EQS: Neighbor Discovery and Rendezvous Maintenance with Extended Quorum System for Mobile Sensing Applications

Desheng Zhang and Tian He
University of Minnesota, USA
{zhang,tianhe}@cs.umn.edu

Fan Ye, Raghu k. Ganti and Hui Lei
IBM T. J. Watson Research Center, USA
{fanye, rganti, hlei}@us.ibm.com

Abstract—In many mobile sensing applications devices need to discover new neighbors and maintain the rendezvous with known neighbors continuously. Due to the limited energy supply, these devices have to cycle their radios to conserve energy, making neighbor discovery and rendezvous maintenance even more challenging. To date, the main mechanism for device discover and rendezvous maintenance in existing solutions is pairwise, direct one-hop communication. We argue that such pairwise direct communication is sufficient but not necessary: there exist unnecessary active slots that can be eliminated, without affecting discovery and rendezvous. In this work, we propose a novel concept of *extended quorum system*, which leverages *indirect* discovery to further conserve energy. Specifically, we use *quorum graph* to capture all possible information flow paths where knowledge about known-neighbors can propagate among devices. By eliminating redundant paths, we can reduce the number of active slots significantly. Since a quorum graph can characterize arbitrary active schedules of mobile devices, our work can be broadly used to improve many existing quorum-based discovery and rendezvous solutions. The simulation and testbed experimental results show that our solution can reduce as much as 55% energy consumption with a maximal 5% increase in latency for existing solutions.

I. INTRODUCTION

There has been a consistent rise of mobile sensing applications where devices equipped with various sensors interact with each other upon encounters [1] [2] [3] [4] [5] [6] [7]. These applications rely on *neighbor discovery* and *rendezvous maintenance*, where new neighbors should be detected timely and contacts with existing neighbors are maintained continuously. Due to the limited battery capacity, these devices usually adopt duty cycling mechanisms [8] that switch their radios between active and inactive slots to conserve energy. Such an energy constrained environment makes it challenging to discover a device's previous unknown neighbors and maintain rendezvous with already discovered ones.

In the past, several neighbor discovery and rendezvous protocols for sensor networks [9] [10] [11] [12] [13] [14] [15] have been proposed. However, they all use a pairwise discovery mechanism via direct one-hop communication. We argue that such direct discovery is not always necessary. Devices can leverage the knowledge of each other, such that neighbors unknown to some devices can be discovered through other devices. Further, most of them [13] [14] [15] treat rendezvous maintenance as a rediscovery problem, even though a large number of neighbors have already been discovered. Both of

these properties lead to unnecessary active slots that can be eliminated to further conserve energy.

Based on these observations, we have developed extended quorum system, which relies on a novel mathematical concept called *quorum graph* to achieve efficient neighbor discovery and rendezvous maintenance. Extended quorum system propagates known neighborhood information indirectly by bridging multiple pairwise communication, thus avoiding the need for full-mesh pairwise discovery. We characterize the information propagation paths of known neighbors using the quorum graph, and propose a reduction algorithm that eliminates redundant paths to reduce the number of active slots. This significantly improves energy efficiency of discovery process.

Specifically, we make the following contributions:

- We propose a novel concept of *quorum graph*, and prove that its property of *reachability* is equivalent to neighborhood information flow among devices. Then we propose *extended quorum system*, which is a quorum graph satisfying indirect reachability among all devices, such that fewer active slots are required.
- Since redundant reachability leads to energy waste, we formalize the *quorum reachability minimization problem*, and prove its NP-hardness by reducing the set covering problem to one of its simplified version.
- To address the quorum reachability minimization problem in practice, we design a heuristic algorithm EQS that reduces redundant pairwise reachability between discovered and undiscovered neighbors. It supports device discovery and rendezvous maintenance with a significantly reduced number of active slots (*i.e.*, less energy).
- We evaluate EQS by larger scale simulations and indoor testbed experiment. The results show that EQS can reduce as much as 55% energy consumption with a maximal 5% increase on latency for existing protocols.

The rest of the paper is organized as follows: Section II presents our motivations and design goal. Section III describes theoretical concepts for quorum graph and extended quorum system. Section IV formalizes the quorum reachability minimization problem, followed by its solution EQS in Section V. Section VI and VII evaluate EQS based on simulation and testbed experiments. Section VIII discusses related work. Finally, Section IX concludes the paper.

Global Time Slot	0	1	2	3	4	5	6	7	8	9	10
S	-	0	1	2	3	4	5	6	7	8	9
A	0	1	2	3	4	5	6	7	8	9	10
B	0	1	2	3	4	5	6	7	8	9	10

	Inactive Slots		Active Slots

Fig. 1.   Examples for Motivations

II. Motivations and Design Goals

A. Motivations

Our work is mainly motivated by the observation that the current discovery and rendezvous protocols suffer from unnecessary active slots due to direct pairwise communication; this is especially inefficient when some neighbors have already been discovered. On the other hand, it is desirable to utilize as few active slots as possible to conserve the energy. Therefore, we explore how to leverage indirect communications for better energy efficiency.

In many scenarios, neighboring devices share common neighbors. This opens the possibility of indirect interaction for discovery and rendezvous. For discovery of new neighbors, if we let devices broadcast their two-hop neighbor tables in active slots, a device can *indirectly* discover its new neighbors via the *direct* rendezvous with discovered neighbors. For rendezvous with already discovered neighbors, *i.e.*, obtaining the neighbor tables from them, a device S can indirectly rendezvous with another device B by the direct rendezvous with a intermediate device A, given A and B have already rendezvoused. This indirect rendezvous between S to B is achieved by the fact that A will pass the neighbor table of B to S. Taking this indirect discovery and rendezvous into consideration, numerous active slots for direct discovery and rendezvous can be reduced.

Fig. 1 gives an example about indirect discovery and rendezvous. Three devices S, A and B have already discovered each other. Based on the state-of-the-art discovery protocol Disco [13], a schedule is given in Fig. 1, where S, A and B asynchronously begin their rendezvous cycles in global time slots 1, 0 and 0, respectively. Based on it, S will rendezvous with its already discovered neighbor A and B in global slots 4 and 7, respectively, via *direct* communications with each of them (two devices being active at the same slot indicates a rendezvous). In addition, S will continue to directly discover new neighbors in slot 1, 4, 7 and 10.

