, in this section, we focus on explaining how our group-based discovery can reduce discovery delay in the network.

4.1 The Design for Group-Based Discovery

In traditional pairwise discovery methods for low-power wireless mobile devices, a node is able to discover a neighboring node if and only if it wakes up at the same time as its neighboring node (such as Disco). Different from pairwise discovery, in our group-based discovery design, we let individual nodes actively share their existing neighbors' working schedules (i.e., the initial waiting time and the pair of primes) with the new node that they have just discovered. In this way, the new node can quickly become aware of the wake-up times of surrounding nodes and actively verify whether it can communicate with those nodes at their wake-up times.

Fig. 1 shows the process of group-based discovery. Without loss of generality, we assume in Fig. 1 except node i, all other nodes have discovered each other and formed a group. We note if the initial size of the group is one, group-based discovery behaves the same as the pairwise discovery, because no schedule reference is needed. With more than two nodes in a group, our group-based discovery follows the following steps:

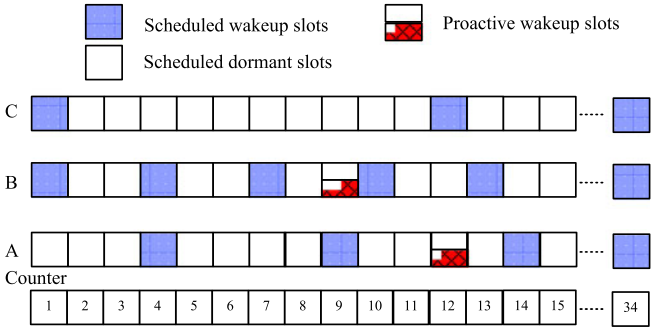


Fig. 2. An example of group-based discovery.

1) For an individual node in the network, according to its working schedule, it will periodically become active during its wakeup time instance and broadcast its existence with its own working schedule. In Fig. 1, this is denoted by broadcasting message 1 from node i and node j, respectively (as a reminder, two messages are needed to accommodate time drift). At time t_0, two nodes i and j wake up with partial overlap and are within each other's communication range, upon successful reception of each other's broadcasting message, these two nodes discover each other and become aware of each other's working schedule.
2) As node j has already discovered its group and is aware of the working schedules of other nodes in the group, it will wake up at the next active instance of node i (time t_1) and send the working schedules of all others in the group to node i. Here we call this neighbor sharing message as the *reference message*, which is shown as message 2 in Fig. 1. The node j, which sends out this reference message, is called *referring node*. And for those nodes included in the reference message, we call them *referred nodes*.
3) Upon the reception of the reference message, node i starts to verify one by one whether the referred nodes from node j are indeed its own neighbors. We note *verification can be done silently without additional messages*. Fig. 1 shows how verification is conducted: if node k is the node that wakes up first after node i has received the reference message, node i wakeup at the next active instance of node k (time t_2), trying to receive broadcasting messages from the node k. Upon reception, node i confirms that it is within the communication range of node k and adds node k to its neighbor table. This verification step continues until node i has finished verifying all referred nodes from node j.

Case study. Fig. 2 shows a case study of group-based discovery process for nodes A, B and C, which are all physically within each other's communication range. First, as both B and C wake up at time 1, they discover each other at time 1 and form a group. Then at time 4, node A and node B wake up simultaneously and become aware of each other. After node B discovers both node A and node C, it proactively wakes up at time 9 when node A is scheduled to be active, and notifies the working schedule of node C to node A. Finally at time 12, node A wakes up and silently verifies whether node C is its neighbor or not. After time 12, node A has discovered both node B and C. In contrast, if we adopt the pairwise discovery approaches [18], node A cannot discover node B and C only before time 34.

It is noted that neighbor schedule can be included in the initial broadcast message. However, as we will discuss in Section 5, this simple broadcasting mechanism ignores node diversity and makes it impossible to selectively refer nodes to different neighbors. On the other hand, referring the entire neighbor list increases the package size which brings a higher transmission overhead. Therefore, in addition to the pre-scheduled broadcasting message, in this paper, nodes proactively wake up and send selective reference list exclusively to their neighbors.

4.2 Qualitative Comparison: Group-Based versus Pairwise

This section proves qualitatively the performance gain of the group-based discovery method over the traditional pairwise discovery in terms of discovery delay.

For a pair of neighboring node i and j, assume they are within each other's communication range. Let Q be a superset of non-empty subsets under $\mathbb{Z}+_n$. Each element Γ_i in Q is the wake up schedule of node i running a certain pairwise discovery algorithm. Let C be the set of possible discovery times between node i and j, we have $C = \Gamma_i \cap \Gamma_j$. Accordingly, the first discovery delay is $\min C$. Suppose the group-based reference mechanism adds additional wake-up instances to node i, so that it can proactively discover other nodes by augmenting the original schedule Γ_i to Γ_i'. The new set of possible discovery times between node i and node j therefore is $C' = \Gamma_i' \cap \Gamma_j$. Since $\Gamma_i \subseteq \Gamma_i'$, clearly $C \subseteq C'$. Therefore the pairwise discovery delay $\min C$ is larger than or at least equal to the group-based discovery delay $\min C'$.

4.3 Analytic Comparison: Group-Based versus Pairwise

In previous section, we qualitatively prove that the group-based discovery design always has smaller or equal discovery delay than that of the pairwise discovery solutions (note the chance of equal discovery delay is extremely small). In this section, we quantitatively compare the difference of discovery delay between pairwise and group-based discovery methods. Based on the analysis, we provide numerical simulation results in Section 4.4, and proceed to the advanced group-based discovery design in Section 5.

Without loss of generality, we choose Disco [18] as the underlying pairwise discovery design. Similar derivation can be applied to the existing discovery designs as well [10], [11], [12], [13], [14], [15], [19], [20]. For pairwise discovery methods such as Disco [18], U-connect [19] and Searchlight [20], they ensure that a pair of neighboring nodes i and j can discover each other within a bounded time $T_{i,j}$. Therefore, if they are within each others communication range and $t \geq T_{i,j}$, node i and node j can discover each other with 100 percent probability. Before time $T_{i,j}$, the probability that node i and node j discovers each other can be represented by $f(i,j,t)$, no matter it is uniformly distributed or not. Consequently, we have the following equation to represent the probability a new node i discovers a node j in the group before time t:

$$P_{ij}(t) = \begin{cases} f(i,j,t), & t \in [0, T_{i,j}) \\ 1, & t \geq T_{i,j}. \end{cases} \tag{1}$$

Then for pairwise discovery methods, the probability distribution function for node i discovering all n nodes in the group before time t can be represented as $P_p^i(t) = \prod_{j=0}^{n-1} P_{ij}(t)$. The corresponding probability density function is:

$$p_p^i(t) = \sum_{j=0}^{n-1} \left[P_{ij}(t)' \prod_{k=0,k\neq j}^{n-1} P_{ik}(t) \right]. \tag{2}$$

To calculate the expected time for node i discovering all n nodes in the group, we have:

$$\begin{aligned} t_p^i = E_p^i(t) &= \int_0^{T_{\max}^i} t p_p^i(t) dt \\ &= \int_0^{T_{\max}^i} t \sum_{j=0}^{n-1} \left[P_{ij}(t)' \prod_{k=0,k\neq j}^{n-1} P_{ik}(t) \right] dt, \end{aligned} \tag{3}$$

where, $T_{max}^i = Max(T_{i,0}, T_{i,1}, \ldots, T_{i,n-1})$.

