Proc. International Workshop on Social Sensing (SocialSens), Pittsburgh, PA, April 21, 2017, pp. 1-6
doi: 10.1038/srep41750

Identifying the Space Buddies to Track Lost Items

Eyuphan Bulut
Department of Computer Science
Virginia Commonwealth University
401 West Main St.
Richmond, Virginia 23284
ebulut@vcu.edu

Boleslaw K. Szymanski
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th St.
Troy, New York 12180
szymansk@cs.rpi.edu

ABSTRACT

Locating missing or lost objects has always been a challenging task. RFID technology and participatory sensing based approaches have offered solutions but often their adoption was limited due to the high hardware costs or low active participation problem. With the introduction of iBeacon technology and smartphones having BLE capability, tracking such objects has become easier and cost-effective. Objects of care are labeled by attaching to them affordable iBeacon tags, and smartphones in the proximity of these tags sense their presence opportunistically through the applications running in the background. In this paper, we study the tracking of lost objects through the collaboration among users. We analyze the visit patterns of users at the same locations and develop a metric that quantifies for each user the potential benefit of others in terms of their capability of finding that user's lost objects. Depending on the predicted benefits, each user's preference list of other users is formed and then utilized to identify the space buddies who can best track her lost items. The identification is based on the adaption of the solution to the roommate matching problem. We apply the proposed system to two different location based social network datasets and show its effectiveness in different settings.

CCS CONCEPTS

•**Networks** →**Network mobility;** *Network protocol design; Network design and planning algorithms; Mobile ad hoc networks;* Network performance analysis;

KEYWORDS

Location tracking, social mobility analysis, lost item tracking, matching.

ACM Reference format:
Eyuphan Bulut and Boleslaw K. Szymanski. 2017. Identifying the Space Buddies to Track Lost Items. In *Proceedings of The 2nd International Workshop on Social Sensing, Pittsburgh, PA USA, April 2017 (SocialSens 2017),* 6 pages.
DOI: 10.1145/3055601.3055611

DOI: 10.1145/3055601.3055611

1 INTRODUCTION

The statistics reported by different reports [24, 31] show that an average person spends 15-20 minutes per day (i.e., around a year in his life) to search for his misplaced or lost items. On average nine items per week are misplaced and the most frequent ones are mobile phones, car/house keys and sunglasses. Several books [26] have been written on this topic and suggestions made to mitigate the impact of such forgetfulness in our lives. Moreover, when we lose things outside, it is much harder and takes longer to locate them as the search area is much larger. With the advancing Internet of Things (IoT) era and widespread proliferation of smartphones, several solutions have been provided to take the advantage of enhancing technology. Smartphones that are capable of Bluetooth Low Energy (BLE) have been recently used to locate items to which iBeacon tags are attached. Several commercial products have been recently released [12, 30] to ease the process of locating the missing items.

These iBecaon tags are wirelessly connected to the smartphones through BLE and they communicate with the device and maintain the information about the presence of the items with tags attached in vicinity. However, the benefit of such tags is lost when they are out of the range of smartphones (e.g., lost). In order to enhance the benefit, a social network of users with similar application running on their device can be formed to enable collaborative localization of lost items. When an item is lost, any nearby smartphone in this group of users can sense the iBeacon tag attached to that item in a transparent way and communicate to the server and eventually to the user who owns it. The geographical coordinates of the smartphone detecting the iBeacon is then utilized to locate the lost item. There could be some concerns about privacy management and battery utilization due to the sensing and providing the user location to the server, but this is a collaborative effort and users can potentially mutually benefit from it. Moreover, commercial producers of such systems do not release the list of such users in contrast to other applications with different purposes (e.g., FireChat [1]).

