Ad Hoc & Sensor Wireless Networks, Vol. 0, pp. 1–17
Reprints available directly from the publisher
Photocopying permitted by license only

ERUPT: A Role-based Neighbor Discovery Protocol for Mobile Social Applications

SIQIAN YANG[1], ZHONG LI[1], MILICA STOJMENOVIC[2], CHENG WANG[1,3] AND CHANGJUN JIANG[1]

[1]*Department of Computer Science Tongji University*
E-mail: youngfourkings@gmail.com, lizhong007@hotmail.com
[2]*SUCCESS, FICT, Swinburne University of Technology, Melbourne, Australia*
E-mail: mstojmenovic@swin.edu.au
[3]*School of Information Technology and Engineering, University of Ottawa*
E-mail: 3chengwang@gmail.com

Received: November 28, 2013. Accepted: February 9, 2014.

With the popularity of mobile social applications, the requirements of rapid and energy efficient become the great challenges for neighbor discovery protocols. In mobile social applications, mobile nodes usually play different roles (active and passive). This relationship between mobile nodes is a characteristic in neighbor discovery problem. In the daily life, the start time and duration of a neighbor detection are determined by the node which launches the application. In other words, the neighbor discovery will occur suddenly at any time in the network. So timelimit is another characteristic in neighbor discovery. This paper proposes a new neighbor discovery protocol called ERUPT that takes two characteristics above into account. In the ERUPT protocol, the mobile nodes which launch applications in the network are named sponsor nodes, and those nodes that participate applications following the sponsor nodes are named participant nodes. The core idea of ERUPT neighbor discovery protocol depends on the common phenomenon that the sponsor nodes will spend more energy to invite more participant nodes into applications as soon as possible. More efficient of discovering neighbors for the sponsor nodes than existing approaches is validated by the theoretical analysis and simulation in this paper. We evaluate the ERUPT protocol through NS2 network simulator, and show almost 26% − 30% improvement in discovery latency while at almost the same energy consumption over existing approaches.

1 INTRODUCTION

More and more social applications follow local communication trend and aim to attract more local individual users to participant in. We called it mobile social application, such as mobile gaming (e.g., Nintendo's StreetPass [1] and Sony's Vita [2]). This requires the network services provide possible nearby participants when a user actively operates a social application by a low-latency and energy-efficient approach in a limited time. Low-latency and energy-efficient are the keys to network services to meet the requirement above, which is called neighbor discovery problem.

Recent neighbor discovery algorithms can be classified into two categories, *randomized algorithms* and *deterministic algorithms*. In a randomized neighbor discovery algorithm such as Birthday protocol [3], each node transmits at randomly chosen timeslots and yet discovers all its neighbors by a given time with high probability. In a deterministic neighbor discovery algorithm, each node transmits according to a pre-determined transmission schedule that guarantees it to discover all its neighbors by a given time, such as Disco [4], U-connect [5] and Searchlight [6].

Since most deterministic neighbor discovery algorithms are based on rendezvous, all nodes play the same role. However, we find two nodes exist a relationship between active role and passive role when they make a neighbor discovery occuring in our daily life. On the other hand, we find the existing discovery protocols are suit for static nodes, but not for mobile nodes. Because they focus on the discovery links of the whole network and ignore each individual nodes, and this leads to a long discovery latency for a single node. When a mobile node operates a local social application, it's unacceptable to update its neighbor list repeatedly to check whether the links discovered before it exists.

In this paper, taking above two issues into account, we propose a neighbor discovery protocol named ERUPT. Based on above analysis, we name the node sponsor when it plays an active role in neighbor discovery; and we name the node participant when it plays a passive role. The design of the ERUPT protocol is based on the philosophy that the sponsors want to use a little more energy to exchange for less discover time and more neighbors to start the application as soon as possible. From the view of a sponsor, the energy consumption displays a eruption in a period of time if it wants to spend a little more energy. In ERUPT, when broadcasting messages to describe the energy eruption of the sponsors, we use recession approach. In this way, the sponsor will invite the neighbors just founding to participant local social application. We only analyze the situation that one node is the sponsor node and the others are participant nodes. If there are more than one node are in sponsor mode, a confliction will occur at the participant nodes. We take the confliction avoidance problem as future work. The ERUPT algorithm performs almost $26\% - 30\%$ improvement than other protocols in discovery latency while consuming almost the same energy of individual users in simulation. The main contributions of this paper are listed as follows.

1. Propose an efficient neighbor discovery algorithm named ERUPT. The ERUPT algorithm distinguishes between active and passive roles from the mobile nodes in neighbor discovery.
2. A proof is presented of the discovery latency and the energy consumption of the ERUPT algorithm. We prove the discovery latency is low and the energy consumption is tolerable.
3. We show a tradeoff of the discovery latency and the energy consumption between the ERUPT protocol and existing protocols. The ERUPT algorithm performs almost $26\% - 30\%$ improvement than other protocols in discovery latency while consumes almost the same energy of individual users in simulation.