However, we can see that in global time slot 0, A has already rendezvoused with B, by exchanging neighbor tables. Therefore, when A and S rendezvous in global time slot 4, A's neighbor table containing the neighbors of B will propagate to S. Therefore, even if S does not activate its radio in global slot 7 to *directly* rendezvous with B, S's rendezvous with B can still be achieved *indirectly* through two other rendezvous (*i.e.*, the rendezvous between B and A, and between A and S). This is an example how neighborhood information propagates indirectly by bridging multiple pairwise communications, which opens the possibility to eliminate
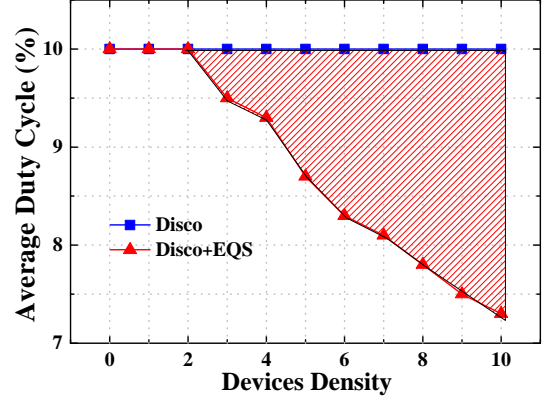

Fig. 2.   Illustration of Disco and Disco+EQS

the need for fully meshed pairwise discovery and rendezvous. Since above schedule is known to S after first time S discovers A and B, S can eliminate some of its active slots yet still achieving the indirectly rendezvous.

How to (1) capture above indirect nature of neighborhood information propagation and (2) leverage this nature to adaptively filter out redundant active slots to conserve energy are primary motivations for this paper. In Section III, we propose extended quorum system to capture this indirect propagation, and in Section IV, we propose quorum reachability minimization problem about filtering out these redundant active slots.

B. Design Goals

To illustrate limitations of existing protocols on energy performance and our design goals, we plot our testbed results for Disco [13] and its EQS assisting version, Disco+EQS, in Fig. 2. Under the networks with average duty cycle 10%, as the device density increases, the gap on average duty cycle between Disco and Disco+EQS is increasingly enlarged. When 4 devices are in the networks, EQS can reduce 7% of total active slots in the networks, *i.e.*, 9.3% vs. 10%. When the number of devices becomes larger, *e.g.*, 10, EQS is capable of reducing 27% of total active slots, *i.e.*, 7.3% vs. 10%. This energy gain of Disco+EQS comes from the reduction of active slots based on indirect discovery and rendezvous. It indicates that with more devices in the networks, EQS can reduce more active slots based on the enriched indirect information propagation among devices. Therefore, Fig. 2 illustrates the design goal of EQS, which filters out redundant active slots to conserve energy, *i.e.*, enlarging the dashed area between two lines.

Given the existence of a plethora of discovery and rendezvous protocols based on quorum system, in Section V we decide to design EQS as a transparent augmenting middleware filter on bottom of them. Given any schedule based on these protocols, EQS can transparently filter out the redundant active slots according to indirect discovery and rendezvous. It provides a unified solution for highly diverse and heterogeneous discovery and rendezvous protocols that may be

deployed at individual mobile devices from various mobile applications. Based on this transparent and non-intrusive design philosophy, we do not consider to synchronize devices by adding common active slots. In fact, we consider how to *deactivate* the slots supposed to be active, not *vise versa*. Note that the deactivation of certain active slots may lead to the fact that a new neighbor cannot be discovered by some devices in the networks immediately. But with rendezvous services, the devices already discovering each others will function as a group, and a device finding a new neighbor will lead to a quick propagation of neighborhood information to others.

III. QUORUM SYSTEM

In this section, we introduce the theoretical concepts for neighbor discovery and rendezvous maintenance. We first present a concept, *quorum graph*, and then give a propriety of quorum graph, *reachability*, in subsection *A*. Based on two different reachabilities, *direct* and *indirect*, we propose two kinds of models for neighbor discovery and rendezvous, *legacy quorum system* and *extended quorum system*, in subsection *B* and *C*, respectively.

A. Quorum Graph

1) Definition of Quorum Graph: Quorum graph is a graph representation $G(V, E)$ (*e.g.*, Fig. 4) of a neighbor discovery and rendezvous maintenance schedule (*e.g.*, Fig. 3). We present formal definition of quorum graph as follows.

Definition 1: Quorum Graph. Given a Discovery and Rendezvous Schedule DRS for devices, a corresponding *quorum graph* QG is a supergraph consisting of several non-overlapping subgraphs of vertices, which is characterized by five following key features.

1) **Vertex**: every active slot of a device in DRS can be represented as a vertex;
2) **Subgraph**: all the vertices corresponding to all active slots for the same device can be represented as a subgraph, called a quorum;
3) **Supergraph**: all the subgraphs corresponding to all the devices can be represented as a supergraph, *i.e.*, QG;
4) **Horizontal Edge**: If two vertices correspond to the same active slot in two different subgraphs, then a *bidirectional* horizontal edge exists between them, which indicates that in this slot two devices corresponding to two subgraphs can *bidirectionally* exchange neighborhood information;
5) **Vertical Edge:** If two vertices correspond to the different active slots in the same subgraph, then a *unidirectional* vertical edge exists between them from the early slot to the later slot, which indicates that this device corresponding to the subgraph can only *unidirectionally* pass neighborhood information it has learnt in the early slot to the later slot, not *vise versa*, so vertical dashed edges are top-down only.

As a result, in Fig. 4, (1) a row of vertices represents the duplicated copies of the same active slot for different devices; (2) a column of vertices represents the different active slots for
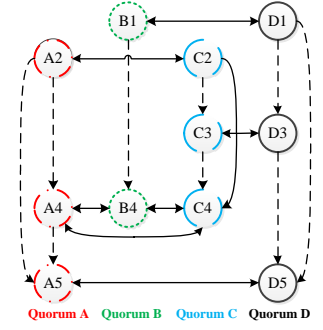

Fig. 3. Schedule for Discovery

Fig. 4. Quorum Graph

the same device; (3) the edges between two vertices represent the links for neighborhood information propagation through networks along with time dimension.

To illustrate how to construct a quorum graph based on a schedule, in Fig. 3 and Fig. 4, we provide a walk-through of quorum graph construction. (1) Building 4 subgraphs (quorums) based on 4 devices in the schedule. (2) According to number of active slots in every devices, building the same number of vertices in every subgraph, *e.g.*, we build three corresponding vertices, *i.e.*, $A2$, $A4$ and $A5$ in a subgraph A for device A. (3) Building unidirectional edges within subgraphs from the vertices associated to early slots to the vertices associated to later slots, *e.g.*, we build two unidirectional edges from vertex $A2$ to $A4$ and $A5$, respectively. (4) Building bidirectional edges for the vertices in different subgraphs but associated to the same slots, *e.g.*, we build a bidirectional edge from vertex $A2$ to $C2$, since these two vertices are associated to the same slots, slot 2.