Such a collaborative sensing system could be utilized for multiple purposes including public safety and emergency preparedness (e.g., child protection and tracking [29]), a national priority issue. Clearly, the benefit of such systems will be enhanced as the number of users participating in the system increases (e.g., campus). For social efforts like finding missing children, there could be enough motivation for users to voluntarily participate in the system and ignore the violation of privacy to some extend. However, when there is no clear incentives and privacy protection methods (e.g., hiding the location information of users), and the user input is not automatically received by the system (i.e., users need to manually provide the data), it could be hard and risky to practically deploy such systems. Thus, adaptive systems that can perform with minimal user interaction and incentive requirements could avoid the potential privacy violation risks and direct user participation.

In this paper, we study the sensing of lost objects (with iBeacon tags) through the best space buddies of users rather than all users in the network having a smartphone with BLE capability. Our contributions are (i) analysing user location visit patterns and developing a metric that can quantify the potential benefits of users in terms of their enhanced ability to find their belongings, (ii) identifying for each user the best space buddy that can track its lost items by adapting the roommate matching algorithm based on the proposed metric, and (iii) performing simulations on two different location based social network dataset to show the algorithm effectiveness with different set of parameters.

The rest of the paper is organized as follows. We discuss the related work in Section 2. In Section 3, we first define a metric to analyze the relation between the visit patterns of nodes at the same location, then we discuss the proposed matching algorithm to find the best space buddy of each user to find their lost belongings. In Section 4, we describe the simulation setting and evaluation of proposed system using real location based social network traces. Finally, we offer conclusions in Section 5.

2 RELATED WORK

Sensing through the mobile devices possessed by people has attracted a lot of interest recently. It has been studied under different names including people centric-sensing [10], participatory sensing and mobile crowd sensing [15]. Smartphones which are equipped with multiple sensors have offered tremendous opportunity for sensing the surrounding without the need for the dedicated devices or supporting their mobility.

Bluetooth Low Energy (BLE) or Bluetooth 4.0 is designed to operate at low data rates with low power consumption and low manufacturing costs. Such a design offered high (i.e., 60-80%) power savings for devices, and let them operate using a coin cell battery for several months to years without replacement [3]. BLE has become more popular after Apple devised the iBeacon standard protocol in 2013 [5]. This also paved the way for other device manufacturers to support BLE.

Beacons are BLE devices with a main purpose of advertising itself to be discovered by other BLE capable devices (so providing location based services to BLE devices). As the format of advertisement packets allows, sometimes additional data available on the device is shared so that nearby devices can collect that data (e.g., sensed information) without making a connection. The advertisements are repeated after constant intervals (e.g., Apple's iBeacon standard calls for an optimal broadcast interval of 100 ms) to let the nearby devices easily detect the beacon.

Recently, the popularity of beacons have been increasing and several applications exploiting iBeacon and BLE functionalities in different contexts have been developed. These include indoor location positioning [22, 28] and navigation [11, 33]), proximity marketing [4], ticketing [14] and possession tracking or localizing missing items. In indoor positioning, as BLE allows direct signal

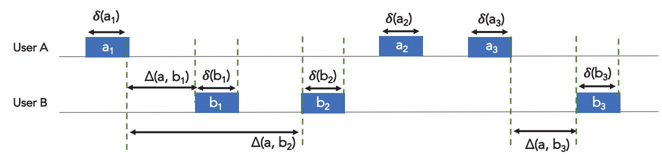

Figure 1: Sample visit patterns of two users

strength measurement, an RSSI (Received Signal Strength Indication) value can further be utilized to find the distance of the sensed objects and to improve accuracy of item's predicted location. For example, in [25], authors propose a search capability for physical objects inside furniture at home or office. Tags are used to make objects searchable while all other localization components are integrated into furniture.

Beacons can offer more convenient solutions compared to the other technologies like QR codes and NFC, as they require the least interactions with users. Compared to RFID based localization [18, 27], beacons also offer advantage of being easily detected by the most of the smartphones.

