Neighbor Discovery and Rendezvous Maintenance with Extended Quorum Systems for Mobile Applications

Desheng Zhang, *Member, IEEE*, Tian He, *Senior Member, IEEE*, Fan Ye, *Member, IEEE*, Raghu K. Ganti, *Member, IEEE*, and Hui Lei, *Fellow, IEEE*

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 duty cycle their radios to conserve the energy and bandwidth, 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. We comprehensively evaluate EQS in three different scales of networks, and the results show that EQS reduces as much as 55 percent energy consumption with a maximal 5 percent increase in latency for existing solutions. To test the real-world values of EQS, we further propose a taxicab dispatching application called EQS-dispatch to navigate taxicab drivers to the area with less competition based on the discovery results of nearby taxicabs.

Index Terms—Mobile network, neighbor discovery, quorum system

1 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 and bandwidth, 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 wireless sensor networks [9], [10], [11], [12], [13], [14] have been proposed. But they typically have two key drawbacks. First, they all use a pairwise discovery mechanism via direct one-hop communication. We argue that such a 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. Second, most of them [12], [13], [14] treats rendezvous maintenance as a rediscovery problem, even though a large number of neighbors have already been discovered. Both of these drawbacks lead to unnecessary active slots that can be eliminated to further conserve energy.

Based on these observations, we develop an extended quorum system, which relies on a legacy mathematical concept called *quorum system* [11] to achieve efficient neighbor discovery and rendezvous maintenance. The extended quorum system propagates known neighborhood information indirectly by bridging multiple pairwise communication, thus avoiding the need for full-mesh pairwise discovery, i.e., avoid the direct one-hop communication. Further, the extended quorum system utilizes the propagation of neighborhood information for the rendezvous maintenance to avoid the rediscovery mechanism. We characterize such information propagation paths of known neighbors using a proposed concept called quorum graph, and propose a reduction algorithm that eliminates redundant paths to reduce the number of active slots. This significantly improves the energy efficiency of the discovery process.

Specifically, we make the following contributions:

- To the best of our knowledge, we present the first neighbor discovery protocol that utilizes the known neighborhood information for both the discovery of

- D. Zhang is with the Department of Computer Science, Rutgers University, New Brunswick, NJ 08901. E-mail: dz220@cs.rutgers.edu.
- T. He is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455. E-mail: tianhe@cs.umn.edu.
- F. Ye is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794. E-mail: fan.ye@stonybrook.edu.
- R.K. Ganti and H. Lei are with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598. E-mail: {rganti, hlei}@us.ibm.com.

Manuscript received 17 Aug. 2015; revised 27 Aug. 2016; accepted 5 Sept. 2016. Date of publication 21 Sept. 2016; date of current version 1 June 2017.
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.2016.2612200

new neighbors and the rendezvous maintenance of old neighbors. In particular, we propose a novel generic concept called *quorum graph* and investigate its property called *reachability*, which is equivalent to neighborhood information flow among mobile devices. Based on the reachability of a quorum graph, we propose the legacy quorum system and extended quorum system, which is a quorum graph satisfying indirect reachability among all devices, such that fewer active slots are required in order to reduce the energy used.

- Based on the quorum graph, we formalize the *quorum reachability minimization problem* to reduce the energy waste due to the redundance of reachability in a quorum graph. We 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 the device discovery and rendezvous maintenance with a significantly reduced number of active slots (i.e., less energy).
- We comprehensively evaluate EQS in three scales of networks: (i) a small-scale testbed experiment with 11 TelosB devices, (ii) a middle-scale simulation with 100 mobile devices, and (iii) a large-scale evaluation based on the GPS data from 14,000 vehicles. The results show that EQS can reduce as much as 55 percent energy consumption with a maximal 5 percent increase on latency for existing protocols.
- To prove the real-world value of EQS, we propose a taxicab dispatching application called EQS-dispatch to show how EQS can be employed by taxicab drivers to select a direction with fewer competing taxis based on the neighboring taxicab discovery. We further evaluate EQS-dispatch based on a 280 GB dataset consisting of 6 months of GPS traces of more than 14,000 taxis in the most crowded city in China, Shenzhen, with 17,150 people per KM^2. Our application shows that a taxicab driver increases the possibility of picking up a passenger by going to the direction with fewer competing taxicabs based on the neighbor discovery of EQS.

The rest of the paper is organized as follows. Section 2 present our motivations and design goal. Section 3 describes theoretical background, followed by our main design EQS in Section 4. Sections 5, 6, and 7 evaluate EQS in three networks with different scales. Section 8 proposes our EQS based application. Section 9 discusses related work. Finally, Section 10 concludes the paper.

2 MOTIVATIONS

Although the existing neighbor discovery protocols guarantee bounded discovery latency (DL) [12], [13], [14], they result in unnecessary active slots (thus a waste of energy) because their discovery schemes are built on direct interactions for pairwise communication. To address such an energy waste, we aim to explore indirect interactions in discovery protocols in terms of energy conservation. In what follows, we first discuss the indirect interactions, and then discuss the potential of using these indirect interactions to conserve the energy. Finally, we present the motivation for our design.

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 motivation.

2.1 Indirect Interaction

For the traditional discovery and rendezvous protocols, the design goal is to ensure that every device can find other devices in the asynchronous networks with direct interactions by themselves, e.g., peer-to-peer communication. Our work is mainly motivated by the observation that these protocols based on direct interactions suffer from unnecessary active slots, and they are especially inefficient when some neighbors have already been discovered. On the other hand, in mobile applications, it is desirable for mobile devices to utilize as few active slots as possible to conserve the energy. To this end, we notice that in many scenarios, neighboring devices share common neighbors, which opens the possibility of indirect interaction for the discovery and rendezvous with fewer active slots. Therefore, we aim to explore how to leverage indirect interactions for better energy efficiency. In particular, the existing protocols typically overlook the fact that neighborhood information may be *indirectly* propagated in the networks during the interactions of devices, e.g., rendezvous among discovered devices. Such an indirect neighborhood information propagation may greatly assist the discovery and rendezvous process, if it is leveraged with existing discovery and rendezvous protocols.