The rest of paper is organized as follows. Section 2 describes the history and related work of neighbor discovery protocols. In Section 3, we propose wireless network model and assumptions which is used for description and analysis. We describe the ERUPT algorithm in Section 4. In Section 5 and Section 6, we analyze the latency and energy consumption under the two-node case and multi-node case. Section 7 presents the simulation-based performance of the ERUPT protocol in comparison to existing neighbor discovery protocols. We conclude our work and outline the direction of future research in Section 8.

2 RELATED WORK

Neighbor discovery is used in proximity-based mobile applications to let mobile devices find each other. The solution of neighbor discovery depends on duty-cycling schemes. For example, when nodes are deployed in a static and dense network, they can synchronize [7] their clocks using GPS, and send packets with long preambles [8, 9] until it is acknowledged. However, this is based on that a sender has a expectation that the receiver is nearby, and its duty cycle is known. On the other hand, synchronization through GPS is usually too energy-expensive for mobile sensors [10] and smartphones [11].

Then neighbor discovery comes into asynchronous protocols time. The majority of asynchronous protocols, for example BMAC [9], SMAC [12], assume that all devices have symmetric sleep patterns. In the other word, all devices run at the same duty cycle. In real world, nodes will formulate their duty cycles with varying energy requirements, resulting in asymmetric duty cycles. McGlynn and Borbash propose "Birthday protocols", which is a asymmetric protocol and focuses on saving energy. "Birthday protocols" uses a probabilistic approach to solve the challenge of asymmetric duty cycles and performs well in static wireless networks with multiple neighbors for average-case. However, it cannot provide a worst-case bound.

To solve the weakness of probabilistic approaches, deterministic protocols appear. Deterministic protocols are divided into two parts. One is Quorum-based protocols [12, 13], and the other is Prime-based protocols, like Disco [4], U-connect [5]. The former puts time as a two-dimensional square, and the node random chooses one row and one column as active slots. The discover will occur when a pair of two nodes has at least two active slots overlapped. Nevertheless, the Quorum-based protocols can not solve asymmetric case, because the two-dimensional time square should be same to every node. The latter takes asymmetric case into account. In Disco, each node choses a pair of unequal prime numbers. When the sequence number of the time slot is divisible by any of its prime numbers, the node turns into active state. For example, one node chooses (p_1, p_2), and the other node chooses (p_3, p_4). The discover must occur in $min\{(p_1 \cdot p_3), (p_1 \cdot p_4), (p_2 \cdot p_3), (p_2 \cdot p_4)\}$ time slots according to Chinese Remainder Theorem [14]. U-connect is the improvement of Disco using one prime instead of two. The deterministic protocols can guarantee the worst-case bound, but performs not as well as probabilistic approaches for average-case.

Searchlight [6], as a deterministic asymmetric protocols, is proposed to improve the performance of average-case. Searchlight proposes anchor slots and probe slots with the rule that the offset between the awake slots of any two nodes is the same when they have the same duty cycle. Anchor slots means a node come into active every first time slot in a period, and probe slots keep moving between periods. If the probe slots of one node have an overlap with the probe slots or anchor slots of the other node, the discover occurs. In asymmetric case, Searchlight use prime-based algorithm such as Disco. Searchlight shows better performance of average-case than traditional prime-based protocols. However, Searchlight only improves the performance in symmetric case.

With the development of neighbor discovery, the challenge of balancing the battery of device and the efficient of discovering neighbors is the key point. Most deterministic protocols, like Disco, U-connect, Searchlight don't consider the relationship between two nodes when a neighbor discovery occurs. In reality, almost all the mobile social applications have a sponsor. The sponsor launches neighbor discovery initiative to invite more nodes to play with. On the other hand, the existing protocols perform worse in discovery latency of a single node. The ERUPT protocol proposed in this paper will take these in consideration.

3 WIRELESS NETWORK MODEL AND ASSUMPTIONS

In this section, we give the wireless network model and assumptions which will be used in our following proposed algorithm.

- **Node** We assume that N denotes the total number of nodes in the network. Each node has its unique ID. It can distinguish from others labeled a MAC address. Each node is equipped with a radio transceiver that allows a node to either transmit or receive messages, but not synchronously.
- **Neighbor** Two nodes become neighbors through neighbor discovery if and only if one node transmits message and the other one listens. When the transmission node has received the listening node's feedback, the two nodes become neighbors.
- **State** A node can be in one of three states: transmit(T), listening(L), or sleeping(S). A node in transmit state broadcasts a discovery message advertising itself. A node which in listening state listens for discovery message. If such a message is heard, the node will feedback to the source address of the message. A node in sleeping state is neither broadcasting nor listening.
- **Mode** Two modes are worked for the mobile nodes in the network. One is sponsor(SP), the other is participant(PA). We call the node which is in sponsor mode *sponsor node*. The sponsor node is either in transmit state or sleeping state. When a node is in participant mode, we call it *participant node*. The participant node is either in listening state or sleeping state.
- **Time** Time is slotted. We let t denote the *time cycle*. Time cycle is a period of time of a node which is determined by the *duty cycle* (The duty cycle is defined as the ratio of transmit or listen period to a complete cycle) of a node. Time cycle consists of several time slots. Based on the assumption of time cycle, we let I denote the *working cycle*. Working cycle denotes that a node is in sponsor mode. The expression of I is as follows:

$$I = \lfloor t/2 \rfloor \cdot t$$

- **Energy** Define energy consumption of a node is zero when a node is in sleeping state. When a node is in transmit state or listening state, we assume the energy consumption is the same in a time slot, and we assume that it is 1.