In an iBeacon based possession or lost item tracking system with a collaborative approach, the smartphone application creates a social network of users and let them identify the coordinates of the beacons (attached to lost item) from the smartphone that detects it. By this way, even though the users are not in the range of their possessions, they could be sensed in a transparent fashion (without active involvement of other users in the network). In [23], a prototype is implemented to let the users identify and localize their personal objects using beacons. There are also commercial devices developed for specific purposes such as child and pet tracking [12].

3 PROPOSED SYSTEM AND SOLUTION

In this section, we present the details of the proposed system. First, we introduce a metric to quantify the relation between the visits of two different users at a region. Then, we study the assignment of users to each other through a stable matching algorithm using their preference lists.

3.1 Metric definition

We define a visit of a location by a user with a visit event $a = (t_s, t_e, loc_{id})$ as a 3-tuple in which t_s and t_e denote the start and end times of the visit and loc_{id} denotes the id of the location visited. The historical visits of a user at a specific location could be defined as the set of visit events, where the previous event's end time is always smaller than the start of the next event.

$$\mathbb{V}_a = \{a_1, a_2, a_3, \ldots, a_n\} \text{ where } a_i.t_e < a_{(i+1)}.t_s, \forall i \in \{1..n\}$$

A mobile device can detect the iBeacon attached item within certain proximity. In order to take this into account, we use a probability factor p, which denotes the probability that a user's mobile device can detect the lost item in the same location at the current time unit.

To quantify the benefit of a user B to another user A in terms of finding his/her lost items, we propose a metric called *Social Tracking*

Distance (STD), inspired by the metrics [7, 8] used in analyzing contact patterns in DTNs. Consider the sample visit history of two nodes A and B in a location shown in Fig. 1. The upper part of the figure shows the visits of user A and the lower one shows the visits of node B. The i^{th} visit of user A and B is labeled as a_i and b_i, respectively. The durations of visits are denoted with $\delta(.)$ and the time passed since the last visit of user A to the user B's i^{th} visit is denoted as $\Delta(a, b_i)$. Assume that there are n visits of node A and node B in a specific area. Moreover, without loss of generality, assume that $b_1.t_s > a_1.t_e$ and $b_n.t_e < a_n.t_s$. We define the $STD_{(A,B)}$ metric as the probabilistic delay that user B's device will sense the lost item of user A and denote by:

$$STD_{(A,B)} = \sum_{x=1}^{n} \frac{d_{(a_s,b_x)}}{p(\delta(b_x))\beta(b_x)} \qquad (1)$$

For each visit of user B, we find the last visit of node A before B's that visit and calculate the time difference for each possibility of losing and finding times. More specifically, here, $d_{(a_s,b_x)}$ is the average delay of finding an item that might have been lost anytime during A's last visit (before B's x^{th} visit) and found anytime during B's x^{th} visit. Here, s is found using:

$$s = \arg\max_i \{a_i.t_e < b_x.t_s\}$$

In the formula, $p(\delta(b_x))\beta(b_x)$ denotes the probability of finding an item during B's x^{th} visit. For a visit of duration d, the probability that the item will be found by the end of duration is:

$$p(d) = 1 - (1-p)^d$$

However, if there are multiple visits from user B to the area before user A visits the area, the probability that the items will be sensed in subsequent visits depends on the probability that the item will not be detected in previous visits (denoted by $\beta(b_x)$):

$$\beta(b_x) = \prod_{\forall k<x} (1-p)^{\delta(b_k)} = (1-p)^{\sum_{\forall k<x} \delta(b_k)}$$

Then,

$$\begin{aligned} d_{(a_s,b_x)} &= \beta(b_x) \sum_{i=1}^{\delta(a_s)} \sum_{j=1}^{\delta(b_x)} (\Delta(a, b_x) + i + j) p_x(j) \\ &\quad \text{where, } p_x(j) = p(1-p)^{j-1} \\ &= \beta(b_x) \left(\Delta(a, b_x) + \frac{\delta(a_s)}{2} + 1/p - \right. \\ &\quad \left. \frac{\delta(b_x)}{p(\delta(b_x))} + \delta(b_x) \right) \end{aligned}$$