For the group-based discovery method, the probability that a new node i discovers one of the node in the group before time t is $P_g^i(t) = 1 - \prod_{j=0}^{n-1}(1 - P_{ij}(t))$. The corresponding probability density function therefore can be represented as:

$$p_g^i(t) = \sum_{j=0}^{n-1} \left[P_{ij}(t)' \prod_{k=0,k\neq j}^{n-1} (1 - P_{ik}(t)) \right]. \tag{4}$$

The expected time for node i discovering at least one node in the group is:

$$\begin{aligned} E_g^i(t) &= \int_0^{T_{\min}^i} t p_g^i(t) dt \\ &= \int_0^{T_{\min}^i} t \sum_{j=0}^{n-1} \left[P_{ij}(t)' \prod_{k=0,k\neq j}^{n-1} (1 - P_{ik}(t)) \right] dt, \end{aligned} \tag{5}$$

where $T_{min}^i = Min(T_{i,0}, T_{i,1}, \ldots, T_{i,n-1})$. After node i discovers a node, say node j in the group, according to our group-based discovery design, node j would share working schedules of all nodes in its group with node i. Then node i proactively wakes up at the active instances of non-discovered nodes in the group. Consequently, as long as all those non-discovered nodes in the group wake up at least once after node i and node j having discovered each other and node i knowing schedules of nodes in the group, node i would have discovered all nodes in the group. Therefore , the maximal time for node i discovering all n nodes in the group t_g^i can be expressed by the following formula:

$$t_g^i \leq E_g^i(t) + 2Max(T_k), k \in [0, N], \tag{6}$$

where T_k is the time gap between two consecutive wake-ups of node k. After the first discovery, in the worst case, a node takes another two $Max(T_k)$ delay to finish reference and verification.

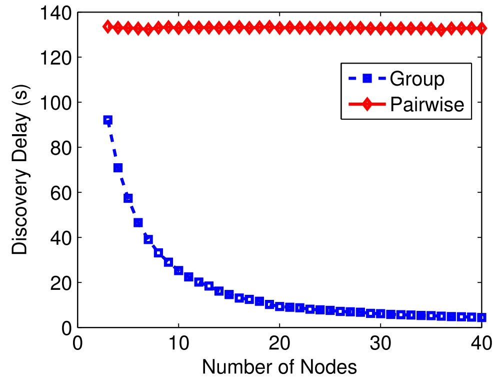


Fig. 3. Delay comparison between disco and group.

4.4 Numeric Comparison: Group-Based versus Pairwise

Based on Equations (3) and (6), we can now numerically show the performance difference in discovery delay between the group-based design and Disco [18]. Disco is an asynchronous and deterministic neighbor discovery protocol based on Chinese Remainder Theorem. When a node running on Disco, two primes as a pair should be selected according to the desired duty cycle and the node awakes only when the value of the slot counter is the multiple of the primes. The worst-case bound is the minimum product of two different primes, one of which is from a node and the other of which must be from its neighboring node. To achieve asynchronous discovery, Disco nodes send a beacon message at both beginning and end of a wakeup slot.

Suppose there is a one-hop network with n sensor nodes, each of which runs Disco with two primes 191 and 211. After $n-1$ nodes have discovered each other (i.e., they all have already known each other's schedules, and thus formed a group), the last node begins to work. Based on both Group-Based and Pairwise discovery methods, we run simulations and record the discovery delay that the last node discovers all nodes in the network in Fig. 3. From Fig. 3, we can see under all numbers of nodes, the average discovery delay of our group-based discovery design is much smaller than that of the Disco discovery design's. For example, when the number of nodes is more than 18, the delay of pairwise discovery is more than 10 times longer than our group-based method. More interestingly, we observe that as number of nodes increases, the discovery delay for our group-based discovery method actually decreases while the discovery delay for pairwise discovery design remains the same with the increasing number of nodes. This is because nodes are more likely to discover their peers in a dense environment, thus can refer neighbors more efficiently in our group-based discovery design to facilitate whole discovery process. A sample performance though Fig. 3 is, it is a clear indication that group-based discovery scales well when a network becomes very dense. In Sections 6 and 7, more evaluation results will be provided to reveal the effectiveness of the group-based approaches.

5 Advanced Group-Based Discovery Design

In Section 4, we introduce the basic concept of group-based discovery. In the basic design, we have a node j announces *all* its neighbor information to node i that it has just discovered. This simple solution ensures that the node i newly added into the group would have a complete picture of nodes in the surrounding area. However this would also waste energy unnecessarily, especially for dense network, because not all known nodes of node j are neighboring nodes of node i.

In this section, we introduce a selective reference mechanism exploiting spatiotemporal properties of mobile nodes in the network. It is based on a simple rule: node B should avoid referring the schedule of its neighbor node (say C) to node A, if node C is *not* a neighbor of node A. This is because that node A cannot communicate with a non-neighboring node C physically, even after node A knows its wake-up schedule. Following this rule, we can reduce the reference overhead of the referring node B, while still expediting the neighbor discovery process of node A.

5.1 Spatial Selection

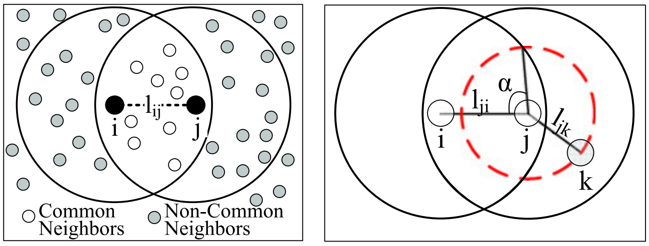
In this section, we provide theoretical foundation for referring neighboring nodes based on spatial properties. Here we do not consider the possibility of mobility, and model the movement of nodes as different spatial distributions. These restrictions will be relaxed later. The main idea of this spatial selection design is to estimate the *closeness (or proximity)* of two neighboring nodes based on the number of common neighbors they share. It is noted that spatial selection does not require calculating the exact distance between two nodes, the estimated distance between two nodes (using connection information) merely serves as an indicator of the closeness of two nodes. In the rest of this section, we use the terms *distance* and *closeness* interchangeably.

According to this estimated closeness among different mobile nodes, we then are able to selectively choose the most appropriate neighbors to refer to the newly discovered mobile node and reduce energy consumption for our group-based discovery design.

5.1.1 Theoretical Foundation

Intuitively, when two mobile nodes are closer to each other, it is more likely they would share more common neighboring nodes. For the purpose of theoretical analysis, here we assume (i) nodes are uniformly distributed with density λ, and (ii) unit disk communication model and the same communication range R of each node. We note that relaxation of these assumptions only degrade the performance of the protocol, but not the correctness of the design. For example, we can use a conservative radius in the unit disk communication model to increase the possibility of successful reference at the cost of fewer opportunities for reference.

As shows in Fig. 4a, under the assumption of local uniform distribution, the number of common neighbors is proportional to the size of the overlapping region between node i and node j with a distance of l_{ij}. Let $N_{ij}(l)$ denote the number of common neighbors, then we can have the following formula:



(a) Overlapping Communication Region Example (b) Neighbor Probability of One Node's Two Neighbors

Fig. 4. Spatial selection.

$$N_{ij}(l_{ij}) = \frac{\lambda}{\pi R^2}\left(2R^2 \arccos\left(\frac{l_{ij}}{2R}\right) - l_{ij}\sqrt{R^2 - \left(\frac{l_{ij}}{2}\right)^2}\right). \quad (7)$$

According to Equation (7), the closeness between two mobile nodes monotonically decides the number of common neighbors they have (assuming local uniform node density). By comparing the neighbor table information of two mobile nodes, we can easily find their common neighbors. Let M_{ij} be the number of common neighbors between node i and j, we estimate the closeness between those two nodes by following formula:

$$l_{ij} = N_{ij}^{-1}(M_{ij}). \quad (8)$$

N_{ij}^{-1} is the inverse function of the function 7.