4 THE DESCRIPTION OF THE ERUPT ALGORITHM

In this section, we propose the description of ERUPT neighbor discovery algorithm. We introduce ERUPT algorithm based on the model and assumptions provided in section 3. In daily life, the sponsor launches a neighbor detection when he or she wants to apply a mobile social application (i.e., there is a sponsor behind every neighbor discovery). According to this, the mobile nodes in the network have distinction between active and passive roles when a neighbor discovery occurs. We find a common phenomenon that the sponsors will spend more energy to invite more devices to join the applications as soon as possible. When a sponsor does this, the neighbor discovery occurs explosively both in time and energy consumption of the whole network. The ERUPT neighbor discovery algorithm is designed based on above analysis. The algorithm is divided into two steps as follows.

1. When a node is in SP mode, we name it sponsor node. The sponsor node has a working cycle I, which is divided into *all-out* part and *recession* part. The *all-out* part is the first time cycle t of I. The sponsor node in the *all-out* part broadcasts message at every timeslot. The *recession* part is from the second time cycle t to $\lfloor t/\lfloor t/2 \rfloor \rfloor$th t in I. In *recession* part, a rule that the sponsor node broadcasts message when $t \equiv 1(\mathrm{mod}\, k)$ will be followed, where k is the serial number of current time cycle t of I.
2. When a node is in PA mode, we name it participant node. It divides its time cycle into *Part One* and *Part Two*. The former consists of $\lfloor t/2 \rfloor$ time slots, the latter consists of $\lceil t/2 \rceil$ time slots. Then the participant node chooses one timeslot randomly in listening state in each part, and is in sleeping state of the other timeslots.

According to the description of ERUPT, a neighbor discovery occurs when a node is in transmit state in SP mode and another node is in listening state in PA mode. An example is showed in Figure 1(a), where the time cycle t of both node S and node P are set to 4.

5 ANALYSIS TWO-NODE CASE

In this section, we discuss the case where there are a sponsor node and a participant node in the network. In the two-node network, the discovery occurs when a node is in transmit state and the other is in listening state. We define the two nodes S and P respectively. Node S is in sponsor mode, and node P is in participant mode.

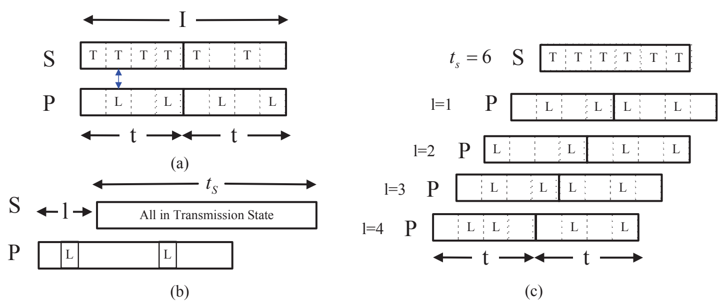

FIGURE 1
In Figure (a), node S is the sponsor node and node P is the participant node when time cycle t=4. Figure (b) shows the offset between the time cycle of the participant node P to the sponsor node S while the sponsor node S starts its working cycle. Figure (c) shows the cases of different possible offsets, where $t_S = 6, t_P = 4$

We analyze ERUPT algorithm from two aspects. First, we investigate the average time it takes a sponsor to discover a neighbor. Here, such a average time is called the *average latency*. Second, we will show the duty cycle and energy consumption of the sponsor node and the participant node.

5.1 Average Latency

In this section, we discuss average latency in ERUPT algorithm. We let t_S denote the time cycle of node S and let t_P denote that of node P. We analyze ERUPT algorithm from $t_S \geq t_P$ and $t_S < t_P$ two aspects. When the sponsor node S starts its working cycle, we assume that the offset between the time cycle of node S and P is l (see Figure 1(b)). l consists of an integral number of timeslots.

Lemma 1. *The offset (see Figure 1(c)) between the time cycle of the participant node P to the sponsor node S while the sponsor node S starts its working cycle is* $l \in \{0, 1, 2, ..., t_P - 1\}$.

We first discuss the case when $t_S \geq t_P$ and show the short average latency.

Lemma 2. *When* $t_S \geq t_P$*, the discovery will occur in the all-out part of the sponsor node S (see Figure 1(c)).*

Proof. Based on the description of the sponsor node in Section 3, when node S starts its working cycle, it will broadcast messages in all timeslots in its *all-out* part. In Lemma 1, we know while node S begins to broadcast messages, l

can be any one in $\{0, 1, 2, ..., t_P - 1\}$. Whatever l is, t_S will contain one part of the participant node P (see Figure 1(c)). Based on the description of the participant node, node P is in listening state in one timeslot in both two parts. So the sponsor node S will discover node P in its *all-out* part.