We assume that when user A loses something during a visit, she will not find it in the same visit but will definitely be able to find it in her next meeting. Thus, we exclude the possibility of item's detection by the same user A in the same time frame it is lost and consider the distance of node B's visits with only last visit of user A. User A can lose the item at any time point during her visit (in range $(0, \delta(a_i)]$) and user B's device can sense the lost item at any time during her visit (in range $(0, \delta(b_i)]$). Thus, the delay for sensing and finding the lost item could be in range $(\Delta(a, b_i), \Delta(a, b_i) + \delta(a_i) + \delta(b_i)]$. However, the probability of each will be different, and can be calculated based on the duration of B's visit (j) in the location using $p_x(j) = p(1-p)^{j-1}$.

If user A visits the location multiple times before user B visits, there will be no benefit of user B in sensing the lost items between two consecutive user A visits without having a user B visit (as user A will definitely find the lost item before B per our assumption).

Note that upper part of the fraction in Eq. 1 is the average delay in the specific case of losing an item at visit a_s and finding it in visit b_x. The lower part is the probability that the item will be found in this specific case. We are dividing the delay by probability to get STD metric so that expected probabilistic delay in such visits could be retrieved (which indeed shows the real benefit of user B to A in that case).

Once the probabilistic delays are calculated, we define a weighted satisfaction value for the efforts of user B in finding user A's lost items in any of the locations A visits. Note that not only the frequency of location visits of a user is significant but also the duration of the visit, and the distribution of all visits within a time frame has impact. Also, there may not be a visit by user B between two consecutive visits by A or visits of B may not continue even if A continues visiting the location. Moreover, at different locations, the STD value may be different for the same pair of nodes. To take into account such differences, the satisfaction value is averaged over all regions (r).

$$\gamma_{(A,B)} = \sum_{\forall r} \left(w(r) \left(\frac{Cov(A,B)_r}{STD_{(A,B)}} \right) \right)$$

where, $w(r)$ is the weight of region r (i.e., total visit duration within all visits in all regions) and

$$Cov(A,B)_r = \frac{\sum_{x=1}^{n} \delta(a_k) I[x]}{\sum_{x=1}^{n} \delta(a_k)} \text{ where,}$$

$$I[x] = \begin{cases} 1, & \text{if } x = \arg\max_i \{a_i.t_e < b_j.t_s\} \exists j \\ 0, & \text{otherwise} \end{cases}$$

3.2 Matching to the Best Spatial Buddies

In order to find out the lost items in an iBeacon based tracker system, we study the matching of people in a community that can help each other the most. Assume that there are N nodes in a network and R possible locations they visit. While these locations could be considered as all the possible locations with well-defined boundaries that users visit, they can simply be considered as the mostly visited locations or the places with some likelihood that users lose their items there. Each node visits all or some of these locations with different frequencies and visit durations.

Once each node (e.g., A) calculates total satisfaction ($\gamma_{(A,B)}$) from every other node (e.g., B) using the visit history, it forms a preference list of other users in terms of their support to track and locate his lost items. After the preference list of each node is determined, in order to maximize the total benefit in the entire network of people, they need to be assigned to the trackers as much as possible from the top of their lists. In order to solve such a matching, we formulate the problem as *stable roommate matching problem* (SRP) in which the matching is stable if there are no two nodes which are not roommates and which both prefer each other to their assigned roommate under the current matching. Note that this problem is distinct from the stable-marriage problem as the stable-roommates problem allows matches between any two nodes, not just between two disjoint classes such as men and women [2]. Note that since the matchings can be asymmetric, N does not need to be even. To solve the problem and guarantee a stable matching, if any, we adapt the Irving's algorithm [20] to our problem.