Using Equation (8), we can estimate the closeness l_{jk} between referring node j and referred node k, as well as the closeness l_{ij} between referring node j and the newly discovered node i.

To calculate the probability that node j's neighboring nodes i and k are within each other's communication range, let us look at the illustration shown in Fig. 4b. From Fig. 4b, it is clear that if node k falls within the overlapping communication region between node i and node j, node k is a common neighbor for node i and node j. Obviously, if we fix l_{jk} and l_{ij}, then node k can only be situated on the dashed circle. If $l_{jk} + l_{ij} > R$, node k is the common neighbor of node i and node j, only if node k is located at the dashed circle segment inside the circle i. According to the law of cosines, the angle α in Fig. 4b can be represented as $\alpha = \arccos(\frac{l_{jk}^2 + l_{ij}^2 - R^2}{2 l_{jk} l_{ij}})$. Then the probability that the referred node k by referring node j is also the neighbor of node i can be expressed as: $\frac{2\alpha}{2\pi} = \frac{1}{\pi}\arccos(\frac{l_{jk}^2 + l_{ij}^2 - R^2}{2 l_{jk} l_{ij}})$. When $l_{jk} + l_{ij} \leq R$, then it is clear from Fig. 4b that node k is always falling into the overlapped region between node i and node j. Therefore, the probability that node k is a common neighbor of node i and j is 100 percent in this scenario.

By combining above two cases, we can have the following equation to represent the probability that node j's neighboring nodes i and node k are also within each others communication range as:

$$P_{j,ik}(l_{jk}, l_{ij}) = \begin{cases} \frac{1}{\pi}\arccos(\frac{l_{jk}^2 + l_{ij}^2 - R^2}{2 l_{jk} l_{ij}}) & l_{jk} + l_{ij} > R \\ 1 & l_{jk} + l_{ij} \leq R. \end{cases} \quad (9)$$

Clearly, by setting different threshold values for $P_{j,ik}(l_{jk}, l_{ij})$, a node can selectively reference its neighboring nodes to a newly discovered node, therefore tradeoff among energy consumption, discovery delay and discovery probability. For example, if system wants to reduce energy consumption, we should set a high threshold value and have the nodes in the network reference less neighbors for discoveries. On the other hand, when system demands low discovery delay, we should set a low threshold value.

5.1.2 Analysis of the Worst and Average Cases

Based on the analysis above, in this section we study the worst and average case that group-based reference would be helpful. By taking the derivative of Equation (7), we have $N_{ij}(l)' = \frac{\lambda(-2+\frac{1}{2}(\frac{l}{R})^2)}{\pi R\sqrt{1-(\frac{l}{2R})^2}} < 0$. Consequently, Equation (7) is a monotonically decreasing function with respect to l. For example, when $l = 0$, which means node i and node j are identically located, we have $N_{ij}(0) = \lambda$. In contrast, when $l = R$, we have $N(R) \approx 0.391\lambda$. This indicates even when node i and node j are at the edge of their communication range (the worst case), they still share about 39.1 percent of common neighbors.

In addition, when mobile nodes are uniformly distributed in the network [25], the probability density function for the distance between two neighboring nodes can be represented as:

$$p(l) = \frac{2l}{R^2}. \tag{10}$$

Then the expected closeness between two neighboring nodes is:

$$E(l) = \int_0^R p(l) l \, dl = \frac{2}{3}R. \tag{11}$$

Given this expected closeness of $\frac{2}{3}R$ between two neighboring nodes, according to Equation (7), in the average case, the number of common neighbors of two neighboring nodes is approximately 0.5836λ . In other words, if node i and node j are neighboring nodes, another node k which is the neighbor of node i has about 58.36 percent chance to be the neighbor of node j, a sufficiently large chance to motivate us to use the group-based discovery method.

5.2 Temporal Selection

Due to the mobility, the neighborhood of a mobile node i also changes dynamically. Therefore we need to have a low-cost and systematic method to discard the stale neighbor information for mobile nodes in the network. For group-based discovery design, this discard of stale neighbor information is particular important as it directly affects the energy consumption for node discoveries. If a node sends out those stale neighbor nodes, it wastes its own energy for data transmission, as well as the energy for the reception node to verify those stale neighbors. In this section, we discuss theoretical foundations for setting Time-to-live (TTL) values for neighbor information in a mobile network.

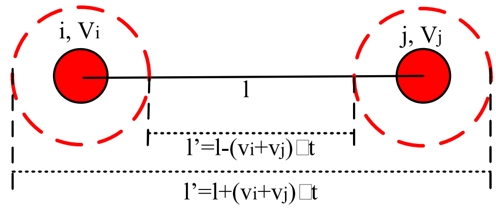


Fig. 5. Temporal selection example.

Taking scenario in Fig. 5 as an example, after time Δt, node i and node j are still within the range of the dashed circles. Assuming the velocity of node i and node j is v_i and v_j respectively, we can calculate the closeness between node i and node j after time Δt is within the range of $[l-(v_i+v_j)\Delta t, l+(v_i+v_j)\Delta t]$.

Let the maximum node velocity in the network be v_{max}, which is an application-specific parameter set by users. As the closeness l between two neighboring nodes i and j can be estimated by Equation (8), the minimal duration for those two nodes moving out each other's communication range is $\frac{R-l}{2v_{max}}$. So as long as we set $TTL_{ij} \leq \frac{R-l}{2v_{max}}$, there is a high probability that a neighboring node j of node i is still within its communication range after they first discovered each other. Similar to the threshold value for spatial selection, a smaller TTL value leads to less energy consumption but also smaller discovery probability. While a larger TTL value costs more energy, but with larger discovery probability for mobile nodes.

5.3 Put Them together

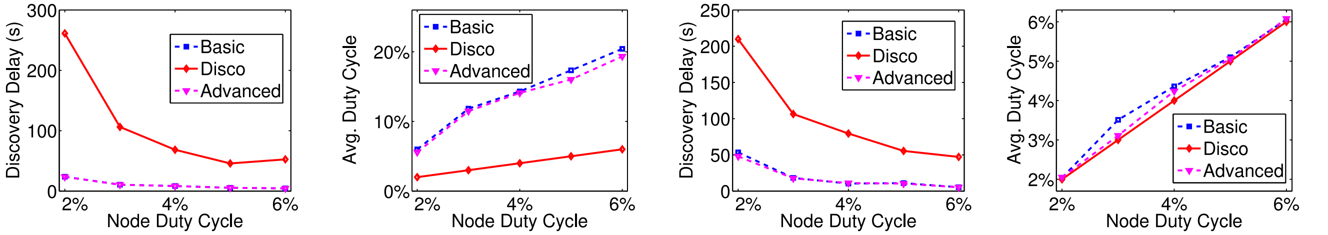
To combine the spatial selection and temporal selection, we first use temporal selection filters to filter neighbor nodes of one node (the referring node). Second, we use spatial selection filters to calculate the distance between the node and all its current neighbors (potential referred nodes), and refer nodes based on the neighbor probability of the current neighbors and the receiver.

6 Experiment

In order to validate our group-based discovery design in practice, we have fully implemented both basic and advanced group-based discovery designs on the TinyOS/Mote platform (using IEEE 802.15.4 standard). To compare the performance of our group-based discovery designs, we also implement Disco [18]—a pairwise asynchronous node discovery method—on our platform.