Theorem 1. *When $t_S \geq t_P$, the average latency of node S discovering node P is almost $\frac{7t_P}{24} + \frac{1}{2}$.*

Proof. We assume the probability of $l \in \{0, 1, 2, ..., t_P - 1\}$ is the same. We introduce $k = \lceil t_P/2 \rceil$. When t_P is an even, $t_P = 2k$. We calculate the expectation through dividing l into $0 \leq l < k$ and $k \leq l < 2k$ two cases, then put them together. We get the average latency is $\frac{7k^2 + 6k - 1}{12k} \approx \frac{7t_P}{24} + \frac{1}{2}$. When t_P is an odd, $t_P = 2k - 1$. Same to the case $t_P = 2k$, we get the average latency is $\frac{7k^2 - k - 2}{12k - 6} \approx \frac{7t_P}{24} + \frac{1}{2}$. So the average latency of node S discovering node P is almost $\frac{7t_P}{24} + \frac{1}{2}$.

In the case $t_S < t_P$, since the sponsor node S may not discover the participant node P if t_P is large enough, we will discuss the relationship between the discoverable probability and the choice of t_S in this case. First, we give some assumptions to assist our analysis. We assume $d_{SP} = t_P - t_S$, where d_{SP} denotes the difference of the time cycle between node S and node P. Then we let P_1 denote the Part One and P_2 denote the Part Two in the time cycle of the participant node P. We have $P_1 = \lfloor t_P/2 \rfloor$, $P_2 = \lceil t_P/2 \rceil$ and $t_P = P_1 + P_2$. At last, we use p_{SP} to denote the discoverable probability and $p_{SP} = p_h + p_t$. p_h denotes the probability that the discovery occurs in the *all-out* part of the working cycle I_S, and p_t denotes that in the *recession* part.

Theorem 2. *The average value of p_h approaches $\frac{17}{24}$.*

Proof. We first discuss the case $1 \leq d_{SP} \leq P_1$. (i.e., $t_S < t_P \leq 2t_S$), There are four cases of l. (Figure 2(a) to Figure 2(d)). We calculate the expectation of l, we use $p_{d_{SP}}$ to denote it. Then we get the expectation of d_{SP} based on $p_{d_{SP}}$. So we have

$$p_h = \sum_{d_{SP}=1}^{P_1} p_{d_{SP}} = \sum_{d_{SP}=1}^{P_1} \left(1 - \frac{d_{SP}^3 - d_{SP}}{3P_1P_2(P_1 + P_2)}\right) \approx \frac{23}{24} \qquad (1)$$

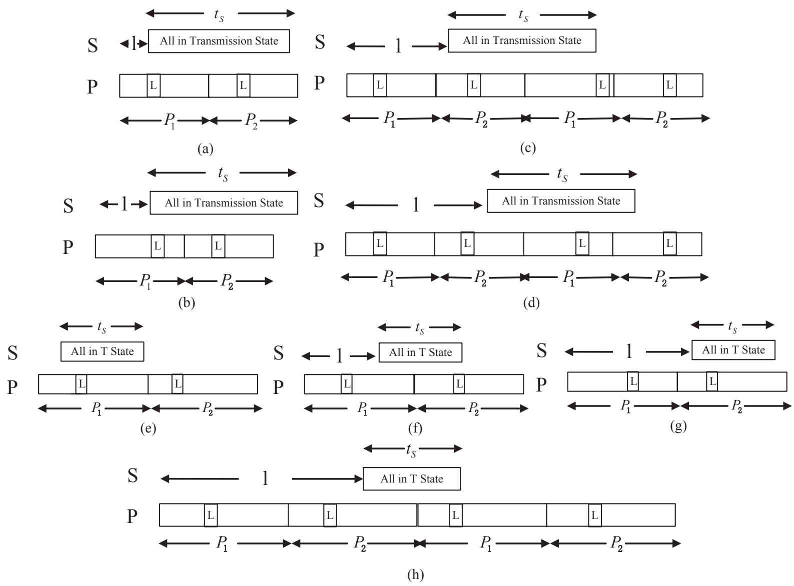

FIGURE 2
Figure (a) to (d) shows four cases of l when $1 \le d_{SP} \le P_1$. Figure (a) shows the case $1 \le l \le d_{SP} - 1$ and the probability is $1 - \frac{l}{P_1 P_2} \cdot (d_{SP} - l)$. Figure (b) shows the case $d_{SP} \le l \le P_1$ and the probability is 1. Figure (c) shows the case $P_1 + 1 \le l \le d_{SP} + P_1 - 1$ and the probability is $1 - \frac{l - P_1}{P_1 P_2}[d_{SP} - (l - P_1)]$. Figure (d) shows the case $d_{SP} + P_1 \le l \le P_1 + P_2$ and the probability is 1. Figure (e) to (h) shows four cases of l when $P_1 + 1 \le d_{SP} \le P_1 + P_2 - 1$. Figure (e) shows the case $1 \le l \le P_1 - t_S$ and the probability is $\frac{P_1 + P_2 - d_{SP}}{P_1}$. Figure (f) shows the case $P_1 - t_S + 1 \le l \le P_1$ and the probability is $1 - \frac{l}{P_1 P_2} \cdot (d_{SP} - l)$. Figure (g) shows the case $P_1 + 1 \le l \le d_{SP}$. Figure (h) shows the case $d_{SP} + 1 \le l \le P_1 + P_2$ and the probability is $1 - \frac{l - P_1}{P_1 P_2}[d_{SP} - (l - P_1)]$

Then we consider the case $P_1 + 1 \le d_{SP} \le P_1 + P_2 - 1$. It's equal to $t_P > 2t_S$. There are still four cases of l (Figure 2(e) to Figure 2(h)).