The rationale behind quorum graph constriction is that with this quorum graph we can capture the neighborhood information propagation among the devices along time dimension. For bidirectional edges, in Fig. 4, device A and C will rendezvous with each other in a bidirectional way in slots 2, which is the reason why a bidirectional edge exist between $A2$ to $C2$. For unidirectional edges, in Fig. 4, when device C and D rendezvous with each other in slots 3, C will pass the information about A to D, but C cannot pass the information about D to A in slot 2, since C will rendezvous with D after slot 2, *i.e.*, slot 3. This is the reason why in Fig. 4 the edge between $C2$ to $C3$ are unidirectional. By above example, a quorum graph represents a high level abstraction about how neighbor information propagates among the devices based on their rendezvous along the time dimension.

2) Reachability: Based on the quorum graph obtained in last subsection, we present a key property of quorum graph, *i.e.*, *reachability*. In traditional Graph Theory [16], reachability is the notion of being able to obtain from one vertex in a directed graph to some other vertices. However, under quorum graph context, we employ reachability to describe the capability of subgraphs, *instead of vertices*, to reach each other in a given quorum graph.

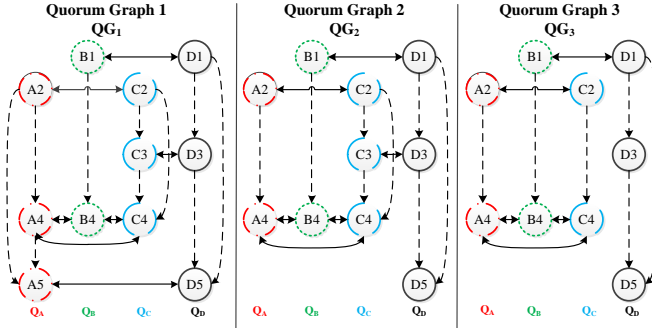**Property 1: Reachability.** In a quorum graph, if any quo-

Fig. 5. Examples of Quorum Graph

rum can reach at least one vertex in every other quorum, then this quorum graph has *reachability*. Further, if any quorum can reach every other quorum only by its own vertices, instead of the vertices of other quorums, then this quorum graph has *direct reachability*; otherwise, it has *indirect reachability*.

For example, Fig. 5 shows three different quorum graphs, *i.e.*, QG_1, QG_2 and QG_3. **(1)** QG_1 has direct reachability, since in QG_1, every quorum has at least one vertex that can directly, with *one-hop*, reach all other quorums by the vertices in its own quorum. **(2)** QG_2 has indirect reachability, since in QG_2, some quorums have to leverage other quorums' vertices for reachability, *e.g.*, vertex $A2$ in quorum Q_A can reach vertex $D3$ in quorum Q_D with the vertices in quorum Q_C, *i.e.*, vertex $C2$ and $C3$. This is the only path that Q_A can reach Q_D. **(3)** QG_3 has no reachability, since in QG_3, no vertex in Q_C can reach Q_D.

Considering the neighbor discovery and rendezvous context, the reachability of a quorum graph indicates how neighborhood information about one device can propagate to other devices, directly by itself or indirectly by others.

B. Legacy Quorum System

In this subsection, via quorum graph, we introduce a theoretical model, *Legacy Quorum System*, used by current neighbor discovery and rendezvous protocols [9] [10] [11] [12] [13] [14].

Definition 2: Legacy Quorum System. A legacy quorum system LQ is a quorum graph with *direct reachability* among any pair of quorums.

In Definition 2, under neighbor discovery context, this direct reachability among any pair of quorums indicates the direct discovery or rendezvous for every two devices. Thus, Definition 2 presents a model used by current protocols for pairwise and direct discovery and rendezvous. For example, in Fig. 5, among three different quorum graphs, QG_1 is a legacy quorum system, since QG_1 has direct reachability, while QG_2 and QG_3 are not. Under our context, Definition 2 can be employed to verify a given schedule whether or not can lead to pairwise and direct discovery and rendezvous between devices.

C. Extended Quorum System

In this subsection, based on quorum graph and legacy quorum system, we define a new kind of model for neighbor

discovery and rendezvous with the indirect nature. Our motivation about this new kind of model is based on the fact that in legacy quorum system, each quorum reaches other quorums directly by its own vertices, and no intermediate quorum is involved to assist the reachability among quorums in an indirect way. In Fig. 5, legacy quorum system only makes use of the *horizontal solid edges*, and our new model tries to employ the *vertical dashed edges* to achieve more *diverse reachabilities* in the same quorum graph. Similar to Definition 2, we give definition of this new model by quorum graph.

Definition 3: Extended Quorum System. A extended quorum system EQ is a quorum graph with *indirect reachability* among any pair of quorums.

In Definition 3, under neighbor discovery context, the indirect reachability indicates the indirect interaction between devices. Thus, Definition 3 presents a model for indirect discovery and rendezvous. For example, in Fig. 5, QG_2 is an extended quorum system, since QG_2 has indirect reachability; while QG_1 and QG_3 are not. Under our context, Definition 3 can be employed to verify a given schedule whether or not can lead to indirect discovery and rendezvous between devices.

IV. PROBLEM FORMULATION

In this section, we first provide network model and assumptions. Then, we formalize *quorum reachability minimization problem* for neighbor discovery and rendezvous maintenance.

A. Network Model and Assumptions

For the networks of static devices with always-available radio, discovery and rendezvous are trivial, since simple broadcasts can enable all neighbors to discover a device [9]. However, for the networks of mobile devices with constrained radio usage, discovery and rendezvous become complicated [13]. Time synchronization will be greatly helpful, but involves considerable and unaffordable cost [15]. Some related work [9] [10] [11] [12] [13] [14] [15] has been proposed to employ legacy quorum system to tackle discovery and rendezvous for asynchronous mobile networks with duty cycled devices.

Since our design should be compatible with current discovery and rendezvous, we present the similar network model and assumptions [13] [14] [15]. **(1)** In single-hop networks, time is divided into slots with equal lengths. **(2)** The radios of devices in networks are activated in certain active slots according to a given schedule based on legacy quorum system. **(3)** In both beginning and end of an active slot, a device broadcasts its two-hop neighbor table. **(4)** An overlapping of active slots between devices indicates a discovery or rendezvous. Note that even with clock drift, since a device broadcasts twice in both beginning and end of an active slot, a partial overlapping on an active slot can still guarantee a successful bidirectional discovery [13]. **(5)** Every device distritbutedly collects and maintains neighborhood information.