Algorithm 1 Find-the-Phase1-Reduced-List
Input: N, preferenceList [ ][ ] pL
Output: Reduced list of preferences

```
1:  for each node i in N do
2:      proposalAccepted[i] ← 0
3:      nextToAsk[i] = 1
4:  end for
5:  while there exists a user whose proposal not accepted do
6:      i ← smallest i whose proposal not accepted
7:      c = pL[i][nextToAsk[i]]
8:      if accepted[c] == nil then
9:          accepted[c] = i
10:         proposalAccepted[i] = 1
11:     else if order(pL[c], i) < order(pL[c],accepted[c]) then
12:         proposalAccepted[accepted[c]] = 0
13:         rejected[c][accepted[c]] = 1
14:         nextToAsk[accepted[c]]++
15:         accepted[c] = i
16:         proposalAccepted[i] = 1
17:     else
18:         rejected[c][i] = 1
19:         nextToAsk[i]++
20:     end if
21: end while
22: for each node i in N do
23:     proposer = accepted[i]
24:     for each j with order(pL[i], proposer) < order(pL[i], j) do
25:         rejected[i][j] = 1
26:         rejected[j][i] = 1
27:     end for
28: end for
```

Algorithm 1 shows the steps of the first phase in which a reduced preference list is obtained. Until there is no user whose proposal is not accepted by someone, we process the next user (i) with the smallest index. The user i proposes to next user (c) in its preference list to which it has not proposed yet (Line 7). If that user has not been proposed by someone else before, it immediately accepts this proposal (Lines 8-10). If it accepted a previous proposal by some other user (i.e., accepted[c]), user c checks if it prefers the new user i more than the node with which it is currently matched. If that is the case, it rejects the previous proposer and accepts this user i's proposal. Previous proposer then needs to propose the next user in its list (Lines 11-16). If the old proposer is preferred over the new one, user c rejects the proposal of user i, then user i needs to propose to the next one in its list (Lines 17-20). This process stops when all users has some user that accepted its proposal. Then, the preference lists are reduced by deleting the not possible matchings. To this end, each user rejects the other users whose order come later in its preference list than the user whose proposal it is holding. Similarly, those rejected users in turn reject the node that rejected them, so that no matching will be possible between the nodes involved (Line 22-28). By the end of this phase, the preference lists of users are reduced. If there is a situation in which each user has only one remaining user in their preference lists, then the stable matching is reached. If there are more than one users at least in one of the user's preference list, phase 2 algorithm should be run.

Algorithm 2 Find-the-Phase2-Matchings
Input: N, reducedPreferenceList [ ][ ] rpL
Output: Stable single matching or not existing

```
1: while ∃i |rpL[i]| > 1 & ∀i |rpL[i]| >= 1 do
2:     Find a cycle ℂ =<p_i, q_i, p_{i+1}, q_{i+1}, ... q_{s−1}, p_s> s.t.
       p_i = a user with more than one not rejected user in RPL
       q_i = second user in rpL[p_i]
       p_{i+l} = last user in rpL[q_i]
       p_s = p_i
3:     Reduce the cycle
       ∀i ∈ ℂ rejected[q_i][p_{i+1}]=1
4: end while
5: if ∀i there is only one user remained in rpL[i] then
6:     return matching 𝕄
7: else
8:     No matching found
9: end if
```

Algorithm 2 shows the steps of the second phase of the Irving's algorithm. In this phase, the preference lists are further reduced to find a stable matching. All-or-nothing cycles are used to reduce the lists. Such cycles are defined as the sequence of users $<p_i, q_i, p_{i+1}, q_{i+1}, \dots q_{s-1}, p_s = p_i>$ such that q_i is the second user in the preference list of user p_i and p_{i+1} is the last user in q_i's preference list. The cycle ends when the last discovered p_s becomes the same with the starting point, p_i. In order to find such a cycle sequence, the algorithm starts with the user who has at least two users (not rejected) in reduced preference list. Once the cycle is found, all q_i users occurring in the sequence rejects user p_{i+1}. There could be multiple such cycle removal process. At the end, the algorithm stops either when list of users includes only one user (i.e. stable matching) or when the preference list becomes empty (i.e., there is no matching found).