6.1 Experimental Setup

During the experiment, we place up to 40 static sensor nodes on a 2.6 m $\times$ 2.6 m square field. Fig. 8 shows a picture of the testbed for the experiment. Algorithm performance with a larger area of distribution of mobile sensor nodes will be evaluated through simulations in the next section. During the experiment, we do not use any synchronization mechanism to synchronize the clock among nodes in the network. To ensure uniformly distributed random starting times, initially all nodes are active and listen to the channel until receive a beacon message from a sink node. They start working after waiting for a uniformly randomly generated waiting time between 0 and 5 seconds.



(a) Single-Hop Discovery Delay vs. Duty Cycle (b) Single-Hop Discovery Energy vs. Duty Cycle (c) Multi-Hop Discovery Delay vs. Duty Cycle (d) Multi-Hop Energy vs. Duty Cycle

Fig. 6. Impact of duty cycle.

Similar to Disco [18], each node in the network randomly generates its working schedule based on a designated duty cycle (e.g., $prime_1 = 67, prime_2 = 71$ in the default 3 percent duty cycle setting), and periodically wakes up according to its working schedule. In reference messages transmission, we simply transmit two consecutive packets if all information cannot fit into a reference packet. We compare the discovery delay (i.e., time spent on discovering all neighbors of all nodes) and the average duty cycle (i.e., average number of wakeup slots to the number of all slots) among Disco, the basic group-based design and the advanced group-based design. In our experiment, we set the duration of one time instance to 200 ms. The default duty cycle for the network is 3 percent and each case runs three times.

6.2 Performance Experiments

One-hop experiments. By using the maximum transmission power for each node, all deployed nodes are within a one-hop neighborhood. For one-hop experiments, we collect the data after each node having discovered all its neighbors, i.e., the percentage neighbor discovered is 100 percent.

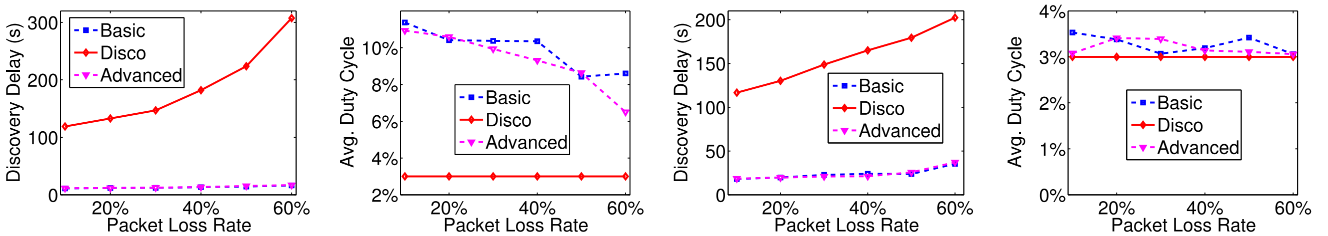
Two-hop experiments. We reduce the communication range of each node to 2 m by decreasing its transmission power and only 20 nodes are deployed. Each experiment lasts for 10 minutes, which is slightly longer than the maximum designed bounded time for discovery.

6.2.1 Impact of Node Duty Cycles

The first experiment tries to investigate the impact of duty cycle on system performances. Fig. 6 shows for both single-hop and multi-hop experiments, discovery delay decreases for all three designs when the duty cycle in the network increases. Identical discovery delays of basic and advanced design can be found in Fig. 6a since all nodes are within each other's communication range in the one-hop scenario. However, under all duty cycles, the discovery delay for Disco is significantly longer than the group-based discovery designs. For example, for single-hop experiments, when the duty cycle is 2 percent, the discovery delay for Disco is 261.32 s, which is over one order of magnitude longer than that of the advanced design (23.56 s). As for energy, both basic and advanced designs consume more than that by Disco due to additional reference and verification operations. For example, in one-hop scenario (Fig. 6b), average duty cycle (average number of the wakeup slots to the number of all slots) of the basic design is 11.8 percent, 8.8 percent higher than Disco which is fixed to 3 percent. In two-hop scenario (Fig. 6d), average duty cycles of the basic design and the advanced design decrease to 3.5 and 3.1 percent respectively, slightly higher than the 3 percent duty cycle in Disco. We note in one-hop scenario, we have a higher node density, therefore more nodes need to wake up proactively for reference and verification.

6.2.2 Impact of Packet Loss

The loss of packets would increase the delay of reference and verification, consequently increasing the discovery delay. In this experiment, we set the packet loss ratio be 10-60 percent by random dropping packets intentionally in the testbed. Fig. 7 shows the impact of packet loss on discovery delay and energy consumption. Discovery delay in the advanced and the basic is far less than that in Disco, especially in the cases with higher packet loss ratio. For example in 60 percent packet loss ratio case in one-hop scenario, the delay of Disco is 307.42 s, nearly 20 times more than 16.35 s of the basic and 16.89 s of the advanced. In addition the delay in the basic and advanced designs keeps to be stable relatively, but increases significantly in Disco. This is because nodes are still able to receive reference messages thus are more likely to discover their neighbors compared with the pairwise method even when packet loss ratio increases. As far as the energy consumption, packet loss

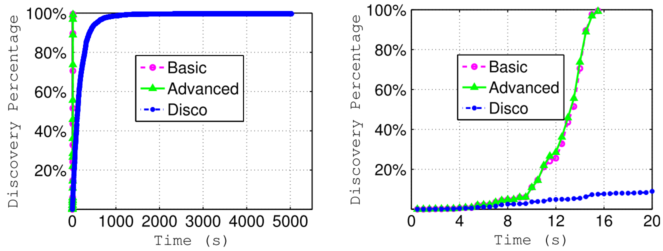


(a) Single-Hop Discovery Delay vs. Packet Loss Ratio (b) Single-Hop Energy vs. Packet Loss Ratio (c) Multi-Hop Discovery Delay vs. Packet Loss Ratio (d) Multi-Hop Energy vs. Packet Loss Ratio

Fig. 7. Impact of packet loss.

(a) One-Hop Scenario (b) Two-Hop Scenario

Fig. 8. The testbed for the experiment.



(a) Long-Tail Delay (b) Discovery Percentage

Fig. 9. The percentage of discovery over time.

does not affect the pairwise method, because broadcasting messages are not retransmitted. However, packet loss reduces energy consumption in advanced and basic methods, because few reference messages will be propagated further within a neighborhood.

6.2.3 *Discovery Percentage over Time*

In this experiments, Fig. 9 plots the node discovery percentage over time in one-hop scenario. Fig. 9a shows that over 98 percent neighbors have been discovered before 1,000 s in the Disco method, but it takes more than 4,000 s for the remaining 2 percent neighbors to be discovered. In contrast, Fig. 9a illustrates that there are nearly no long-tail for the discovery time in both basic and advanced methods, a clear indication that group methods are much better than the Disco method in the worst discovery delay, thanks to our reference mechanism. Fig. 9b takes a close look at the first 16 seconds. It shows that all neighbors in group designs have been discovered before time 16 s, i.e., 100 percent discovery percentage, which is far more than that in Disco (below 10 percent).

7 Evaluation

To understand the system performance of our group discovery designs under diverse network settings, in this section we provide simulation results with up to 500 mobile nodes.