Conceptually, the indirection interactions can be achieved by letting devices broadcast their two-hop neighbor tables in active slots during the neighbor discovery. As a result, for the discovery of new neighbors, a device S can *indirectly* discover its new neighbors via the *direct* rendezvous with discovered neighbors; for the rendezvous with already discovered neighbors (i.e., obtaining the latest neighbor tables from them) a device S can *indirectly* rendezvous with a device B by the direct rendezvous with an intermediate device A, given A and B have already rendezvoused, and 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 a concrete example about such direct and indirect interactions in the neighbor discovery protocols. Three devices S, A and B have already discovered each other. Based on a state-of-the-art discovery protocol Disco [12], 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. Two devices being active at the same slot indicates a rendezvous. As a result, 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. Moreover, S will continue to directly discover new neighbors in the global time slot 1, 4, 7 and 10. However, we can see that in the global time slot 0, A has already rendezvoused with B by exchanging

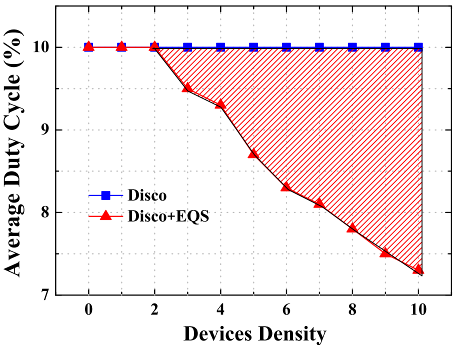

Fig. 2. Illustration of Disco and Disco+EQS.

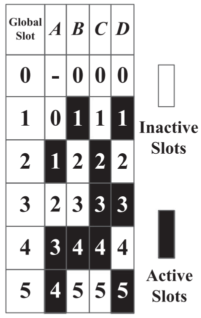

Fig. 3. Schedule.

neighbor tables. Therefore, when A and S rendezvous in the global time slot 4, A's neighbor table containing the neighbors of B will propagate to S. As a result, even if S does not activate its radio in the 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 the rendezvous between A and S). This is an example how neighborhood information propagates indirectly by bridging multiple pairwise communications, which opens the possibility to eliminate the need for fully meshed pairwise discovery and rendezvous. Since the above schedule is known to S after the first time S discovers A and B, S eliminates some of its active slots yet still achieving the indirectly rendezvous.

2.2 Energy Conservation

To intuitively show the effectiveness of the indirect interaction, we plot our testbed results for Disco [12] and its EQS assisting version, Disco+EQS, which takes the indirect interaction into account, as in Fig. 2. The detailed testbed setup is given in Section 6.

Under the networks with an average duty cycle (ADC) 10 percent, as the device density increases, the gap on an average duty cycle between Disco and Disco+EQS is increasingly enlarged. When four devices are in the networks, EQS reduces 7 percent of total active slots in the networks, i.e., 9.3 versus 10 percent. When the number of devices becomes larger, e.g., 10, EQS is capable of reducing 27 percent of total active slots, i.e., 7.3 versus 10 percent. This energy gain of Disco+EQS comes from the reduction of active slots based on indirect interaction, e.g., discovery and rendezvous. It indicates that with more devices in the networks, EQS reduces 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.

2.3 Design Motivation

Given the existence of a plethora of discovery and rendezvous protocols, in Section 4 we decide to design EQS as a transparent augmenting middleware filter on the bottom of them. Given any schedule based on these protocols, EQS transparently filters out the redundant active slots according to indirect discovery and rendezvous. It provides a unified solution for highly diverse, 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 *vice 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 other function as a group, and thus a device finding a new neighbor leads to a quick propagation of neighborhood information to others.

According to the above design philosophy, we face two design challenges: How to capture above indirect nature of neighborhood information propagation, and how to leverage such a nature to adaptively filter out redundant active slots to conserve the energy. To address these two challenges, we first propose a theoretical concept called extended quorum system to capture the indirect propagation as in Section 3, and then we present our main design to filter out the redundant active slots to conserve the energy as in Section 4.

3 Theoretical Background

In this section, we introduce the theoretical concepts for neighbor discovery and rendezvous maintenance. In Section 3.1, we first present a concept called *quorum graph*, and we then give one of its key propriety called *reachability*; in Section 3.2, based on the reachability of quorum graphs, we first propose two special quorum graphs called *legacy quorum system* and *extended quorum system*, and we then discuss their utilization in neighbor discovery and rendezvous schemes; in Section 3.3, based on the legacy and extended quorum systems, we define a *quorum reachability minimization problem* to minimize the energy usage in neighbor discovery and rendezvous, and then we discuss its complexity.

3.1 Quorum Graph

3.1.1 Definition of Quorum Graph

A 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 with an infinite duty cycling loop). We present the formal definition of a 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 features:*

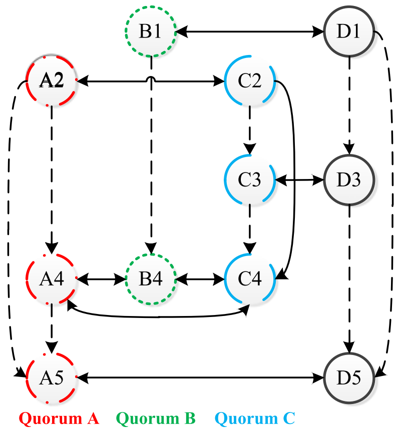

Fig. 4. Quorum graph.

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 an* 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* vice versa. *As a result, the vertical dashed edges are top-down only.*

Based on the above discussion, in Fig. 4, (i) a row of vertices represents the duplicated copies of the same active slot for different devices; (ii) a column of vertices represents the different active slots for the same device; (iii) 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 Figs. 3 and 4, we provide a walk-through of quorum graph construction. (i) Building four subgraphs (quorums) based on four devices in the schedule. (ii) According to the 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. (iii) 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. (iv) 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 slot, i.e., slot 2.