B. Quorum Reachability Minimization Problem

We present *quorum reachability minimization problem* for neighbor discovery and rendezvous maintenance.

Definition 4: Quorum Reachability minimization Problem. Given a quorum graph representing *legacy quorum system*, *Quorum Reachability minimization Problem*, QRP, is to choose minimal number of vertices to maintain the reachabilities between every two quorums, forming a new quorum graph representing *extended quorum system.*

Under our context, by distributedly solving QRP, all devices already discovering each others will obtain the same schedule for the networks based on extended quorum system. Moreover, by distributedly filtering out the redundant active slots not in this new schedule, a better energy performance can be achieved.

Before presenting the solution for QRP, we shall evaluate its complexity. We have proved that QRP is NP-hard by reducing set covering problem to its simplified version. Due to the space limitation, see detailed proof in appendix.

V. EQS Design

Before proposing our main design, in subsection A, we present some intuitions to show what are differences between schedules based on legacy quorum system and extended quorum system in discovery and rendezvous. In subsection B, based on above intuitions, we propose our main design EQS, which a heuristic solution to a NP-hard quorum reachability minimization problem that obtains an extended quorum system from a legacy quorum system. Finally, we give an example about EQS.

A. Quorum System for Discovery and Rendezvous

Let A, B, C and D be four devices in the networks, whose schedules are given by a corresponding quorum graph QG_1 in Fig. 6. Based on Definition 2 in Section III, we can easily verify that QG_1 is a legacy quorum system, since a simple traversal will show that QG_1 has direct reachability. Based on the assumptions, it ensures all the devices can successfully and directly discover and rendezvous with others.

However, a new schedule QG_2 that is obtained by reducing some active slots in QG_1 based on extended quorum system can still achieve discovery and rendezvous, when some devices have already discovered each others. Let QG_2 in Fig. 6 be a new schedule. Based on Definition 3, we can easily conclude that QG_2 is an extended quorum system, since a simple traversal will show that QG_2 has indirect reachability. In QG_2, the neighborhood information about device B can be propagated to A via $A4$, to C via $C4$, and to D via $A5$ and $D5$, *i.e.*, via already discovered neighbors. There are similar situations to the information for A, C and D.

From above example in Fig. 6, we can see that the key difference between QG_1 and QG_2 is that QG_1 does not take discovered devices into consideration, since legacy quorum system only focuses on *pairwise* and *directly* discovery and rendezvous. However, when some neighbors have been already discovered, the schedules based on extended quorum system can make *indirectly* discovery and rendezvous with fewer active slots. For example, in QG_2, the information of B can be propagated to D *indirectly* via intermediate A. Whereas, if we
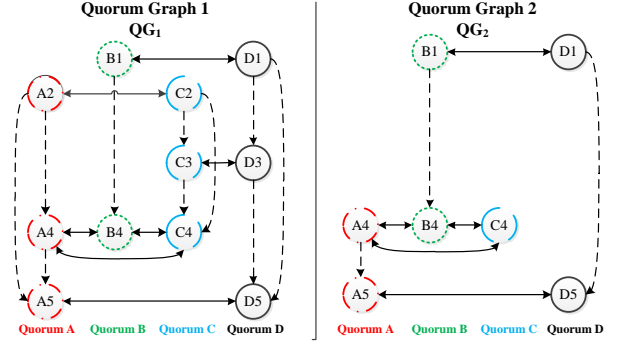

Fig. 6. Example of Different Quorum System

still use QG_1 as rendezvous schedule where no intermediate device is considered, then it will lead to redundant active slots. How to obtain a new schedule with minimal active slots based on an extended quorum system from a given discovery schedule based on legacy quorum system is our NP-hard *Quorum Reachability Minimization Problem* proposed in last Section. Therefore, in the next subsection we will present a heuristic solution to tackle this problem.

B. Main Design

In this subsection, given a schedule based on legacy quorum system, we propose a design, EQS, which outputs a filter vector, FV, by solving quorum reachability minimization problem. By FV, a device in the networks can filter out the unnecessary active slots for neighbor discovery and rendezvous maintenance.

The main idea of our heuristic scheme EQS is simple and based on two following observations. **(1)** Given a quorum graph, we have to select a new subgraph with the minimal number of vertices to maintain the reachabilities for every two quorums, and then reduce other unselected vertices. In our discovery scenario, given a schedule, we have to select minimal number of active slots to maintain reachabilities for every two devices in the networks, and reduce other redundant active slots to conserve energy, *i.e.*, given total N devices, total $N \times (N-1)$ reachabilities (every N device for every other $N-1$ devices) have to be maintained with minimal number of active slots. **(2)** Every time we shall select some active slots that should provide the *maximal contribution* to total $N \times (N-1)$ reachabilities for all the quorums, and the *minimal burden* to the reachabilities between themselves.

For the *minimal burden* to reachabilities, selecting a row of vertices together at a time will obtain minimal burden for reachabilities between themselves, since the vertices in the same row will always reach each others, *i.e.*, no extra effort should be made for the reachabilities of the vertices belonging to the same row.

For the *maximal contribution* to reachabilities, the contribution of a row x, C_x, is computed as follows.
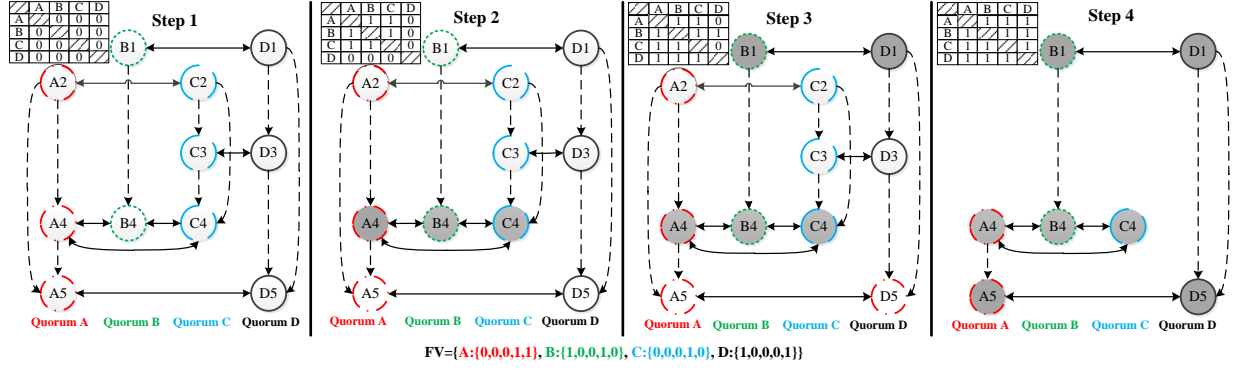
$$C_x = \frac{T_x}{N_x},$$

Fig. 7. Example of EQS

where T_x is the number of new provided reachabilities after selecting the vertices in row x; N_x is the number of the vertices in row x. By above formula, every time we select few *efficient* vertices to provide the *maximal* contributions to total reachabilities (maximal T_x) with the minimal number of active slots (minimal N_x).