7.1 Simulation Setup

We build our own simulation environment using C++ with 500 mobile nodes deployed in a $1{,}000\ \mathrm{m} \times 1{,}000\ \mathrm{m}$ two-dimensional field. Except where otherwise specified, the average velocity of mobile nodes in the network is $1\ \mathrm{m/s}$ with a random deviation of 30 percent (i.e., $v_{\max} = 1.3\ \mathrm{m/s}$),[1] which represents the normal adult walking speed. To reveal the system performance under more realistic communication models, we adopt asymmetric communication links and consider packet collisions in all simulations. Specifically, the communication range of the mobile nodes is set to be 100 meters. Default density of nodes is 8.48, the packet loss ratio and the degree of radio irregularity [2] are set to 10 percent. In addition, the default TTL is set to be $50\,\mathrm{s}$ (5,000 slots), the default spatial reference probability is set to be 0.5.

For the mobility model, we adopt waypoint model (uniform node distribution) [26], [27] as our default mobility model and also study system performance under other mobility models such as hotspot model (non-uniform node distribution) [28] in Section 7.3.3.

We adopt Searchlight [20], the state-of-the-art pairwise discovery protocol as the baseline method and compare it with our group-based discovery method with different node velocities and node densities. We further compare our proposed method with two other pairwise-based protocols (Disco [18] and U-Connect [19]). Each simulation is repeated 40 times and the average results are reported in the following sections.

7.2 Metrics

To evaluate system performance, we mainly use three metrics to reveal the advantages and disadvantages of various designs.

- *Discovery Probability:* In low-power mobile networks [29], even two nodes are physically within each other's communication range, they may still not be able to discover each other due to the asynchronous working schedules. The discovery probability metric indicates the probability that two nodes are physically within each other's communication range and have successfully discovered each other.
- Discovery Delay: The discovery delay denotes the time duration that mobile node A first discovers node B after node A entered into node B's communication range. This metric is intended to reveal the agility of various designs.
- *Energy Consumption:* In our group discovery designs, nodes are proactively wake up so as to increase discovery probability and reduce discovery delay. In our simulations, we use the *average duty cycle* of individual nodes to investigate the actual energy consumption of various designs. This includes both the slots that a node is scheduled to wake up according to its working schedule and the slots when node proactively wakes up to discover other nodes.

7.3 Performance Evaluation

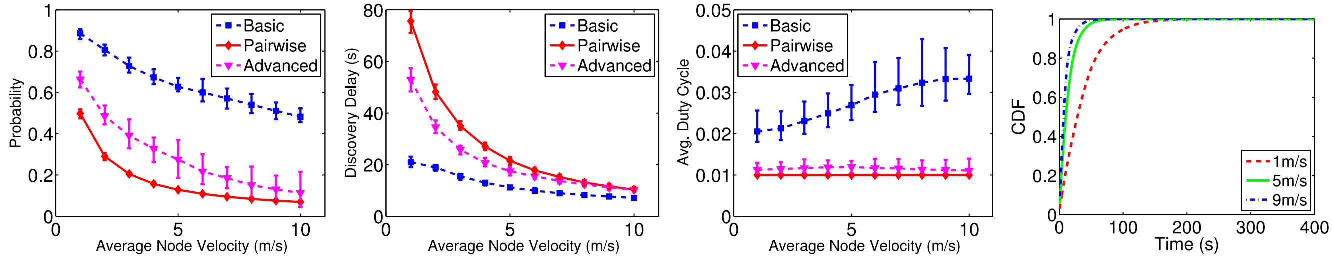
In this section, we systematically study the performance of our group discovery designs under different node velocities, node densities, mobility patterns and radio irregularities.

7.3.1 *Impact of Node Velocities*

In mobile networks, the velocity of nodes in the network has significant effects on the system performance. Fig. 10 studies the impact of average node velocity for mobile node discoveries. Fig. 10 reveals that as node velocity in the network

1. In all speed settings, the actual node velocity is a uniformly randomly generated number between 0.7 and $1.3\,v$, where v is the average velocity.

2. Please see Section 7.3.4 for definition of the degree of radio irregularity.



(a) Discovery Probability vs. Node Velocity (b) Discovery Delay vs. Node Velocity (c) Avg. Duty Cycle vs. Node Velocity (d) CDF of Discovery Delay (Advanced Design)

Fig. 10. Impact of node velocities.

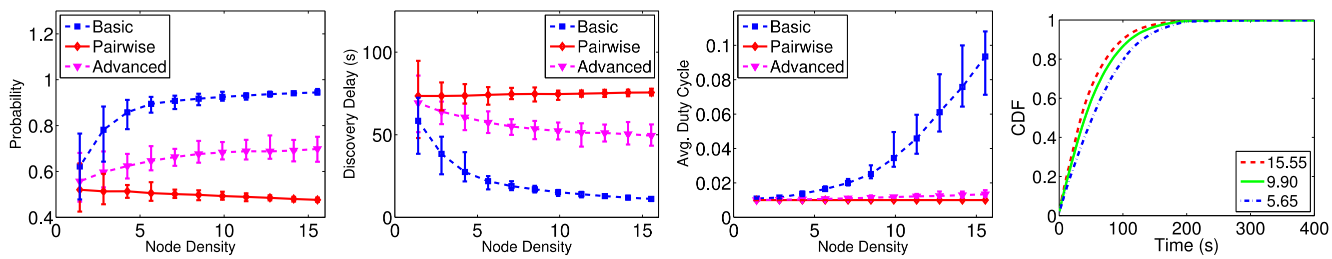
increases, the discovery probability decreases for all three design schemes. This is because the duration that two nodes are physically within each other's communication range shortens as node velocity increases. Consequently, the discovery delay for three design schemes also decreases. For the energy consumption, as increasing node velocity leads to increased number of mobile node discoveries in the network, our basic group discovery design wakes up more on referring and verifying neighbors, therefore consumes more energy as average node velocity in the network increases. However the advanced design, which selectively refers neighbors, achieves better delay-energy tradeoffs. From Fig. 10, we can see our group discovery design performs well under a wide range of node velocities, from 1 meter per second for normal walking to over 20 miles per hour for a typical vehicle velocity in urban settings. At about $2\ \mathrm{m/s}$, the discovery probability for advanced group discovery is about 1.67 times higher than the pairwise solution, while the discovery delay is about 28.36 percent shorter. Furthermore, for energy consumption, the actual duty cycles for pairwise design and our advanced group discovery are 1.00 and 1.15 percent, respectively.

To examine the variation of the group-based discovery design, we also plot error bars of all designs in Fig. 10 and show the CDF of the discovery delay of the advanced design in Fig. 10d. From Fig. 10, we find the the performance of the advanced design is relatively stable and the discovery delay varies within a small range. For example, the discovery of 80 percent neighbors takes less than 60 seconds when the average velocity of nodes is $1\ \mathrm{m/s}$.