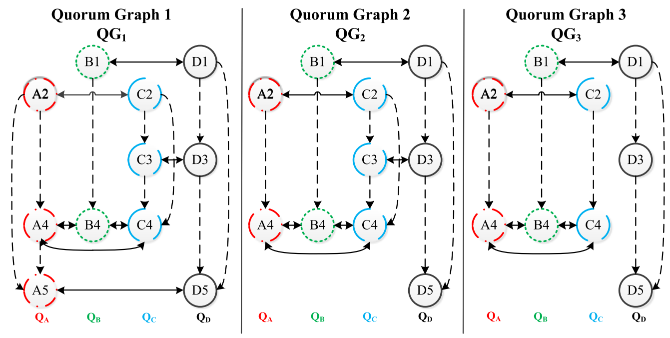

Fig. 5. Examples of quorum graph.

The rationale behind the quorum graph constriction is that with this quorum graph we can capture the neighborhood information propagation among the devices along the time dimension. For bidirectional edges, in Fig. 4, device A and C will rendezvous with each other in a bidirectional way in slot 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 slot 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 the 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.

3.1.2 Reachability

Based on the quorum graph obtained in the last section, we present a key property of the quorum graph, i.e., *reachability*. In the traditional Graph Theory [15], reachability is the notion of being able to access from one vertex in a directed graph to some other vertices. However, under the quorum graph context, we employ the 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 every quorum can reach at least one vertex in every other quorum, then this quorum graph has reachability as a relationship property. Further, if every 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. (i) QG_1 has the 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. (ii) QG_2 has the 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. (iii) 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.

3.2 Quorum System

In this section, via the quorum graph, we introduce a theoretical model, *Legacy Quorum System*, used by current neighbor discovery and rendezvous protocols [9], [10], [11], [12], [13], [16]. Then, we improve this legacy quorum system by an extended quorum system.

3.2.1 Legacy Quorum System

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 the 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 a pairwise and direct discovery and rendezvous between devices.

3.2.2 Extended Quorum System

In this section, based on quorum graph and legacy quorum system, we define a new kind of models for neighbor discovery and rendezvous with the indirect nature. Our motivation about this new kind of model is based on the fact that in the 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, the 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 the definition of this new model by the quorum graph.

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

In Definition 3, under the 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.

3.2.3 Relationship of Direct and Indirect Discovery

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 3, 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.

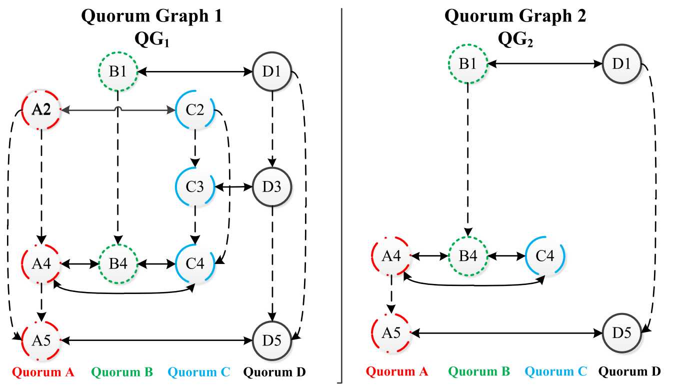

Fig. 6. Example of different quorum system.

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 the 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 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 (based on legacy quorum system) with minimal active slots (based on an extended quorum system) from a given discovery schedule is the key problem we aim to investigate. The formal definition of this problem is given in next section.

3.3 Quorum Reachability Minimization Problem

We present a *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 select minimal number of vertices to maintain 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, better energy performance can be achieved.

Before presenting the solution for QRP, we evaluate its complexity. We prove that QRP is NP-hard by reducing set covering problem to it. Due to the space limitation, the proof is given in [17].

4 EQS DESIGN

In this section, we present some preliminaries network models, and then 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, along with an example about EQS.

4.1 Network Model

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, both the discovery and the rendezvous become complicated [12]. Time synchronization will be greatly helpful, but involves considerable and unaffordable cost [14]. Some related work, e.g., [9], has been proposed to employ the legacy quorum system to tackle the discovery and the 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 [12], [13], [14]. (i) In single-hop networks, time is divided into slots with equal lengths. (ii) The radios of devices in networks are activated in certain active slots according to a given schedule based on the legacy quorum system. (iii) In both beginning and end of an active slot, a device broadcasts its two-hop neighbor table. (iv) An overlapping of active slots between devices indicates a discovery or a rendezvous. Note that even with the clock drift, since a device broadcasts twice in both beginning and end of an active slot, a partial overlapping in an active slot can still guarantee a successful bidirectional discovery [12]. (v) Every device distributedly collects and maintains neighborhood. (vi) Any nodes within two hops are considered as neighbors.

4.2 Main Design

In this section, given a schedule based on the legacy quorum system, we propose a design, EQS, which outputs a filter vector, FV, by solving quorum reachability minimization problem. By the filter vector 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. (i) 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 filter out other unselected vertices. In our discovery scenario, given a schedule, we have to select the minimum number of active slots to maintain reachabilities for every two devices in the networks, and the filter out 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. (ii) 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 introduce *minimum overhead* to reachabilities between themselves.

For the *minimum overhead* to reachabilities, selecting a row of vertices together at a time will obtain the minimum overhead 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 *maximum contribution* to reachabilities, the contribution of a row x, C_x, is computed as follows. $C_x = \frac{T_x}{N_x}$ 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 the above formula, every time we select few *efficient* vertices to provide the *maximum* contributions to total reachabilities (maximum T_x) with the minimal number of active slots (minimum N_x).

Therefore, our scheme is that every time we select a row of vertices such that this row will contribute to minimum 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 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 the 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 the original neighbor discovery schedule, every device in the networks can maintain the discovery and rendezvous with fewer active slots. This FV is constantly changed by EQS with an online updating process 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 $4 \times (4-1)$ (i.e., $N \times (N-1)$) reachabilities. For example, the contribution of the 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 the local maximal in the first round and is $\frac{6}{3} = 2$. Therefore, we select the 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 the 1st row to subgraph, only two reachabilities need to be maintained shown by the table. The 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.