We get the value of p_h under this case is

$$p_h = \sum_{d_{SP}=P_1+1}^{P_1+P_2-1} p_{d_{SP}} = \sum_{d_{SP}=P_1+1}^{P_1+P_2-1} \left\{ 1 + \frac{1}{3P_1P_2(P_1+P_2)}[d_{SP}^3 - (3P_1 + 3P_2)d_{SP}^2 + (3P_1^2 + 3P_2^2 - 1)d_{SP} - . \; P_1^3 - P_2^3 + P_1 + P_2] \right\} \approx \frac{11}{24} \tag{2}$$

We combine equation (1) and (2), and we get $p_h = \dfrac{17}{24}$

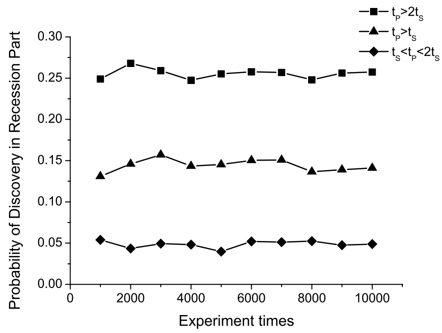

FIGURE 3
The probability of the discovery in recession part from $t_P > 2t_S$, $t_P > tA$ and $t_S < t_P < 2t_S$ three cases

Since p_t is hard to get the formula, we get it through experiments by using mathematical simulator tools. As illustrated in Figure 3, when $1 \leq d_{SP} \leq P_1,\ p_t \approx 0.0487$. When $P_1 + 1 \leq d_{SP} \leq P_1 + P_2 - 1,\ p_t \approx 0.2555$. When $1 \leq d_{SP} \leq P_1 + P_2 - 1,\ p_t \approx 0.1440$. It's obvious that the *recession* part of the sponsor node has almost an improvement with 16% in discovering the participant nodes, especially the nodes are in very low duty cycle.

5.2 Duty Cycle and Energy Consumption

Based on our assumption in Section 3, the energy consumption problem becomes the number of timeslots of a node is in transmit state or listening state in a period of time. We use E denote the energy consumption and DC denote the duty cycle. In this section, we will discuss the energy consumption of the sponsor node S.

Theorem 3. *The energy consumption in the working cycle I_S of the sponsor node S is $E_S = \sum_{i=1}^{k} \left\lceil \frac{t_S}{i} \right\rceil$, and the duty cycle is $\frac{\sum_{i=1}^{\lfloor \frac{t_S}{2} \rfloor} \left\lceil \frac{t_S}{i} \right\rceil}{t_S \cdot \lfloor \frac{t_S}{2} \rfloor}$, where $k = \left\lfloor \frac{t_S}{2} \right\rfloor$.*

Proof. Based on the description of ERUPT algorithm, the node S will broadcast messages in all timeslots in the *all-out* part. In *recession* part, node S will broadcast messages $\lceil \frac{t_S}{2} \rceil$ timeslots in the second time cycle t_S, and $\lceil \frac{t_S}{3} \rceil$ timeslots in the third and go on by the same analogy until the $\lfloor \frac{t_S}{2} \rfloor$th time cycle ends. The energy consumption of all time cycles in the working cycle of the node S is

$$E_S = t_S + \left\lceil \frac{t_S}{2} \right\rceil + ... + \left\lceil \frac{t_S}{\lfloor \frac{t_S}{2} \rfloor} \right\rceil = \sum_{i=1}^{\lfloor \frac{t_S}{2} \rfloor} \left\lceil \frac{t_S}{i} \right\rceil \quad (3)$$

We can get the duty cycle of node S is, $DC_S = \frac{E_S}{I_S} = \frac{\sum_{i=1}^{\lfloor \frac{t_S}{2} \rfloor} \left\lceil \frac{t_S}{i} \right\rceil}{t_S \cdot \lfloor \frac{t_S}{2} \rfloor}$

6 ANALYSIS THE MULTI-NODE CASE

In this section, we discuss the case where there are more than two nodes in the network. We assume the sponsor node is n_s, and the other $N-1$ nodes is $n_1, n_2, ..., n_{N-1}$. We will analyze our algorithm from the simple case and the real case two aspects. In the simple case, each node has the same time cycle and the participant nodes have the same duty cycle. The real case means each node has its own time cycle. On the other hand, when two nodes or more than two nodes broadcast messages at the same time, a collision will occur at the nodes in listening state. In this paper, we do not consider this case that there are two or more nodes in sponsor mode in the network.

6.1 The Simply Case

In this section, we will show the *discovery latency* of the ERUPT algorithm. Discovery latency defines the number of the nodes that the sponsor node has discovered in a certain timeslots. We let c denote the number of a certain timeslots. We use t denote the time cycle and DL denote the discovery latency.