7.3.2 *Impact of Node Densities*

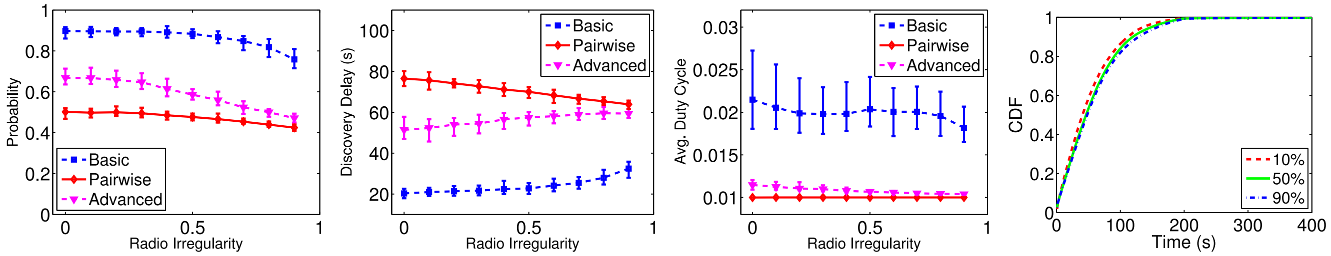
As our group discovery design leverages on the information sharing among neighboring nodes, we are also interested to investigate the impact of node densities on system performance. Interestingly, from Fig. 11 we can see as node density increases, the discovery probability for group discovery increases. This is because for the pairwise design, depending on the node velocity, a node can only discover a fixed number of neighboring nodes within a time frame. And this leads to relatively stable discovery probability of Searchlight (actually more collisions in a denser environment causes a little decreases). In contrast, our group discovery method benefits from the increasing node density as more neighbor information is shared in the network. For the same reason, the discovery delay for pairwise design stays relatively stable with increasing node density, while the delay for group discovery decreases. For example, at node density of 8.48, the discovery probability for pairwise design and advanced group discovery is 49.81 and 67.59 percent, respectively. The corresponding discovery delay is 74.73 and 53.51 s, respectively. From Fig. 11d we can see more than 80 percent neighbors have been discovered in 100 seconds under all density settings. For energy consumption, the actual duty cycle for pairwise design and our advanced group discovery is 1.00 and 1.14 percent.

However, the gain of the group-based design drops with decreasing node density. Taking the advanced design and the pairwise design as an example, compared with the 28.4 percent delay decrease when node density is 8.48, discovery delay of the advanced design is only 6.23 percent shorter (69.50 versus 74.12) than the pairwise design when the density becomes 1.40. In addition, the variation of discovery probability and delay become larger in sparser networks. For example when the network density is 1.4, the worst-case discovery probability and delay of the advanced design can be worse than the average performance of the pairwise design.



(a) Discovery Probability vs. Node Density (b) Discovery Delay vs. Node Density (c) Avg. Duty Cycle vs. Node Density (d) CDF of Discovery Delay (Advanced Design)

Fig. 11. Impact of node densities.



(a) Discovery Probability vs. Radio Irregularity (b) Discovery Delay vs. Radio Irregularity (c) Avg. Duty Cycle vs. Radio Irregularity (d) CDF of Discovery Delay (Advanced Design)

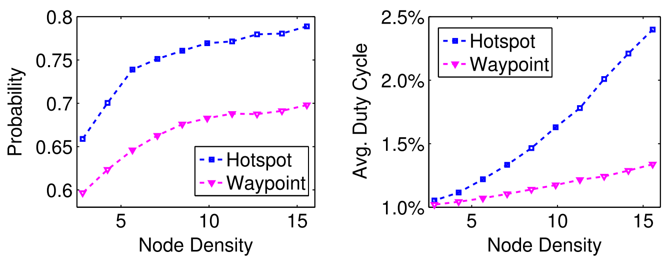
Fig. 12. Impact of radio range irregularity.

7.3.3 Impact of Mobility Patterns

In the design section, we assume the mobile nodes in the network are uniformly distributed. In order to investigate the impact of different node distributions, in Fig. 13, we show the system performance using advanced group discovery design under both random waypoint mobility model [26] (uniform node distribution) and hotspot mobility model [28] (nonuniform node distribution). For hotspot mobility model, it mimics the behavior of people moving from a point of interest to another and stopping for a certain time at each point of interest before moving to the next one. From Fig. 13a, we can see with increasing node density, the discovery probabilities for both mobility models are increasing. However, the hotspot mobility model has higher discovery probability than the random waypoint mobility model at all node densities. This is because under hotspot mobility model, nodes are distributed at several hot spots in the network, incurring more group discovery among neighboring nodes. And due to individual nodes increase their active time instances for neighbor reference and verification under hotspot mobility model, we also observe higher energy consumption under the hotspot model over the random waypoint model.

7.3.4 Impact of Radio Range Irregularities

In our analytical model, for the purpose of simplicity in presentation, we model communication range as a unit disk and assume the uniform distribution of nodes in the network. In this section, we study the impact of non-disk communication range on our design. Specifically, we use the Degree of Irregularity (DOI) model in [30] to characterize the irregular communication range. DOI is defined as the maximum range variation per unit degree change in the direction of radio propagation. When the DOI is zero, the communication range is a perfect circle whereas the communication range becomes more and more irregular when the DOI increases.



(a) Discovery Probability (b) Avg. Duty Cycle

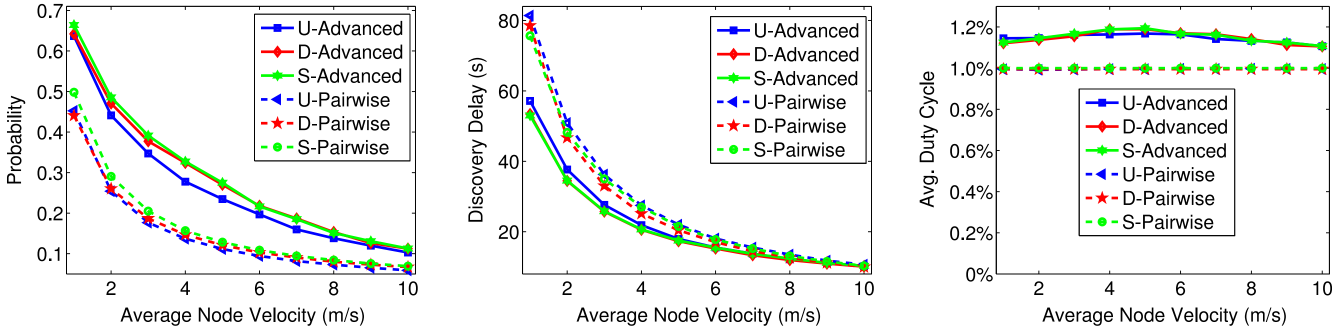
Fig. 13. Impact of mobility patterns.

In Fig. 12, we show the system performance under different degrees of radio irregularity. As degree of radio irregularity increases, we can see the discovery probability for all three schemes decreases. This is because with the increasing range of radio irregularity, the accuracy of the neighbor estimation using a fixed radio range between two nodes decreases. Therefore the discovery probability decreases.

For the discovery delay, both basic and advanced discovery methods increase as the degree of radio irregularity increases, while the delay for pairwise discovery decreases. This is due to the reason that when a referring message in our group discovery designs is lost, nodes can still discover their neighbors within a relatively short period of time using other referring messages. However for pairwise discovery, nodes have to wait until the next pre-scheduled wakeup slots to discover their neighbors, which is a relatively long period of time. Therefore, the pairwise discovery delay, which averages delays of much fewer discovered node, decreases with increasing radio irregularity. However, even with 70 percent degree of radio irregularity, compared with the pairwise design, our advanced group design owns a 7.10 percent higher discovery probability with a 11.69 percent lower discovery delay and 1.05 percent duty cycle. In Fig. 12d, we can see that 80 percent of neighbors can be discovered in 96 seconds even when the DOI is as high as 90 percent.

7.3.5 Comparison with Baseline Methods

In this section, we are interested to compare the discovery performance of our proposed method with state-of-the-art approaches. We apply our Group-Based method to three baseline pairwise discovery methods, namely Disco [18], U-Connect [19] and Searchlight [20], and show discovery probability, discovery delay and energy consumption with different average node velocities in Fig. 14. From Fig. 14, we find that our advanced group-based method can be efficiently applied to all three pairwise approaches which can simultaneously increases the discover probability and reduce the discovery delay. Taking U-Connect as an example (U-Advanced versus U-Pairwise), when the average velocity of nodes is 5 m/s, the discovery probability has been improved for 90 percent (0.11 versus 0.21) with a 21 percent shorter (24 versus 19) discover delay at the cost of 20 percent more energy consumption.