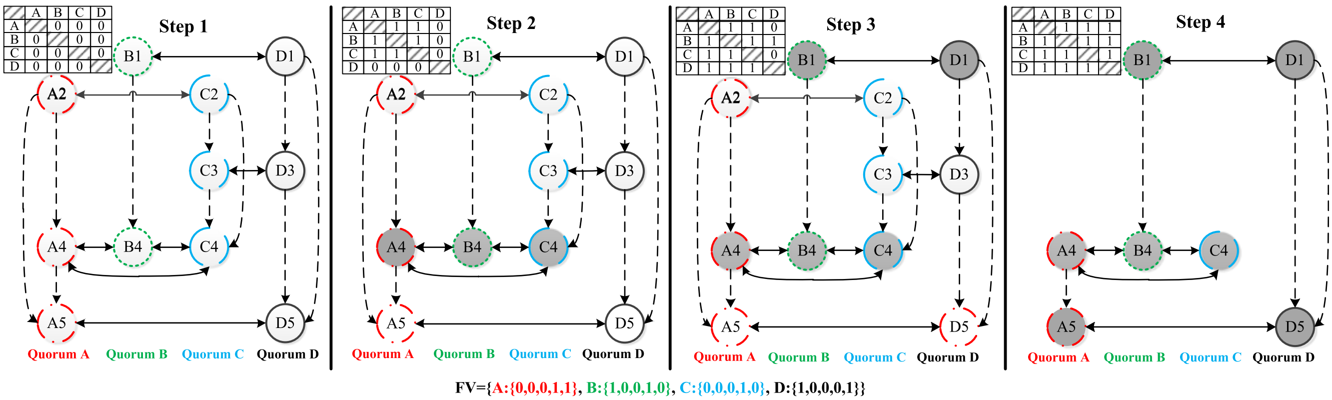

Fig. 7. Example of EQS.

By changing the subgraph to its adjacent matrix, every device can distributedly choose its own column to obtain its FV. For example, device A chooses $\{0, 0, 0, 1, 1\}$ 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 the above example, we can see that EQS takes the legacy quorum system based schedules as input, and outputs a FV for every device, by converting this schedule to a new schedule based on the 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 *vice 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.

5 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 [12] by Dutta et al. in SenSys'08.
- U-Connect [13] 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.

5.1 Simulation Setup

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

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

5.2 Impact of Device Density

The impact of device densities on energy consumption, represented by Average Duty Cycle of all the devices in the network, is shown in Figs. 8 and 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

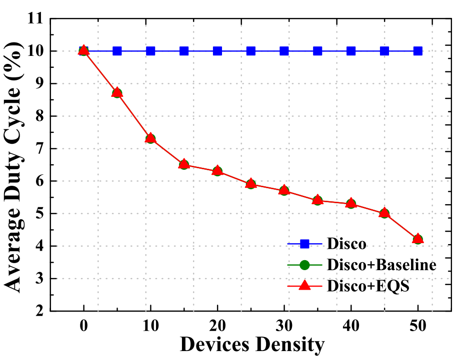

Fig. 8. Disco density.

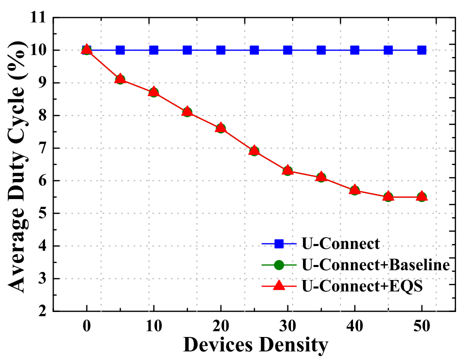

Fig. 9. U-Connect density

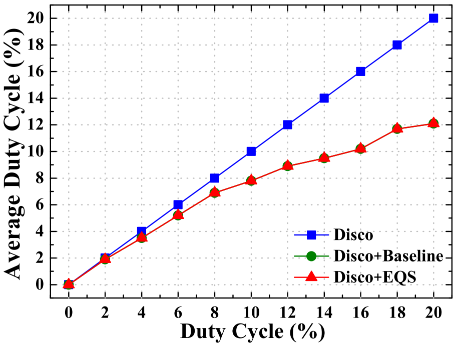

Fig. 10. Disco DC.

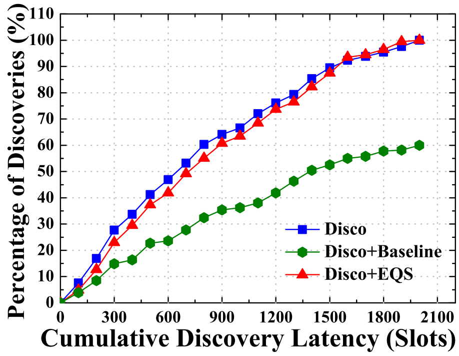

Fig. 12. Disco CDF.

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 of 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 Fig. 8, when the device density is below 20, Disco+EQS achieves an energy gain more than 35 percent. When the device density increases to 50, Disco+EQS achieves more than 55 percent energy gain over Disco by reducing more than half the total active slots. The similar observation is shown in Fig. 9 with less yet still obvious energy gain about 45 percent when the device density is 50. The above observations indicate that EQS functions 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.

5.3 Impact of Device Duty Cycle

In this section, we investigate the impact of device's duty cycle on the energy consumption, which is also shown by the average duty cycles of all the devices in the networks. As shown in Figs. 10 and 11, 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 Fig. 10, we can see that when the duty cycle is 8 percent, the average duty cycle of devices under Disco+EQS obviously outperforms that under Disco with a performance gain of 13.7 percent. As the duty cycle increases to 20 percent, this performance gain is also enlarged to as much as 40 percent. In Fig. 11, even though not as much as it outperforming Disco, EQS still has a maximal 31 percent energy performance gain over U-Connect.

5.4 Impact of Reduction of Active Slots

We plot the CDF of the number of discovery for both of schemes in Figs. 12 and 13. In these figures, as more discovery time is allowed (i.e., increasing cumulative discovery latency), the percentages of discoveries also increase for all schemes. However as in Fig. 12, the devices under Disco and Disco+EQS are able to discover neighbors much faster than under Disco+Baseline. This is because Disco+Baseline only reduces active slots at random, which leads to a maximal 40 percent 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 Fig. 12, in the end of discovery process, EQS enables devices to make a 4 percent 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 a more effective discovery. The similar phenomenon is observed in Fig. 13.