Theorem 4. *In the simple case, the discovery latency has three cases.*
When $1 \leq c \leq t_l - 1$,

$$DL(c) = (N-1) \cdot \left[\frac{2c}{t_f + t_l} - \frac{c^3 - c}{3t_f t_l (t_f + t_l)} \right] \quad (4)$$

When $t_l \leq c \leq t - 1$,

$$DL(c) = (N-1) \cdot \left[\frac{-c^3 + c}{3t_f t_l (t_f + t_l)} + \frac{c^2}{t_f t_l} - \frac{(t_f + t_l)c}{t_f t_l} + \frac{t_f^2 + t_l^2 - t_f t_l - 1}{3t_f t_l} \right] \quad (5)$$

When $c > t - 1$,

$$DC(c) = 1$$

In equation (4) and (5), $t_f = \lfloor \frac{t}{2} \rfloor$ and $t_l = \lceil \frac{t}{2} \rceil$.

Proof. Based on Lemma 2, we know that all the participant nodes will be discovered in the *all-out* part of the working cycle of the sponsor node. When $c \leq t - 1$, we randomly choose n_r in $n_1, ..., n_{N-1}$. Then we focus on n_s and n_r. It's the same as two-node case and the time cycle of n_s is c. The discovery probability between two nodes is $p_{d_{SP}}$ showed in equation (1) and (2). We replace P_1, P_2 and d_{SP} with t_f, t_l and $t - c$ in the equation (1) and (2), where $t_f = \lfloor \frac{t}{2} \rfloor$ and $t_l = \lceil \frac{t}{2} \rceil$. Then we get the *discovery latency* showed in the equation (4) and (5), where $1 \leq c \leq t_l - 1$ and $t_l \leq c \leq t - 1$ respectively. When $c \geq t$, $DL(c) \equiv N - 1$.

6.2 The Real Case

In the daily life, each node have its own time cycle and duty cycle. *Discovery latency* depends on the choice of the time cycle of all nodes, especially the sponsor node. Based on Lemma 2, we find if $t_s \geq max\{t_1, t_2, ..., t_{N-1}\}$, the participant nodes will be discovered in the *all-out* part of n_s, where $t_s, t_1, t_2, ..., t_{N-1}$ denote the time cycle of $n_s, n_1, n_2, ..., n_{N-1}$ respectively. Then if we set the time cycle up to 100 timeslots, the sponsor node will discover all the participant nodes in above 2% duty cycle. It's hard to use a mathematical method to obtain the average discovery latency in this case. We will show it in the Simulation Section.

6.3 Duty Cycle and Energy consumption

In this section, we will show the energy consumption of the multi-node network.

Based on Theorem 3, the energy consumption in the sponsor node n_s is showed in equation (3). In the working cycle I_s, the participant node n_i at most have $\left\lceil \frac{t_s \cdot \lfloor \frac{t_s}{2} \rfloor}{t_i} \right\rceil$ time cycles, where $i = 1, 2, ..., N - 1$. So the worst case of the energy consumption of all the participant nodes is $2 \cdot \sum_{i=1}^{N-1} \left\lceil \frac{t_s \cdot \lfloor \frac{t_s}{2} \rfloor}{t_i} \right\rceil$. Put the energy consumption of the sponsor node and the participant nodes together, we get the energy consumption of the whole network is $E_{all} = \sum_{i=1}^{\lfloor \frac{t_s}{2} \rfloor} \left\lceil \frac{t}{i} \right\rceil + 2 \cdot \sum_{j=1}^{N-1} \left\lceil \frac{t_s \cdot \lfloor \frac{t_s}{2} \rfloor}{t_j} \right\rceil$.

7 SIMULATION

We evaluate the performance of ERUPT algorithm by running a ERUPT MAC protocol on NS2 [15] network simulator. In this section, about the deterministic neighbor discovery protocols (Disco, U-connect, Searchlight), the probabilistic Birthday protocol, and ERUPT protocol, we will show the tradeoffs of the energy consumption and the discovery latency in symmetric and asymmetric situation respectively.

When we compare the performance of the ERUPT algorithm with others, we focus on the energy consumption and the discovery latency of the sponsor node instead of all the nodes, since the sponsor node concerns more about the energy consumption and the discovery latency than the participant nodes in mobile social applications. In other protocols, all the nodes are the same. So we choose one node as the sponsor, then measure the energy consumption and the discovery latency of the sponsor. Based on the assumption of the energy consumption in Section 3, the absolute latency in terms of time units depends on the width of a timeslot which is based on platform and is the same for all protocols. So we use the number of active slots as the metric for the energy consumption. Active here means a node is in transmit state or listening state. In our simulation, we set the slot-width is $10ms$.

7.1 Symmetric

In the symmetric situation, we evaluate the performance of the ERUPT protocol under the situation that all the nodes in the network have the same time cycle. In this section, we will show the tradeoff of the energy consumption and the discovery latency between the ERUPT and other existing protocols when the duty cycle chooses 5%. In the simulation, we set 10 nodes in the network. To achieve a 5% duty cycle, Disco makes that each node randomly choose one pair of the primes $(37, 43)$, $(29, 67)$ and $(23, 157)$; U-Connect uses the prime 31; and Searchlight uses $t = 40$. In the ERUPT protocol, we use time cycle $t = 40$ for the sponsor node and the participant nodes. The cumulative distribution of the energy consumption and the discovery latency that one node (its the sponsor node in ERUPT) discovers other nine nodes for all protocols of 5% duty cycle (it's 20% duty cycle of the sponsor node in the ERUPT) is showed in Figure 4.