Note that by assigning each user to another single user with highest chance of finding his lost items, our goal is to minimize the user involvement and to avoid the potential privacy violation risks. As only one best spatial buddy is used for that purpose, the benefit of a single such user will be limited compared to the cumulative collaborative benefit of all users in the network. The matching algorithm could be extended with more than one user assignment to each other so that a good number of users could be found to achieve as high likelihood of finding the lost items as the likelihood that all users can provide. In that case, extensions of roommate matching problem with room sizes more than two could be considered. However, even the triple room extension [21] of the problem is NP-complete. Thus, looking for heuristics based algorithms will be the subject of our future work.

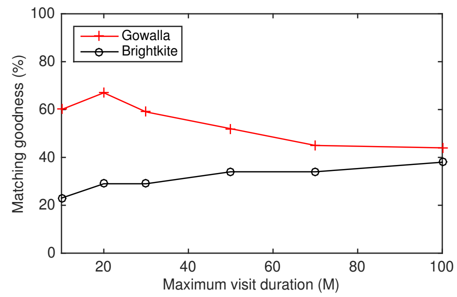

Figure 2: Matching goodness with different maximum visit durations.

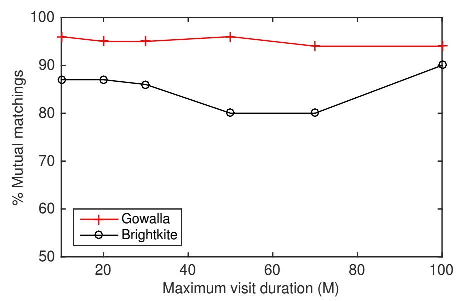

Figure 3: Percentage of mutual matchings with different maximum visit durations.

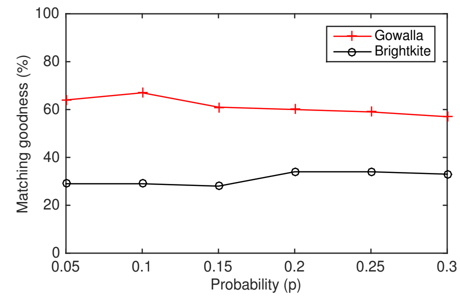

Figure 4: Matching goodness with different probabilities.

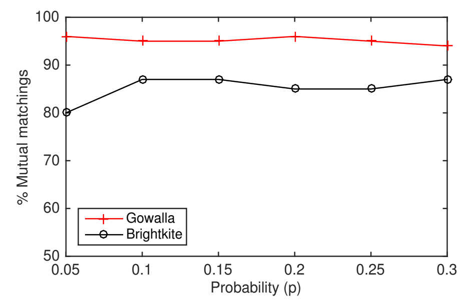

Figure 5: Percentage of mutual matchings with different probabilities.

	Gowalla	Brightkite
Total user count	6187	3331
Top user count matched	294	140

Table 1: Comparison of two datasets.

4 SIMULATION RESULTS

In this section, we present the results of simulations performed on the proposed system. We used two different online location-based social network datasets to capture the user visits to different locations. Specifically, we used the Gowalla and Brightkite datasets [13] and considered the check-ins as the start of the location visits. As there were no check-outs available for the locations, we considered randomly decided durations from a visit duration range. The datasets provide location ids in addition to the coordinates of the locations, thus we determined the visits from different users using these information.

In order to restrict us to a geographical area, we focused on the check-ins that are reported in San Francisco area. We first calculated the relations and probabilistic tracking distance between all pairs of users. Then, we found the preference lists of nodes, and the assignment of each node to another node using the Algorithm 1 and Algorithm 2.