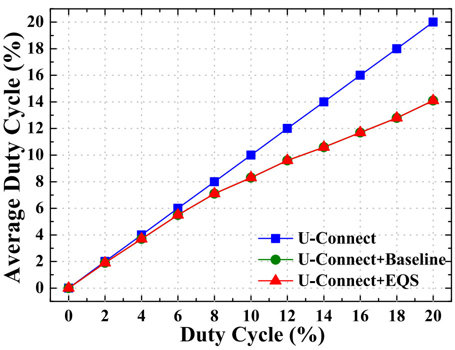

Fig. 11. U-Connect DC.

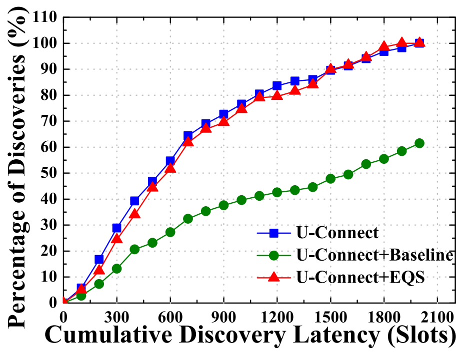

Fig. 13. U-Connect CDF.

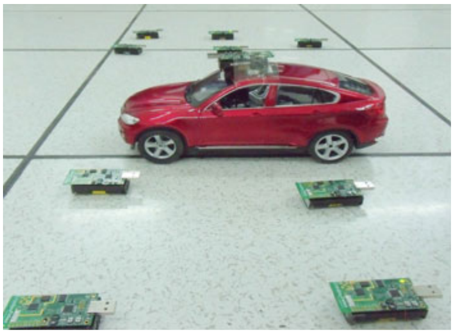

Fig. 14. Testbed setup.

6 Testbed Evaluation

In Section 5, 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 [21]. 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 Fig. 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 25 ms. Due to the conceptual similarity between Disco and U-Connect, as well as the results from simulation, we only implement Disco in our testbed.

6.1 Impact of Device Density

In this section, we report the effectiveness of EQS for energy conservation in the testbed experiment. Fig. 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. 8. However, in Fig. 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 percent. This is because 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.

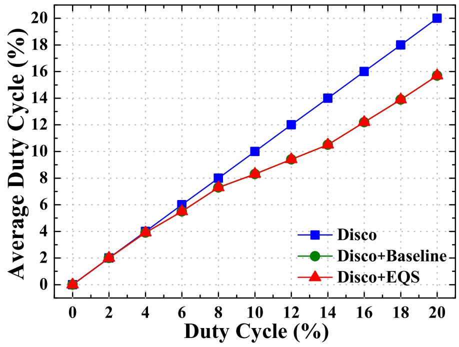

Fig. 16. Impact of DC.

6.2 Impact of Duty Cycle

Fig. 16 shows the impact of different duty cycles on the energy consumption, shown by average duty cycle. In the testbed experiment, when the duty cycle is low, e.g., 4 percent, no reduction is preformed by EQS since a low device density and a 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 percent, EQS can reduce 17 percent active slots according to the extended quorum system. When the duty cycle becomes 20 percent, EQS achieves the maximal energy performance gain, i.e., 21.5 percent. This maximal energy performance gain is smaller than we obtained in large-scale simulation. This is because in the simulation EQS is under much larger and denser networks than in the testbed. This may indicate that EQS is more suitable for large-scale networks.

6.3 Impact of Reduction of Active Slots

Fig. 17 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 percent. The similar results are observed in Fig. 12, but the performance gain is smaller. Again, as observed in Fig. 12, Disco+Baseline has a much worse performance compared with others, and the devices under Disco+Baseline only discover 51 percent of their neighbors, when the devices under Disco+EQS and Disco have discovered all their neighbors.

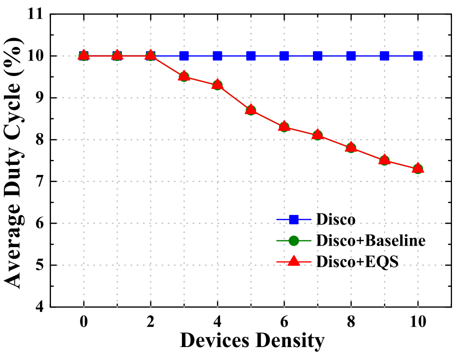

Fig. 15. Impact of density.

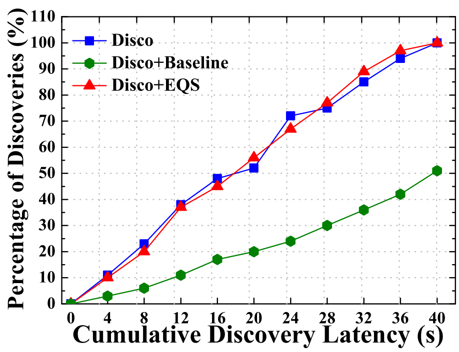

Fig. 17. Testbed CDF.

Fig. 18. Distribution of taxis.

7 Trace-Driven Evaluation

In this section, we evaluate *EQS* with a real-world taxi GPS trace dataset. We first present our evaluation methodology, and then we show impacts of device density, duty cycles, and discover latency on the performance of *EQS*, respectively.

7.1 Methodology

In this trace-driven evaluation, our dataset consists of 6 months of GPS traces from 14,453 taxis in the Chinese City Shenzhen. The data are collected by the Shenzhen government for the urban transportation pattern research. Each taxi uploads records on an average of every 30 seconds, with each record consisting of the following parameters: (i) Plate Number; (ii) Date and Time; (iii) GPS Coordinates; and (iv) Availability: with a passenger or not when the record is uploaded to the dispatch center. Fig. 19 summarizes details of the used datasets. Based on the above GPS trace records, we have a real-time location trace of every taxi in the networks (as shown in Fig. 18).