From Figure 4(a) and Figure 4(b), we get the cumulative energy consumption when the sponsor node has discovered all the other nine nodes is less than any other protocols in the ERUPT protocol, though the duty cycle of the sponsor node is almost 20% during its working cycle. From Figure 4(c), we get the discovery latency of the sponsor node of the ERUPT protocol is much lower than other protocols. Figure 4(d) shows a tradeoff between the energy

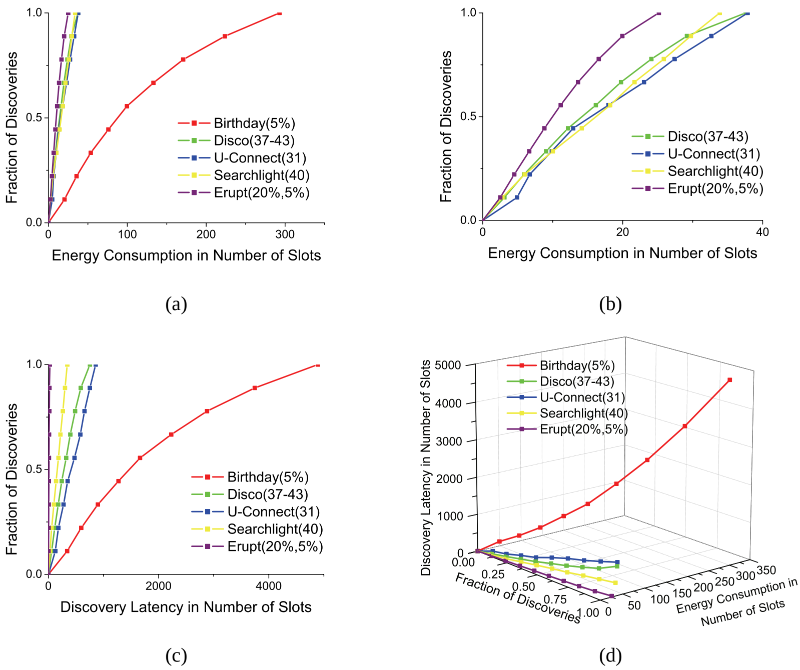

FIGURE 4
Figure (a) shows the energy consumption with the Birthday Protocol. Figure (b) shows the energy consumption without the Birthday Protocol. Figure (c) shows the CDF of the Discovery Latency-Symmetric. Figure (d) shows the CDF of the Energy Consumption and the Discovery Latency-Symmetric

consumption and the discovery latency of different protocols in the symmetric situation.

7.2 Asymmetric

In the asymmetric situation, we evaluate the performance of the ERUPT protocol in the situation that the duty cycle of the participant nodes are random chosen in 1% – 10%, and the time cycle of the sponsor is 20. (the duty cycle of the sponsor node is 30%). In the other protocols, we assume a sponsor node and set its duty cycle at 10%, and that of the other nodes are random chosen in 1% – 10%. The number of nodes in the network is set 10. The cumulative distribution of the energy consumption and the discovery latency that one node discovers other nine nodes (it's the sponsor node in ERUPT) for all protocols is showed in Figure 5(a) and Figure 5(b).

In Figure 5(a), the cumulative energy consumption of the sponsor node using Disco, U-Connect and ERUPT protocol are very close. On the other hand, ERUPT protocol has proven its competitive ability in the discovery

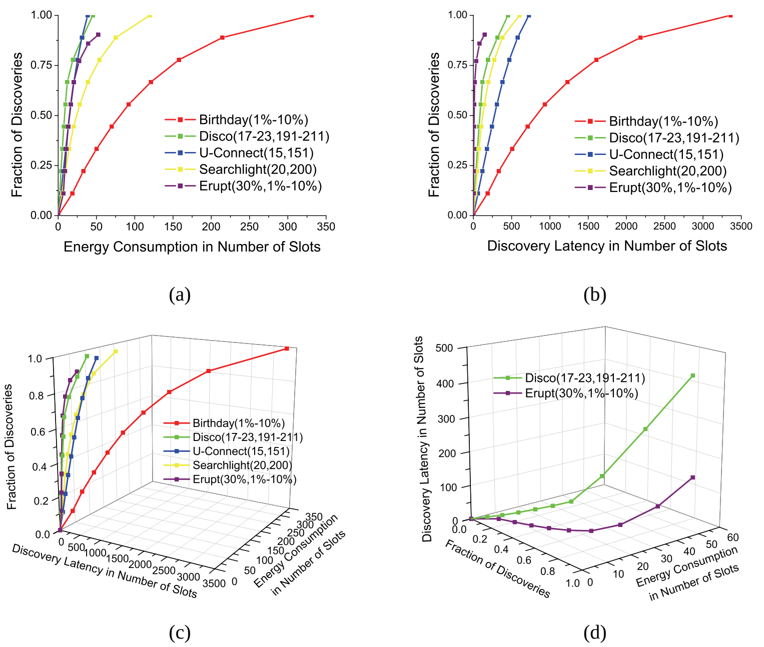

FIGURE 5
Figure (a) shows CDF of Energy Consumption-Asymmetric. Figure (b) shows CDF of Discovery Latency-Asymmetric. Figure (c) shows CDF of Energy Consumption and Discovery Latency-Asymmetric. Figure (d) shows ERUPT vs Disco in the Energy Consumption and Discovery Latency