Therefore, our scheme is that every time we select a row of vertices such that this row will contribute to maximal remaining portion of total $N \times (N-1)$ reachabilities, until already selected vertices can maintain all total $N \times (N-1)$ reachabilities. This is the key idea of our heuristic scheme. However, since we select a row of vertices as a whole, some vertices in this row may not contribute to the total $N \times (N-1)$ reachabilities. Therefore, after every selection of a row of vertices, we delete the vertices that do not contribute to the total $N \times (N-1)$ reachabilities.

After obtaining complete subgraph, EQS outputs a 0-1 filter vector FV where 1 indicates corresponding active slot remains and 0 indicates otherwise. By this FV and original neighbor discovery schedule, every device in the networks can maintain the discovery and rendezvous with fewer active slots. This FV is constantly updated by EQS according to the latest neighborhood information in a device's lifecycle.

Fig. 7 gives an example about EQS.

1) Based on a given schedule, we can obtain its corresponding quorum graph. The reachabilities after every step is shown in the above left corner table, where 0 indicates that the reachability from row to column is not maintained, and 1 indicates otherwise.

2) Based on its quorum graph, we compute the contribution per vertex of every row to the total $4 \times (4-1)$ ($N \times (N-1)$) reachabilities. For example, the contribution of first row C_1 is $\frac{1+1}{2} = 1$, since selecting vertex $B1$ and $D1$ to the subgraph only contributes two reachabilities, *i.e.*, from Q_B to Q_D as well as from Q_D to Q_B. By the same method, we compute that C_4 is local maximal in the first round and is $\frac{6}{3} = 2$. Therefore, we select 4th row of vertices to the subgraph by marking them to grey as in Step 2.

3) In the remaining quorum graph, C_1 and C_3 have the same maximal value, which is $\frac{4}{2} = 2$. We select 1st row

according to the alphabetical order.

4) After we select 1st row to subgraph, only two reachabilities need to be maintained shown by the table. 5th row has the local maximal contribution with $C_5 = \frac{2}{2} = 1$, so we select 5th row. After this, all reachabilities are maintained, and we complete the subgraph by reducing other unselected vertices. By changing the subgraph to its adjacent matrix, every device can distributedly choose its own column to obtain its FV. For example, device A will choose $\{0, 0, 0, 1, 1\}$ as its FV. Therefore, when upper-layer discovery and rendezvous protocol activates A in 2nd slot, FV of A will filter out this active slot and makes A maintain inactive.

Via above example, we can see that EQS takes the legacy quorum system based schedules as an input, and will output a FV for every devices, by converting this schedule to a new schedule based on extended quorum system. After obtaining this FV consisting of 0s and 1s, in every slots, a device will conduct logic intersection between FV and Schedule Vector SV, which is a schedule for device itself (If discovery and rendezvous protocol requires this slot to be active, then the corresponding bit on SV is 1, and *vise versa*). Therefore, only both corresponding bits on FV and SV are 1, then a device will activate itself in this slot. If the corresponding bit on FV is 0 and the corresponding bit on FV is 1, then FV filters this active slot out, since based on extended quorum system this slot is no longer necessary to be active.

VI. SIMULATION EVALUATION

To evaluate the effectiveness and flexibility of our EQS design, in this section we integrate EQS with two state-of-the-art discovery and rendezvous protocols:

- Disco [13] by Dutta *et al.* in SenSys'08.
- U-Connect [14] by Kandhalu *et al.* in IPSN'10.

To understand how much energy efficiency EQS can offer, we also compare EQS with a Baseline design, which filters out the same amount of active slots with EQS, but *at random*, instead of employing extended quorum system. Thus, we simulate three versions of above protocols, *i.e.*, original, Baseline and EQS.
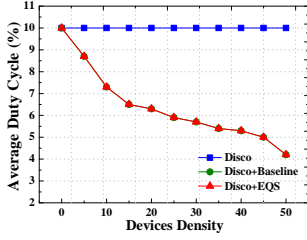
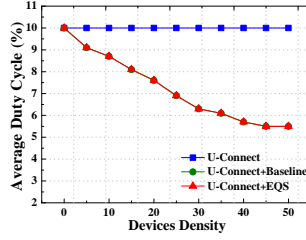Fig. 8.   Disco ADC vs. DD

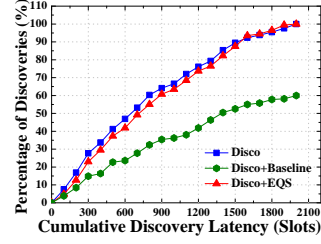Fig. 9.   U-Connect ADC vs. DD
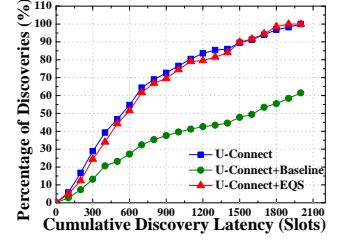

Fig. 10.   Disco Discovery CDF

Fig. 11.   U-Connect Discovery CDF

A. Simulation Setup

In our simulation, 100 mobile devices are uniformly deployed in a square area of size $200m \times 200m$. The transmission ranges of devices are set from $20m$ to $110m$, which leads to average mobile device densities from 3.6 to 55.36 neighbor devices. For the mobility model of mobile devices, we use the random waypoint model [17] [18] [19], with the average device velocity setting to be $1m/s$. Each simulation is repeated 20 times and the average results are reported.

Three groups of simulations are conducted. **(1)** To show the performance gain, the key metric energy consumption, represented by Average Duty Cycle (ADC) of devices in the networks, are evaluated with different Device Densities (DD). **(2)** The reduction of active slots for energy consumption in EQS may increase the Discovery Latency (DL). To verify the impact of the reduction of active slots on DL, we also show the CDF of discovery and rendezvous. **(3)** The impact of different Duty Cycles (DC) on energy consumption is also shown.