Based on this dataset, we perform a trace-driven evaluation to verify how *EQS* can reduce the communication overhead in this taxi network if taxi drivers want to find nearby taxis by neighbor discovery. Note that different from regular sensor networks, a taxi network does not have energy constraints, but reducing active slots for neighbor discovery can also minimize the communication overhead for bandwidth efficiency. Essentially, we treat this taxicab network as a mobile network where every taxicab is a mobile device with unique a mobility pattern. We test the impact of cumulative discovery latency on the percentage of discoveries. Further, we test the impact of different device density (by changing communication ranges of taxis) and different duty cycles on energy consumption. With a default duty cycle 10 percent (equal to the uploading speed of GPS record) and 500 m communication range, we compare *EQS* with Baseline and Disco as in the testbed experiment.

Taxicab Network Summary	
Collection Period	6 Months
Collection Date	01/01/12-06/30/12
Numbe of Taxicabs	14,453
Number of Passengers	98,472,628
Total Travel Distance	594,031,428 (KM)
Total Fare	2,255,052,932 (CNY)
Average Travel Distance	6.032 (KM)
Average Fare	22.9 (CNY)

Fig. 19. Statistics.

7.2 Impact of Device Density

In this section, we report the effectiveness of *EQS* for energy conservation in our trace-driven evaluation. Fig. 20 shows the impact of the device density in terms of communication range on average duty cycles. 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, because they filter out the same number of active slots. But we found that when low densities, e.g., 10, there is no reduction of average duty cycle under Disco+*EQS*. When the density increases, the performance gain also increases, shown by reduced average duty cycle. This is because low densities cannot form an extended quorum system for the reduction of active slots in the taxi network, but when density increases, an extended quorum system can be formed to reduce the redundant active slots.

7.3 Impact of Duty Cycle

Fig. 21 shows the impact of different duty cycles on the average duty cycle in our trace-driven evaluation.

We found that when the duty cycle is low, e.g., from 0 to 8 percent, reduction is limited by *EQS* since a low duty

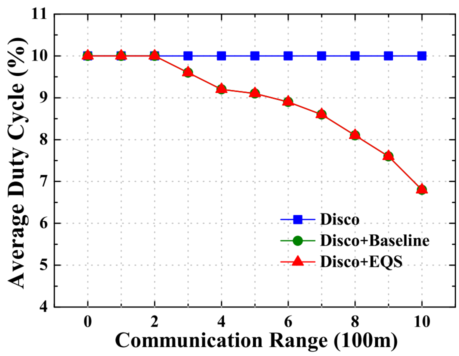

Fig. 20. Impact of density.

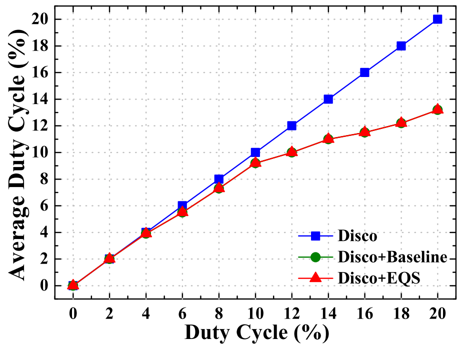

Fig. 21. Impact of duty cycle.

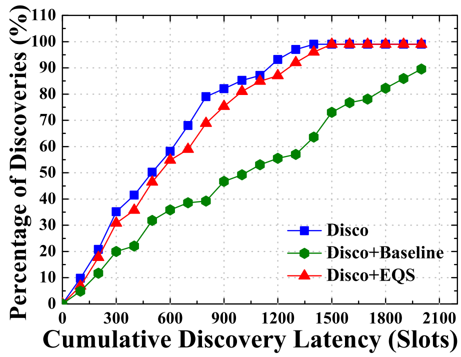

Fig. 22. Trace-driven CDF.

cycle leads to fewer total active slots which cannot form an extended quorum system. But as the duty cycle becomes higher, e.g., 14 percent, EQS significantly reduces average duty cycle compared to Disco. This maximum energy performance gain is large than we obtained in large-scale simulation. This is because in our trace-driven simulation, the density of taxicabs is much higher than that of simulation. This indicates that EQS works better in denser networks.

7.4 Impact of Device Discovery Latency

Fig. 22 plots the trace-driven evaluation results on the CDF of discovery. From this figure, we can see that Disco+EQS continues to exhibit a similar performance to Disco but significantly better than . The similar results are observed in Figs. 12 and 17. Again, as observed before, Disco+Baseline has much worse performance compared with Disco and Disco+EQS, and the devices under Disco+Baseline only discover 73 percent of their neighbors, when the devices under Disco+EQS and Disco have discovered all their neighbors.

8 *EQS*-Dispatch Application

In this section, to prove the real-world value of EQS, we propose and evaluate an application called EQS-Dispatch with which in a taxicab network, taxi drivers quickly navigate optimal directions to cruise the road segments to optimize pickup by an EQS assisted neighbor discovery.

8.1 Application Background

Specifically, in our application, at an intersection, an empty-taxicab driver would select a direction with fewer empty taxicabs to avoid competition, thus increasing the change of picking up a new passenger. A trivial solution to this problem is to install a centralized system, which can be used to collect the information from nearby taxis in terms of locations and status. But such a system requires that each taxi has with cellular data connectivity, which leads to a potential cost. We are thus faced with the challenge of obtaining taxi status with no extra hardware installation on the taxis. In this work, we envision that individual taxi drivers are equipped a smart phone with a peer-to-peer communication interface, e.g., *ad hoc* WiMax or *ad hoc* WiFi, and can install an app to obtain the status of nearby taxis. Taxi drivers with this app can optimize their profits by acting as a group of common interest. The drivers with this app can cruise to the areas with a low density of empty taxis, thus to maximize pickups (and thus profits). Our proposed neighbor protocol plus EQS enables a drive use few active slots to learn status of nearby taxis, which leads to reduced communication overhead, thus more bandwidth for other applications.

In our application, every taxicab broadcasts its own status record (i.e., date and time, availability, direction, GPS coordinates, etc.) to its neighboring taxis with in the communication range. The broadcast is based on Disco and Disco+EQS. According to the information collected, when a taxi driver navigates to the optimal directions, as determined by the number of nearby empty taxis. This metric can potential maximize the probability of picking up the next nearby passengers due to the absence of competing taxis. Based on the distributions of nearby empty taxis, EQS-Dispatch can maximize the possibility of picking up passengers by guiding a taxi to a direction with fewer empty taxis.