Table 1 shows the comparison of two datasets. As there are many users with smaller number of visits and only distributed to few number of areas, we used only the top users with more instances of data. 294 and 140 users are used in these dataset, respectively.

First, we look at the goodness of matchings. We define the goodness of a matching assignment as the ratio of obtained benefit (i.e. satisfaction) with that assignment to the maximum possible benefit that would be achieved if the users could choose first nodes in their preference lists. Figure 2 shows the change of matching goodness with different visit durations. In x-axis of the graph, we show the upper limit of the range that we use to decide the visit durations of each user. Each visit duration is randomly determined in range $[0, M]$, and M is value shown in x-axis. As the figure shows, matching goodness starts to decrease after some M value in Gowalla, while it increases in Brightkite. Figure 3 shows the impact of maximum visit duration on percentage of mutual matchings among all matchings. In terms of reducing potential privacy violation, high percentage of mutual matchings are preferred compared to asymmetric matchings. As the figure shows, the percentage of mutual matchings is pretty stable within the given range of durations for the Gowalla dataset. There is some changes for Brightkite dataset during middle M values. We will investigate the factors that affects this in detail in our future work.

Next, we look at the impact of probability, p, used for detection of items. Figure 4 shows the change of goodness with different p values. As p increases the impact on goodness is different on two datasets. It slightly decreases in Gowalla, and increases in Brightkite. Similarly, in Figure 5, we show the change in percentage of mutual matchings with different p values. In Brightkite results, we see some increase as p increases, but not a remarkable change is seen in Gowalla. In our future work, we will analyze the root causes of different impacts of probability and duration on different datasets.

5 CONCLUSION

In this paper, we study the tracking of lost objects through the collaboration among users. We match each user to others, whom we termed space buddies, in a way that may not be symmetric but which maximizes benefits from the matching. Analyzing the visit patterns of users at the same location, we introduce a metric called Social Tracker Distance (STD) that quantifies the benefit of potential space buddies in terms of their capability of finding the user's lost objects. Once each user determines the preference list of other users based on this metric, the roommate matching problem is used to find the space buddies of each user. In simulations, we applied the proposed matching to two different location based social network datasets. Based on the changes on visit duration and probability p, we look at the goodness of matchings and the percentage of mutual relationships in all matchings, which are more desired in terms of reducing privacy violation than asymmetric matchings.

The proposed idea of matching the users with their space buddies helps minimizing the risks of privacy violations as only two users interact with each other and moreover share their locations. However, in order to increase the benefit and get close to the aggregate benefit from all users multiple space buddies could be selected. In our future work, we will look at this and the impact of other parameters in the simulation setting. We will also integrate mobility pattern prediction algorithms [6, 16, 17] to detect the space buddies based on nodes' future movements. Moreover, we will consider the network community structure [9, 32] and different frequency of demands from users [19] to find their lost items to optimize the network level performance of finding lost items.

ACKNOWLEDGMENTS

This work was partially supported by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS CTA).