B. Impact of Device Density on Energy Consumption

In this subsection, we report the effectiveness of EQS to conserve energy for Disco and U-Connect under different device densities. The impact of device densities on energy consumption, represented by Average Duty Cycle (ADC) of all the devices in the network, is shown in Figure 8 and Figure 9. In both figures, as the device density increases, the average duty cycles of Disco and U-Connect keep the same, while the average duty cycles of others decrease. This is because for Disco and U-Connect, since they do not reduce any active slots, the average duty cycles of devices under them keep unchanged. Whereas both their Baseline versions and their EQS versions enable the devices under them to reduce their duty cycles. Since compared to EQS, Baseline reduces the same amount of active slots at random, the devices under both of them have the same average duty cycle. In Figure 8, when the device density is below 20, Disco+EQS can achieve a energy gain more than 35%. When the device density increases to 50, Disco+EQS achieves more than 55% energy gain over Disco by reducing more than half the total active slots. The similar observation is shown in Figure 9 with less yet still obvious energy gain about 45% when the device density is 50. Above observations indicate that EQS serves more effectively in the networks with more devices. The explanation for energy

improvement under the networks with more devices is that a larger number of devices enables neighborhood information propagation more diversely in the networks, which is leveraged by EQS to reduce more active slots to achieve a better energy performance.

C. Impact of Reduction of Active Slot

An potential drawback about the reduction of active slots in EQS is that it may increase discovery latency. To verify this impact, we plot the CDF of number of discovery for both of schemes in Figure 10 and Figure 11. In these figures, as more discovery time is allowed (increasing cumulative discovery latency), percentages of discoveries also increase for all schemes. However as in Figure 10, the devices under Disco and Disco+EQS are able to discover neighbors much faster than under Disco+Baseline. This is because that Disco+Baseline only reduces active slots at random, which leads to a maximal 40% lower performance. As for Disco and Disco+EQS, even though Disco+EQS reduces some of active slots to conserve energy, the reduction of EQS is based on extended quorum system, not at random. Therefore, under Disco+EQS, neighborhood information propagation assists devices to find their new neighbors based on already discovered ones. In fact, this neighborhood information propagation enables some devices to discover each other earlier under Disco+EQS than under Disco itself, which in part compensates for reductions of active slots. For example, in Figure 10, in the end of discovery process, EQS enables devices to make a 4% faster discovery than the devices under Disco. This is because with the increase of discovery latency, the cumulative effect of neighborhood information propagation becomes more obvious, which leads to an more effective discovery. The similar phenomenon is observed in Figure 11. The devices under U-Connect+Baseline only discover 61% of their neighbors on average, when the devices under U-Connect and U-Connect+EQS have discovered all their neighbors. U-Connect outperforms U-Connect+EQS by 5%, when slots are fewer than 1500. After that the devices under U-Connect+EQS make a 3% faster discovery than those under U-Connect.

D. Impact of Device Duty Cycle on Energy Consumption

In this subsection, we investigate the impact of device's duty cycle on the energy consumption, which is also shown
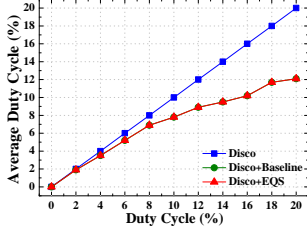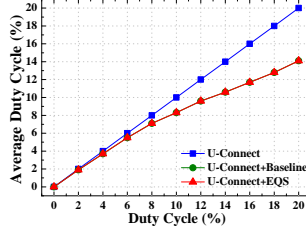
Fig. 12. Disco ADC vs. DC
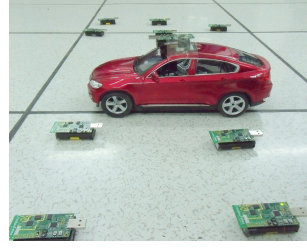

Fig. 13. U-Connect ADC vs. DC
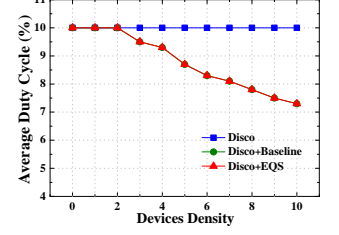

Fig. 14. Testbed Setup

Fig. 15. Testbed ADC vs. DD

by the average duty cycles of all the devices in the networks. As shown in Figure 12 and Figure 13, we can see that with the increase of the duty cycle, the average duty cycles of all schemes increase. Since no reduction of active slots is preformed in Disco and U-Connect, in both figures the increase of average duty cycles of Disco and U-Connect is steady. Because reductions in number of active slots in Baseline and EQS are the same, the curves of them overlap with each other all the time. In Figure 12, we can see that when the duty cycle is 8%, the average duty cycle of devices under Disco+EQS obviously outperforms that under Disco with a performance gain of 13.7%. As the duty cycle increases to 20%, this performance gain is also enlarged to as much as 40%. In Figure 13, even though not as much as it outperforming Disco, EQS still has a maximal 31% energy performance gain over U-Connect. The reason why in both figures the performance gain of EQS increases with the duty cycle is that more active slots indicate more opportunities for devices to share their neighborhood information. In addition, a higher duty cycle may indicate that more active slots can be reduced.

VII. Testbed Evaluation

In Section VI, we have shown that EQS effectively reduces energy consumption for discovery and rendezvous with extended quorum system. To evaluate the performance of EQS in a real world setting, we have implemented EQS on the TinyOS/Mote platform [20]. During the testbed experiments, we deploy 10 TelosB sensor devices and utilize a mobile toy car attached with a TelosB as a mobile device. The testbed setup is shown in Figure 14. All experiments are repeated 10 times and the average results are reported. At individual sensor devices, we set the duration of one time slot to be $25ms$. Due to the conceptual similarity between Disco and U-Connect, as well as the results from simulation, we only implement Disco in our testbed.