8.1.1 *Application Evaluation*

In this section, we evaluate the performance of EQS in navigation for taxis in EQS-disptach. With a total duty cycle $\frac{1}{10}$, we compare three navigating results based on different discovery results of discovery schemes.

(i) *Dispatching with Disco*; navigating taxis with results of Disco;
(ii) *Dispatching with Disco+*EQS; navigating taxis with results of Disco+EQS;
(iii) *Dispatching with Disco+Baseline:* navigating taxis with results of Disco+Baseline as in the previous evaluation.

Under all dispatching, a taxi has the same preferable directions for fewer competing empty taxis. But since the employed discovery schemes are different, a dispatching with a faster discovery scheme may achieve better performance. The performance is characterized by two metrics: empty taxis density and an average duty cycle (for communication overhead). To show the difference with or without our EQS-dispatch, we also compare the above three schemes with *Ground Truth without Navigation*, where the density is computed based on original taxi traces without altering the routes of any taxis. Note that given the density of competing taxis, how to select the optimal route to achieve the optimal density is outside the scope of this paper. We simply let taxi drivers greedily select one out of four directions in an intersection according to empty taxi densities in every direction and then compute densities of competing taxis in its neighborhood every minute.

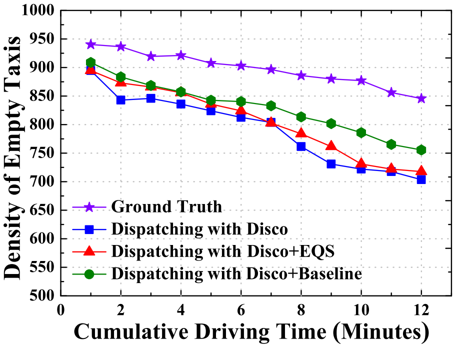

Fig. 23. Empty taxi density.

We investigate the densities of empty taxis in three different dispatching strategies. We report the results of navigating taxis using a 3 KM communication radius in Fig. 23. We found that with the increase of the cumulative driving time, all dispatching schedules have decreasing density of empty taxis, but the dispatching based on Disco and Disco+EQS has better performance than the dispatching based on Disco+Baseline. This is because Disco and Disco+EQS can discover neighboring taxis faster, which makes it go to the direction with low density of empty taxis faster. Compared to Disco, even though dispatching based on Disco+EQS has slighter lower performance but dispatching based on Disco+EQS is more efficient, since it has less communication overhead as shown before.

9 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. We also provide some related work about our taxi-dispatching application.

9.1 Explicitly Quorum System Based Schemes

To address discovery and rendezvous problem with a bounded worst-case latency, the explicitly quorum systems 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 a global parameter, 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 [16], 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.

9.2 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 [12], [13], [14]. These protocols select one or multiple prime numbers for every device to represent their duty cycles. Based on the Chinese Remainder Theorem [22], [23], these devices would have bounded discovery and rendezvous latency based their chosen duty cycles. These protocols implicitly employ the idea of quorum systems to enable every two of devices in the networks have at least one common active slots for each other. In Disco [12], 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 [13] 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 [14] 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.

9.3 Taxi Applications

Recently data-driven applications in taxi system receive significant attention due to availability of large-scale GPS traces. Some systems are proposed to assist taxicab operators for better taxicab services, e.g., inferring mobility patterns for taxicab passengers [24], detecting anomalous taxicab trips to discover driver fraud [25], exploring carpooling opportunities [26], dispatching taxicabs based on inferred passenger demand [27], [28], [29], and discovering temporal and spatial causal interactions to provide timely and efficient services in certain areas with disequilibrium [30]. In addition to taxicab operators, several systems are proposed for the benefit of passengers or drivers, e.g., allowing taxicab passengers to query the expected duration and fare of a planed trip based on previous trips [31], computing faster routes by taking into account driving patterns of taxicabs obtained from historical GPS trajectories [32], estimating city traffic volumes for drivers [33], and recommending a taxicab driver with a sequence of pick-up points to maximize profits [27]. But almost all these applications are based on centralized dispatching, but our application is based on distributed dispatching where a taxi learns status of nearby taxis by peer to peer communications.

10 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 percent energy with a maximal 5 percent increase on latency. Finally, we propose a taxi-dispatching application called EQS-Dispatch based on EQS, and the evaluation results show that it can quickly navigate an empty taxi to a direction with few competing taxis to maximize potential profit.

Acknowledgments

This research was supported in part by the US National Science Foundation grants CNS-1446640 and CNS-1544887 and IBM's Open Collaborative Research Award Program. A preliminary version of this work is published in [17].