latency of the sponsor node in Figure 5(b). Figure 5(c) shows a tradeoff between the energy consumption and the discovery latency of different protocols in the asymmetric situation. The performance of the ERUPT and the Disco protocol are very close in Figure 5(c). Then we extract the performance of these two protocols to compare. The result is showed in Figure 5(d). It's obvious that ERUPT performs a more cost-efficient neighbor protocol than Disco in this case.

We find there is no worse bound for the ERUPT protocol in this situation. It's suit for the users in daily life. The sponsor user just need to find appropriate numbers of neighbors to begin a social application. In our simulation, the probability of discovery of the sponsor in asymmetric situation is about 90%.

8 CONCLUSION AND FUTURE WORK

This paper proposes the ERUPT neighbor discovery protocol which is more suitable for daily life than existing protocols. The ERUPT protocol separates the nodes in the network into sponsor nodes and participant nodes two categories, since most mobile social applications have a sponsor. Our ERUPT protocol uses a recession strategy in neighbor detection to grasp the sponsors' philosophy that want to use a little more energy to exchange for less discover latency and more neighbors. ERUPT performs well in the energy consumption and the discovery latency of individual users in the simulation.

As future work, we would like to investigate how to choose a more reasonable length of the working cycle of the sponsor nodes to integrate neighbor discovery with more efficiency. In this paper, we assume the length is $\lfloor \frac{t}{2} \rfloor$ time cycles. Finally, we would like to overcome the collisions which occur at the participant nodes when there two or more sponsor nodes in the network. These collisions may infect the discovery latency and waste energy. So how to avoid the collisions is an important work.

ACKNOWLEDGMENT

The research of authors is partially supported by the Shanghai Foundation for Development of Science and Technology under grant No. 11JC1412800, the National Basic Research Program of China (973 Program) under grant No. 2010CB328101, the Integrated Project for Major Research Plan of the National Natural Science Foundation of China under grant No. 91218301, the National Natural Science Foundation of China (NSFC) under grant No. 61202383, Shanghai Rising-Star Program under grant No. 14QA1403700, the Program for New Century Excellent Talents in University (NCET) under grant No. NCET-12-0414, the Natural Science Foundation of Shanghai under grant No. 12ZR1451200, and the Research Fund for the Doctoral Program of Higher Education of China (RFDP) under grant No. 20120072120075.

REFERENCES

[1] Nintendo 3ds - streetpass. http://www.nintendo.com/3ds/hardware

[2] Sony ps vita - near. http://us.playstation.com/psvita

[3] McGlynn, M.J., Borbash, S.A.: Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, ACM (2001) 137–145.

[4] Dutta, P., Culler, D.: Practical asynchronous neighbor discovery and rendezvous for mobile sensing applications. In: Proceedings of the 6th ACM conference on Embedded network sensor systems, ACM (2008) 71–84.

[5] Kandhalu, A., Lakshmanan, K., Rajkumar, R.R.: U-connect: a low-latency energy-efficient asynchronous neighbor discovery protocol. In: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, ACM (2010) 350–361.

[6] Bakht, M., Kravets, R.: Searchlight: asynchronous neighbor discovery using systematic probing. ACM SIGMOBILE Mobile Computing and Communications Review **14**(4) (2011) 31–33.

[7] Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., *et al.*: A macroscope in the redwoods. In: Proceedings of the 3rd international conference on Embedded networked sensor systems, ACM (2005) 51–63.

[8] Liu, T., Sadler, C.M., Zhang, P., Martonosi, M.: Implementing software on resource-constrained mobile sensors: Experiences with impala and zebranet. In: Proceedings of the 2nd international conference on Mobile systems, applications, and services, ACM (2004) 256–269.

[9] Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: Proceedings of the 2nd international conference on Embedded networked sensor systems, ACM (2004) 95–107.

[10] Elson, J., Römer, K.: Wireless sensor networks: A new regime for time synchronization. ACM SIGCOMM Computer Communication Review **33**(1) (2003) 149–154.

[11] Paek, J., Kim, J., Govindan, R.: Energy-efficient rate-adaptive gps-based positioning for smartphones. In: Proceedings of the 8th international conference on Mobile systems, applications, and services, ACM (2010) 299–314.

[12] Ye, W., Heidemann, J., Estrin, D.: An energy-efficient mac protocol for wireless sensor networks. In: INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE. Volume 3., IEEE (2002) 1567–1576.

[13] Lai, S.: Heterogenous quorum-based wakeup scheduling for duty-cycled wireless sensor networks. PhD thesis, Virginia Polytechnic Institute and State University (2009).

[14] Hardy, G.G.H., Wright, E.M.: An introduction to the theory of numbers. Oxford University Press (1979).

[15] Issariyakul, T., Hossain, E.: An introduction to network simulator NS2. Springer (2012).