A. Impact of Device Density on Energy Consumption

In this subsection, we report the effectiveness of EQS for energy conservation in testbed experiment. Figure 15 shows the impact of the device density on energy consumption which is represented by average duty cycle. As the increase of the device density, the average duty cycle of Disco keeps the same and those of Disco+Baseline and Disco+EQS decrease and overlap with each other, which is due to the same reason in
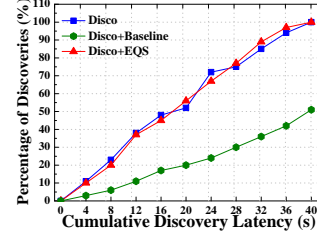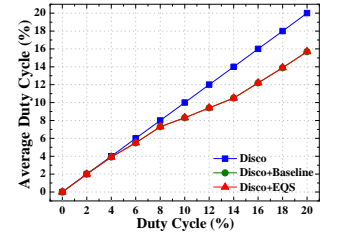

Fig. 16. Testbed Discovery CDF

Fig. 17. Testbed ADC vs. DC

Figure 8. However, in Figure 15, we observe that when few devices in the networks, $e.g.$, 2, there is no reduction of average duty cycle under Disco+EQS. However, when the number of devices increases, the performance gain is enlarged, shown by reduced average duty cycle as much as 27%. This is because that few devices cannot form an extended quorum system for the reduction of active slots, and when number of devices becomes bigger, an extended quorum system can be formed to reduce the redundant active slots.

B. Impact of Reduction of Active Slot

Figure 16 plots the testbed experiment results on the CDF of discovery. From this figure, we can see that Disco+EQS continues to exhibit a similar performance, compared with Disco itself. However, in the testbed experiment, Disco+EQS outperforms Disco in the later half of discovery process by 6%. The similar results are observed in Figure 10, but the performance gain is smaller. Again, as observed in Figure 10, Disco+Baseline has a much worse performance compared with others, and the devices under Disco+Baseline only discover 51% of their neighbors, when the devices under Disco+EQS and Disco have discovered all their neighbors.

C. Impact of Duty Cycle on Energy Consumption

Figure 17 shows the impact of different duty cycles on energy consumption, shown by average duty cycle. In testbed experiment, when the duty cycle is low, $e.g.$, 4%, no reduction is preformed by EQS since low device density in testbed and low duty cycle lead to fewer total active slots which cannot form an extended quorum system. However, when the duty cycle becomes bigger, $e.g.$, 10%, EQS can reduce 17% active slots according to extended quorum system. When the duty cycle becomes 20%, EQS achieves the maximal

energy performance gain, *i.e.*, 21.5%. This maximal energy performance gain is smaller than we obtained in large-scale simulation. This is because that in simulation EQS is under much larger and denser networks than in testbed. This may indicate that EQS is more suitable for large-scale networks.

VIII. RELATED WORK

Neighbor discovery and rendezvous maintenance in low-power mobile networks has been extensively studied in the literature. In general, the existing discovery and rendezvous schemes can be divided into two categories, *explicitly* and *implicitly* quorum system based.

A. Explicitly Quorum System based Schemes

To address discovery and rendezvous problem with a bounded worst-case latency, explicitly quorum system based protocols ensure the existence of overlapped active slots between any pair of devices within a bounded time. In these protocols, time is normally divided into $m \times m$ continuous slots as a matrix and each device selects one row and one column to activate its radio. Consequently, regardless which row and column a device chooses, it is guaranteed to have at least two common active time slots with other devices. The main drawback for this kind of protocols is the global parameter of m, which forces all devices to have the same duty cycle [9]. To support asymmetric duty-cycle setting, Zheng *et al.* [10] apply optimal block designs using difference sets for discovery and rendezvous in bounded latency. Based on their methods, discovery and rendezvous problem in asymmetric duty-cycle setting reduces to an NP-complete minimum vertex cover problem requiring a centralized solution [10]. More recently, Lai *et al.* present CQS-pair [11] and GQS-pair [12], which support heterogeneous quorum-based systems where devices can have different cycle lengths and hence different duty-cycle settings. However, only two kinds of duty cycles are supported.

B. Implicitly Quorum System based Schemes

Implicitly quorum system based protocols are also referred as *deterministic* protocols, which are proposed recently to handle the asynchronous neighbor discovery problem in mobile wireless networks [13] [14] [15]. These protocols select one or multiple prime numbers for every device to represent their duty cycles. Based on Chinese Remainder Theorem [21] [22], these devices would have bounded discovery and rendezvous latency based their chosen duty cycles. These protocols implicitly employ the idea of quorum system to enable every two of devices in the networks have at least one common active slots for each other. In Disco [13], each device selects a pair of prime numbers and generates its period independently based on the requirement of duty cycles. To improve the performance of Disco, U-Connect [14] is proposed as a unified discovery and rendezvous protocol for symmetric and asymmetric duty cycle settings. U-Connect achieves higher performance compared to Disco and Quorum-based protocol, especially in asynchronous symmetric case. More recently, to improve U-Connect, WiFlock [15] combines discovery and maintenance

using a collaborative beaconing mechanism with a temporal time synchronization. However, these deterministic protocols do not consider the neighborhood information propagation among the devices in the networks, which leads to redundant active slots in the networks. Our EQS can serve as a augmenting middleware for all above discovery and rendezvous protocols to conserve more energy.

IX. CONCLUSION

In this paper, we introduce EQS, an augmenting layer to conserve energy for existing neighbor discovery and rendezvous maintenance schemes that use pairwise direct communication. Our work is mainly motivated by the insight that when devices share common neighbors, they can leverage the knowledge of each other to detect such neighbors indirectly. Thus fewer active slots are needed and energy is conserved, especially when a device needs to maintain rendezvous with previously discovered neighbors. To capture such information sharing among devices theoretically, we propose a novel extended quorum system concept where information flow paths are equivalent to graph reachability. We then propose a graph reduction algorithm EQS that filters out redundant paths but still maintains graph reachability. We have integrated our EQS design with two discovery and rendezvous protocols, and evaluated its performance with both simulations and testbed experiments. The evaluation results show that EQS can effectively filter out redundant active slots to conserve as much as 55% energy with a maximal 5% increase on latency.

X. ACKNOWLEDGEMENTS

APPENDIX
COMPLEXITY OF QUORUM REACHABILITY MINIMIZATION PROBLEM

In this appendix, we shall evaluate the complexity of QRP. However, it is hard to directly evaluate the complexity of QRP, due to its particularity under discovery and rendezvous context. Alternatively, we first introduce a simplified version of QRP, and evaluate its complexity. Then, based on its complexity, we evaluate the complexity of QRP.

Definition 5: Simplified Quorum Reachability minimization Problem. Simplified Quorum Reachability minimization Problem, $SQRP$, is a QRP under two simplified conditions. (1) Converting all unidirectional edges to bidirectional edges. (2) Selecting vertices only by the unit of rows.