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Commun. Mag.*, vol. 49, no. 11, pp. 32–39, Nov. 2011.
[2] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 140–150, Sep. 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 *Proc. 5th ACM Conf. Embedded Netw. Sensor Syst.*, 2007, pp. 87–101.
[4] H. Liu, et al., "Automatic and robust breadcrumb system deployment for indoor firefighter applications," in *Proc. 8th Int. Conf. Mobile Syst. Appl. Serv.*, 2010, pp. 21–34.
[5] P. Lukowicz, T. Choudhury, and H. Gellersen, "Beyond context awareness," *IEEE Pervasive Comput.*, vol. 10, no. 4, pp. 15 –17, Apr. 2011.
[6] J.-H. Huang, S. Amjad, and S. Mishra, "CenWits: A sensor-based loosely coupled search and rescue system using witnesses," in *Proc. 3rd Int. Conf. Embedded Networked Sensor Syst.*, 2005, pp. 180–191.
[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. 10th Int. Conf. Archit. Support Program. Languages Operating Syst.*, Oct. 2002, pp. 96–107.
[8] Y. Wang, et al., "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proc. 7th Int. Conf. Mobile Syst. Appl. Serv.*, 2009, pp. 179–192.
[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 *Proc. IEEE 21st Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2002, pp. 200–209.
[10] 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.
[11] S. Lai, B. Zhang, B. Ravindran, and H. Cho, "CQS-pair: Cyclic quorum system pair for wakeup scheduling in wireless sensor networks," in *Proc. 12th Int. Conf. Principles Distrib. Syst.*, 2008, pp. 295–310.
[12] P. Dutta and D. Culler, "Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, 2008, pp. 71–84.
[13] A. Kandhalu, K. Lakshmanan, and R. R. Rajkumar, "U-connect: A low-latency energy-efficient asynchronous neighbor discovery protocol," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2010, pp. 350–361.
[14] A. Purohit, N. Priyantha, and J. Liu, "WiFlock: Collaborative group discovery and maintenance in mobile sensor networks," in *Proc. 10th Int. Conf. Inf. Process. Sensor Netw.*, 2011, pp. 37–48.
[15] M. Mitzenmacher and U. Upfal, *Probabilitty and Computing*. Cambridge, U.K.: Cambridge Univ. Press, 2007.
[16] 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.
[17] D. Zhang, T. He, F. Ye, R. K. Ganti, and H. Lei, "EQS: Neighbor discovery and rendezvous maintenance with extended quorum system for mobile applications," in *Proc. 32nd Int. Conf. Distrib. Comput. Syst*, 2012, pp. 72–81.
[18] 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.
[19] 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.
[20] T. Friedman, J. Leguay, and V. Conan, "Evaluating mobility pattern space routing for DTNSS," in *Proc. 25th IEEE Int. Conf. Comput. Commun.*, 2006, pp. 1–10.
[21] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, "System architecture directions for networked sensors," in *Proc. 9th Int. Conf. Archit. Support Program. Languages Operating Syst.*, 2000, pp. 93–104.
[22] H. L. I. Niven and H. S. Zuckerman, *An Introduction to the Theory of Numbers*. Hoboken, NJ, USA: Wiley, 1991.
[23] X. Zheng, C.-T. Huang, and M. Matthews, "Chinese remainder theorem based group key management," in *Proc. 45th Annu. Southeast Regional Conf.*, 2007, pp. 266–271.
[24] C. Kang, S. Sobolevsky, Y. Liu, and C. Ratti, "Exploring human movements in Singapore: A comparative analysis based on mobile phone and taxicab usages," in *Proc. 2nd ACM SIGKDD Int. Workshop Urban Comput.*, 2013, pp. 1:1–1:8.
[25] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, and S. Li, "iBAT: Detecting anomalous taxi trajectories from GPS traces," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 99–108.
[26] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proc. Nat. Academy Sci. USA*, vol. 111, pp. 13290–13294, 2014.
[27] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani, "An energy-efficient mobile recommender system," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 899–908.
[28] Y. Huang and J. W. Powell, "Detecting regions of disequilibrium in taxi services under uncertainty," in *Proc. 20th Int. Conf. Advances Geographic Inf. Syst.*, 2012, pp. 139–148.
[29] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun, "Where to find my next passenger," in *Proc. 13th Int. Conf. Ubiquitous Comput.*, 2011, pp. 109–118.
[30] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 1010–1018.
[31] R. K. Balan, K. X. Nguyen, and L. Jiang, "Real-time trip information service for a large taxi fleet," in *Proc. 9th Int. Conf. Mobile Syst. Appl. Serv.*, 2011, pp. 99–112.
[32] H. Gonzalez, J. Han, X. Li, M. Myslinska, and J. P. Sondag, "Adaptive fastest path computation on a road network: A traffic mining approach," in *Proc. 33rd Int. Conf. Very Large Data Bases*, 2007, pp. 794–805.
[33] J. Aslam, S. Lim, X. Pan, and D. Rus, "City-scale traffic estimation from a roving sensor network," in *Proc. 10th ACM Conf. Embedded Netw. Sensor Syst.*, 2012, pp. 141–154.

Desheng Zhang is an assistant professor in the Department of Computer Science, Rutgers University. He is broadly concentrated on bridging cyber-physical systems (also known as Internet of Things under some contexts) and big urban data by technical integration of communication, computation, and control in data-intensive urban systems. He is focused on the life cycle of big-data-driven urban systems, from multi-source data collection to streaming-data processing, heterogeneous-data management, model abstraction, visualization, privacy, service design, and deployment in complex urban setting. He is currently interested in real-time interactions among heterogeneous urban systems including cellphone, smartcard, taxi, bus, truck, subway, bike, personal vehicle, electric vehicle, and road networks. He is a member of IEEE.

Tian He is an associate professor in the Department of Computer Science and Engineering, University of Minnesota Twin Cities. 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 received the 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 of six international journals including the *ACM Transactions on Sensor Networks*. His research include wireless sensor networks, cyber-physical systems, intelligent transportation systems, real-time embedded systems, and distributed systems, supported by National Science Foundation, IBM, Microsoft, and other agencies. He is a senior member of the IEEE.

Fan Ye is an assistant professor in the Electrical and Computer Engineering Department, Stony Brook University. His research interests include mobile sensing systems and applications, wireless and sensor networks, indoor localization, and floor plan. He has worked on a number of projects in mobile and distributed systems, including mobile crowd sourced sensing and its infrastructure, cloud based publish/subscribe service, wide area messaging, hybrid high availability, and collaborating sites in stream processing systems. He is a member of the IEEE.

Raghu. K. Ganti has been a research staff member with the IBM Thomas J. Watson Research Center, Yorktown Heights, since September 2010. He is part of the Cloud Based Networks Department. His research interests span wireless sensor networks, privacy, and data mining. For the past two years, he has been dabbling with spatiotemporal analytics—the analysis of moving objects and been developing various algorithms for spatiotemporally enabling IBM's big data products. Parts of his work are now embedded in products such as IBM InfoSphere Streams, SPSS statistical modeler, and InfoSphere SenseMaking. He is a member of the IEEE.

Hui Lei is a senior manager in the Cloud Platform Technologies Department, IBM Thomas J. Watson Research Center, and co-leads IBM's worldwide research strategy in cloud infrastructure services and cloud managed services. He is a member of the IBM Academy of Technology, and has assumed various leadership roles with IBM Research, including chair of the Mobile Computing Professional Interests Community, strategy consultant of IBM's worldwide research in distributed computing and in programming models and tools, global research co-lead of connectivity and integration middleware, manager of the Messaging and Event Systems Department, and manager of the Cloud Management Services Department. He is a fellow of the IEEE.