(a) Discovery Probability vs. Node Velocity (b) Discovery Delay vs. Node Velocity (c) Avg. Duty Cycle vs. Node Velocity

Fig. 14. Comparison with baseline methods.

7.4 Evaluation of Design Decisions

For our advanced group design, we need to set appropriate spatial and temporal selection thresholds to make a tradeoff between system performance and energy consumption. In this section, we investigate the impact of different choices of spatial and temporal selection thresholds on the system performance and offer some guidelines to determine the parameters.

7.4.1 *Choices of Spatial Selection Threshold*

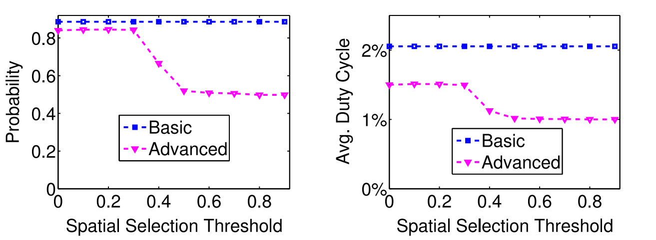
Spatial selection threshold is an essential parameter for our advanced group discovery design. Fig. 15 shows the impact of spatial selection threshold on the discovery probability and the corresponding energy consumption. As shown in Fig. 15a, the discovery probability drops significantly when the spatial selection threshold increases above 60 percent. This result actually confirms our theoretical analysis in Section 5.1 that two neighboring nodes of a common node have about 58.36 percent probability to be neighbors. From this study, we suggest set default spatial selection threshold to be 0.50 in practice. In addition, the higher threshold values above 60 percent eliminate the exchange of most of neighbor information, consequently lead to reduced energy consumption as shown in Fig. 15b. In theoretical analysis, we assume a given node density λ. However in our simulations, we relax the assumption and nodes do not need to know the density λ. Instead, we use the ratio of the number of current common neighbors between two nodes to the number of the total neighbors to estimate the distance between the referring node and the new discovered node. Initially, since there is no exchange of neighbor information, nodes will refer all their neighbors.

7.4.2 *Choices of Temporal Selection Threshold*

In addition to the spatial selection threshold, in this section, we discuss the impact of temporal selection threshold on the performance of advanced group discovery design. In Fig. 16, it is clear to see that as TTL threshold value increases, the discovery probability also increases. This is because as individual nodes keeping neighbor entries in their neighbor table longer, there is more information that it can share with the newly discovered node and resulting the higher discovery probability. Meanwhile, as more information is shared among neighboring nodes in the network, individual nodes wake up more and try to verify more nodes in our group discovery design. And this leads to the higher energy consumption. From Fig. 16, we also see that with the increase of the TTL, the discovery probability in advanced group discovery design is closer to that in basic group discovery method, but its energy consumption is still much lower than that in basic group discovery design. This further demonstrates that our advanced group discovery design is much more energy efficient than the basic design.

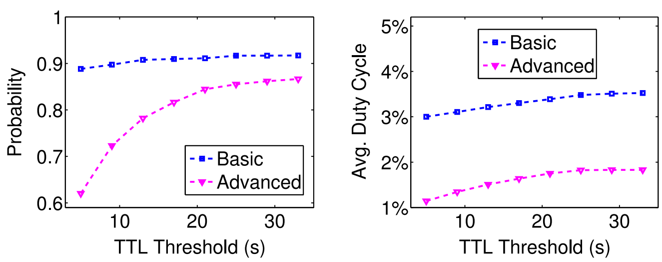
8 Conclusion

This paper presents a Group-based Discovery method as a performance add-on to existing pairwise discovery designs. It essentially builds a schedule reference mechanism among nodes to expedite the discovery process of pairwise discovery designs, and provides theoretical analysis of group-based discovery delay. We first introduce the basic reference design, followed by an advanced group-based discovery design, which exploiting spatiotemporal properties to selectively choose neighboring nodes for energy-efficient reference. This way, we can further reduce the discovery delay with small overhead. We evaluate our designs in a physical



(a) Discovery Probability vs. Spatial Threshold (b) Avg. Duty Cycle vs. Spatial Threshold

Fig. 15. Impact of spatial selection threshold.



(a) Discovery Probability vs. TTL (b) Avg. Duty Cycle vs. TTL

Fig. 16. Impact of time-to-live threshold.

test-bed and through large-scale simulations. Compared with the state-of-the-art pairwise discovery solutions, our designs show one order of magnitude reduction in discovery delay in dense, uniformly distributed homogeneous sensor networks with maximum 8.8 percent increase in energy.

Acknowledgments

This research was supported in part by the US National Science Foundation (NSF) grants CNS-0845994, CNS-0917097, NSFC grants 61228302, 61373091, 61273361, 111 Program under grant B07031, grant SUTD SRG ISTD 2010 002 and SUTD-ZJU/RES/03/2011. A preliminary version of this paper was presented in 2012 IEEE SECON in Seoul, South Korea, and appears in the conference proceedings. Jiming Chen is the corresponding author.

References

[1] J. Polastre, J. L. Hill, and D. E. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM 2nd Int. Conf. Embedded Netw. Sens. Syst.*, 2004, pp. 95–107.

[2] M. Kohvakka, J. Suhonen, M. Kuorilehto, V. Kaseva, M. Hännikäinen, and T. D. Hämäläinen, "Energy-efficient neighbor discovery protocol for mobile wireless sensor networks," *Ad Hoc Netw.*, vol. 7, no. 1, pp. 24–41, 2009.

[3] Texas Instruments, 2.4 GHz IEEE 802.15.4 / ZigBee-Ready RF Transceiver (Rev. B). [Online]. Available: http ://focus.ti.com/docs/prod/folders/print/cc2420.html, 2013.

[4] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, "An analysis of a large scale habitat monitoring application," in *Proc. ACM 2nd Int. Conf. Embedded Netw. Sens. Syst.*, 2004, pp. 214–226.

[5] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," in *Proc. 10th Int. Conf. Archit. Support Programm. Lang. Operating Syst.*, 2002, pp. 96–107.

[6] X. Wu, K. N. Brown, and C. J. Sreenan, "SNIP: A sensor node-initiated probing mechanism for opportunistic data collection in sparse wireless sensor networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2011, pp. 726–731.

[7] M. D. Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Trans. Sens. Netw.*, vol. 8, no. 1, pp. 7:1–7:31, Aug. 2011.

[8] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G-S. Ahn, and A. T. Campbell, "The BikeNet mobile sensing system for cyclist experience mapping," in *Proc. ACM 5th Int. Conf. Embedded Netw. Sens. Syst.*, 2007, pp. 87–101.

[9] P. Dutta and D. Culler, "Mobility changes everything in Low-power wireless sensornets," in *Proc. USENIX 12th Conf. Hot Topics Operating Syst.*, 2009, p. 7.

[10] M. J. McGlynn and S. A. Borbash, "Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks," in *Proc. ACM 2nd ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2001, pp. 137–145.

[11] S. A. Borbash, A. Ephremides, and M. J. McGlynn, "An asynchronous neighbor discovery algorithm for wireless sensor networks," *Ad Hoc Netw.*, vol. 5, no. 7, pp. 998–1016, 2007.