For two simplified conditions, (1) the bidirectional edges in quorum graph increase the reachability of a quorum graph, making $SQRP$ simpler than QRP; (2) selecting vertices only by the unit of rows reduces the number of possible selections for vertices in quorum graph, making $SQRP$ simpler than

QRP as well. The rationale behind these two simplified conditions is to facilitate the reduction from classic set covering problem to our problem, which is introduced in following Lemma.

Lemma 1: $SQRP$ is NP-hard.

Proof: We prove $SQRP$ is NP-hard by reducing set covering problem to $SQRP$. Given a collection of weighted subsets, $S_1...S_k$, of a given full set F, set covering problem, SCP, is to seek a minimum-weighted collection of subsets whose union is F [16]. We reduce SCP to $SQRP$ by following steps. (1) Building a star graph with a central node F and a leaf for every subset S_i. (2) Expanding each leaf node by adding subleaf nodes, and the number of subleaf nodes equals the number of subset's elements, and all the edges of subleaf nodes have weights 0. (3) Every element e in F defines a set Q_e of subleafs such that the subleafs containing e are belonged to Q_e, and all leaf nodes also define a set Q_S. (4) Giving the weight on a edge from a subset node to center node by the number of elements in this subset. Via the above constructions, we can see that a solution to SCP (*i.e.*, finding a minimal weighted collection of subsets such that the union of chosen subsets can cover full set F) is a minimal weighted quorum graph with indirect reachability, after (1) reducing node F; (2) adding bidirectional edges by definition of quorum graph; (3) considering all Q_es and Q_S as quorums. Via Definition 3, this minimal weighted quorum graph represents extended quorum system, which is a solution to $SQRP$ (*i.e.*, finding a minimal weighted extended quorum system based quorum graph). Therefore, we prove that $SQRP$ is NP-hard.

Given a full set and 5 subsets in Fig. 18, an example is shown. (1) Based on 5 subsets, we build a star graph with a center node F and 5 leafs for every subset S_i. (2) Expanding leaf node S_1 by two subleaf nodes B_1 and D_1, and so do other leaf nodes. All weights on edges from a leaf to subleaf are 0. (3) Element A defines a set $Q_A = \{A2, A4, A5\}$, so do other elements in F. (4) The weight on the edge from node S_1 to node F equals the number of elements in S_1, *i.e.*, 2, so do other weights. The final result is shown by left figure in Fig. 18. Via middle figure, we reorganized left figure to right figure (we omit vertical edges for clearness). Under simplified conditions, a solution to SCP (*e.g.*, S_1 and S_2) is a solution to $SQRP$, finding a minimal weighted quorum graph with indirect reachability, and *vise versa*.

Based on Lemma 1, we present following Theorem 1 about the complexity of QRP.

Theorem 1: QRP is NP-hard.

Proof: Since $SQRP$ is a simplified version of QRP, by Lemma 1, we indirectly prove that QRP is NP-hard.

Fig. 18.   Reduction from Set Covering Problem to $SQRP$

REFERENCES

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[2] N. Lane *et al.*, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 140–150, 2010.
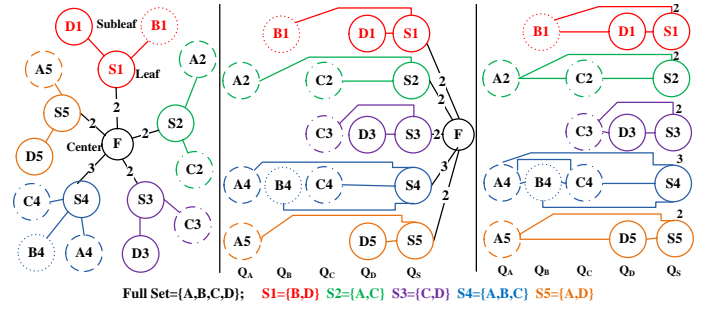
[3] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T.Campbell, "The bikenet mobile sensing system for cyclist experience mapping," in *In Proceedings of the 5th ACM conference on Embedded network sensor systems (Sensys 2007)*, 2007.

[4] H. Liu, J. Li, Z. Xie, S. Lin, K. Whitehouse, J. A. Stankovic, and D. Siu, "Automatic and robust breadcrumb system deployment for indoor firefighter applications," in *MobiSys*, 2010.

[5] P. Lukowicz, T. Choudhury, and H. Gellersen, "Beyond context awareness," *Pervasive Computing, IEEE*, vol. 10, no. 4, pp. 15 –17, april 2011.

[6] J.-H. Huang, S. Amjad, and S. Mishra, "Cenwits: a sensor-based loosely coupled search and rescue system using witnesses," in *SenSys'05*, 2005.

[7] 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. of ASPLOS-X*, October 2002.

[8] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09, 2009.

[9] Y.-C. Tseng, C.-S. Hsu, and T.-Y. Hsieh, "Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks," in *INFOCOM'02*, 2002.

[10] R. Zheng, J. C. Hou, and L. Sha, "Asynchronous wakeup for ad hoc networks," in *MobiHoc'03*, 2003.

[11] S. Lai, B. Zhang, B. Ravindran, and H. Cho, "Cqs-pair: Cyclic quorum system pair for wakeup scheduling in wireless sensor networks," *Principles of Distributed Systems, LNCS*, 2008.

[12] S. Lai, B. Ravindran, and H. Cho, "Heterogenous quorum-based wake-up scheduling in wireless sensor networks," *Computers, IEEE Transactions*, 2010.

[13] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *SenSys '08*, 2008.

[14] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol," in *IPSN'10*, 2010.

[15] A. Purohit, N. Priyantha, and J. Liu, "Wiflock: Collaborative group discovery and maintenance in mobile sensor networks," in *IPSN'11*, 2011.

[16] M. Mitzenmacher and U. Upfal, "Probabilitty and computing," 2007.

[17] 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.*, 2007.

[18] E. Hyytia, P. Lassila, and J. Virtamo, "A markovian waypoint mobility model with application to hotspot modeling," in *ICC '06*, 2006.

[19] T. F. J. Leguay and V.Conan, "Evaluating mobility pattern space routing for dtnss," in *Proceedings of the 25th IEEE international conference on Computer Communication*, 2006.

[20] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System Architecture Directions for Networked Sensors," in *ASPLOS*, 2000, pp. 93–104.

[21] H. L. I. Niven and H. S. Zuckerman, "An introduction to the theory of numbers," in *John Wiley and Sons*, 1991.

[22] X. Zheng, C.-T. Huang, and M. Matthews, "Chinese remainder theorem based group key management," in *Proceedings of the 45th annual southeast regional conference*, ser. ACM-SE 45, 2007.