[12] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-saving protocols for ieee 802.11-based Multi-hop ad hoc networks," in *Proc. IEEE 21st Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2002, pp. 200–209.

[13] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2003, pp. 35–45.

[14] S. Lai, B. Zhang, B. Ravindran, and H. Cho, "CQS-pair: Cyclic quorum system pair for wakeup scheduling in wireless sensor networks," in *Principles Distrib. Syst.*, Lecture Notes in Computer Science, 2008, vol. 5401, pp. 295–310.

[15] S. Lai, B. Ravindran, and H. Cho, "Heterogenous quorum-based wake-up scheduling in wireless sensor networks," *IEEE Trans. Comput.*, vol. 59, no. 11, pp. 1562–1575, Nov. 2010.

[16] J. B. Link, C. Wollgarten, S. Schupp, and K. Wehrle, "Perfect difference sets for neighbor discovery: Energy efficient and fair," in *Proc. 3rd Extreme Conf. Commun.: The Amazon Expedition*, 2011, pp. 5:1–5:6.

[17] N. Karowski, A. C. Viana, and A. Wolisz, "Optimized asynchronous multi-channel neighbor discovery," in *Proc. IEEE INFOCOM*, 2011, pp. 536–540.

[18] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. 6th ACM Conf. Embedded Netw. Sens. Syst.*, 2008, pp. 71–84.

[19] A. Kandhalu, K. Lakshmanan, and R. (Raj) Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sens. Netw.*, 2010, pp. 350–361.

[20] M. Bakht, M. Trower, and R. H. Kravets, "Searchlight: Won't you be my neighbor?" in *Proc. ACM 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 185–196.

[21] S. Vasudevan, J. Kurose, and D. Towsley, "On neighbor discovery in wireless networks with directional antennas," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2005, pp. 2502–2512.

[22] G. Jakllari, W. Luo, and S. V. Krishnamurthy, "An integrated neighbor discovery and MAC protocol for ad hoc networks using directional antennas," *IEEE Trans. Wireless Commun.*, vol. 6, no. 3, pp. 1114–1124, Mar. 2007.

[23] S. Vasudevan, D. Towsley, and D. Goeckel, "Neighbor discovery in wireless networks and the coupon collectors problem," in *Proc. ACM 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 181–192.

[24] G. H. Hardy, E. M. Wright, D. R. Heath-Brown, and J. H. Silverman, *An Introduction to the Theory of Numbers* (Oxford mathematics). London, U.K.: Oxford Univ. Press, 2008.

[25] D. Moltchanov, "Distance distributions in random networks," *Ad Hoc Netw.*, vol. 10, no. 6, pp. 1146–1166, 2012.

[26] D. N. Alparslan and K. Sohraby, "Two-dimensional modeling and analysis of generalized random mobility models for wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 3, pp. 616–629, Jun. 2007.

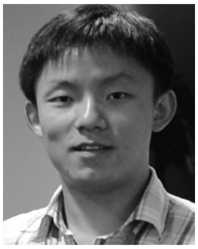
[27] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proc. IEEE 22nd Annu. Joint Conf. IEEE Comput. Commun.*, 2003, pp. 1312–1321.

[28] E. Hyytia, P. Lassila, and J. Virtamo, "A Markovian waypoint mobility model with application to hotspot modeling," in *Proc. IEEE Int. Conf. Commun.*, 2006, pp. 979–986.

[29] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 640–652, Jun. 2006.

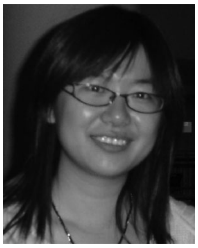
[30] G. Zhou, T. He, S. Krishnamurthy, , and J. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proc. ACM 2nd Int. Conf. Mobile Syst., Appl., Serv.*, 2004, pp. 125–138.

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, and natural language processing.

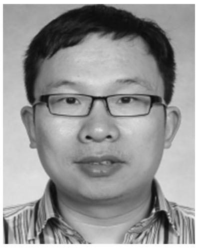


Yuanchao Shu received the PhD degree from Zhejiang University in 2015. From 2013 to 2015, he was a joint PhD student in computer science at the University of Michigan, Ann Arbor. He is currently an associate researcher at Microsoft Research Asia. He is the author and coauthor of more than 10 papers in premier journals and conferences, including the *IEEE Journal on Selected Areas in Communications*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Parallel and Distributed Systems*, ACM MobiCom, IEEE INFOCOM, and is the recipient of the IBM PhD fellowship and the INFOCOM'14 Best Demo Award. His research interests include cyber-physical systems, mobile computing, wireless sensor network, and big data analysis. He is a member of ACM and the IEEE.

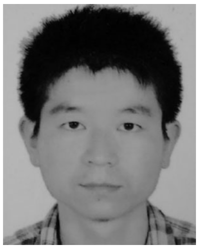
Yu (Jason) Gu received the PhD degree from the University of Minnesota, Twin Cities, in 2010. He is currently a research staff member at IBM Watson Health. He is the author and coauthor of more than 100 papers in premier journals and conferences. His publications have been selected as graduate-level course materials by more than 20 universities in the United States and other countries. His research includes networked embedded systems, wireless sensor networks, cyber-physical systems, wireless networking, real-time and embedded systems, distributed systems, vehicular ad-hoc networks, and stream computing systems. He is a member of ACM and the IEEE.



Shuo Guo received the BS degree in electronics engineering from Tsinghua University in 2006 and the PhD degree in electrical engineering from the University of Minnesota, Twin Cities, in 2011. She received a Best Paper Award at IEEE MASS 2008, and has authored and co-authored more than 20 papers in premier journals and conferences. These publications have received more than 500 citations worldwide. She is now a software engineer with Arista Networks.



Tian He received the PhD degree under Professor John A. Stankovic from the University of Virginia, Virginia, in 2004. He is currently an associate professor in the Department of Computer Science and Engineering at the University of Minnesota-Twin City. He is the author and co-author of more than 190 papers in premier network journals and conferences with more than 15,000 citations (H-Index 48). His publications have been selected as graduate-level course materials by more than 50 universities in the United States and other countries. He has received a number of research awards in the area of networking, including five best paper awards. He also received the US National Science Foundation (NSF) CAREER Award 2009 and McKnight Land-Grant Professorship. He served a few program chair positions in international conferences and on many program committees, and also currently serves as an editorial board member for six international journals. His research includes wireless sensor networks, cyber-physical systems, intelligent transportation systems, real-time embedded systems, and distributed systems, supported by US NSF, IBM, Microsoft, and other agencies. He is a senior member of the IEEE.



Fang Zhang received the BSc degree in control science and engineering from Chu Kochen Honors College, Zhejiang University, in 2009 and the MSc degree in electrical engineering from the University of Notre Dame in 2013. His research interests include target tracking and sensor scheduling in wireless sensor networks. He is currently working at Google as a software engineer.

Jiming Chen received the BSc and PhD degrees both in control science and engineering from Zhejiang University in 2000 and 2005, respectively. He was a visiting researcher at INRIA in 2006, National University of Singapore in 2007, and the University of Waterloo from 2008 to 2010. Currently, he is a full professor with the Department of Control Science and Engineering, and a vice director of the State Key laboratory of Industrial Control Technology and the Institute of Industrial Process Control at Zhejiang University, China. He currently serves as an associate editor for several international journals including the *IEEE Transactions on Parallel and Distributed System*, *IEEE Network*, *IEEE Transactions on Control of Network Systems*, *etc.* He was a guest editor of the *IEEE Transactions on Automatic Control*, etc. His research interests include sensor networks and networked control. He is a senior member of the IEEE.