REFERENCES

[1] *Open Garden.* https://www.opengarden.com/firechat.html.
[2] *Stable matching problem.* https://en.wikipedia.org/wiki/Stable_roommates_problem.
[3] 2015. *Bluetooth and beyond.* http://www.telecomreview.com/index.php?option=com_k2&view=item&id=1545:bluetooth-and-beyond&Itemid=427.
[4] Navalkrushna Allurwar, Balasaheb Nawale, and Swapnesh Patel. 2016. Beacon for proximity target marketing. *Int. J. Eng. Comput. Sci* 15 (2016), 16359–16364.
[5] Apple. 2016. iBeacon for Developers. (2016). https://developer.apple.com/ibeacon/
[6] Eyuphan Bulut, Sahin Cem Geyik, and Boleslaw K Szymanski. 2010. Efficient routing in delay tolerant networks with correlated node mobility. In *Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on.* IEEE, 79–88.
[7] Eyuphan Bulut, Sahin Cem Geyik, and Boleslaw K Szymanski. 2014. Utilizing correlated node mobility for efficient DTN routing. *Pervasive and Mobile Computing* 13 (2014), 150–163.
[8] E. Bulut and B. K. Szymanski. 2012. Exploiting Friendship Relations for Efficient Routing in Mobile Social Networks. *IEEE Transactions on Parallel and Distributed Systems* 23, 12 (2012), 2254fi?!2265.
[9] Eyuphan Bulut, Zijian Wang, and Boleslaw K Szymanski. 2009. Impact of social networks on delay tolerant routing. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE.* IEEE, 1–6.
[10] Andrew Campbell, SB Eisenman, ND Lane, E Miluzzo, RA Peterson, H Lu, X Zheng, M Musolesi, K Fodor, and GS Ahn. 2009. The Rise of People-Centric Sensing.. In *ICDCN.* Citeseer, 9.
[11] Hsuan-Eng Chen, Yi-Ying Lin, Chien-Hsing Chen, I Wang, and others. 2015. BlindNavi: a navigation app for the visually impaired smartphone user. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems.* ACM, 19–24.
[12] Trax Play GPS: Child and Pet Tracking. 2016. (2016). https://traxfamily.com/
[13] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 1082–1090.
[14] Rui Couto, João Leal, Pedro Maurício Costa, and Teresa Galvão. 2015. Exploring Ticketing Approaches Using Mobile Technologies: QR Codes, NFC and BLE. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems.* IEEE, 7–12.
[15] Raghu K Ganti, Fan Ye, and Hui Lei. 2011. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine* 49, 11 (2011).
[16] Sahin Cem Geyik, Eyuphan Bulut, and Boleslaw Szymanski. 2010. PCFG based synthetic mobility trace generation. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE.* IEEE, 1–5.
[17] Sahin Cem Geyik, Eyuphan Bulut, and Boleslaw K Szymanski. 2013. Grammatical inference for modeling mobility patterns in networks. *IEEE Transactions on Mobile Computing* 12, 11 (2013), 2119–2131.
[18] Josef Hallberg, Chris Nugent, Richard Davies, and Mark Donnelly. 2009. Localisation of forgotten items using RFID technology. In *Information Technology and Applications in Biomedicine, 2009. ITAB 2009. 9th International Conference on.* IEEE, 1–4.
[19] Buster O Holzbauer, Boleslaw K Szymanski, and Eyuphan Bulut. 2014. Impact of socially based demand on the efficiency of caching strategy. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on.* IEEE, 401–406.
[20] Robert W Irving. 1985. An efficient algorithm for the fistable roommatesfi problem. *Journal of Algorithms* 6, 4 (1985), 577–595.
[21] Kazuo Iwama, Shuichi Miyazaki, and Kazuya Okamoto. 2007. Stable roommates problem with triple rooms. In *Proc. 10th KOREA-JAPAN Joint Workshop on Algorithms and Computation (WAAC 2007).* 105–112.
[22] Paul Martin, Bo-Jhang Ho, Nicholas Grupen, Samuel Muñoz, and Mani Srivastava. 2014. An iBeacon primer for indoor localization: demo abstract. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings.* ACM, 190–191.
[23] Silvia Mirri, Catia Prandi, Paola Salomoni, and Lorenzo Monti. 2016. Social Location Awareness: A Prototype of Altruistic IoT. In *New Technologies, Mobility and Security (NTMS), 2016 8th IFIP International Conference on.* IEEE, 1–5.
[24] US News. 2016. . http://www.usnews.com/.
[25] Jens Nickels, Pascal Knierim, Bastian Könings, Florian Schaub, Björn Wiedersheim, Steffen Musiol, and Michael Weber. 2013. Find my stuff: supporting physical objects search with relative positioning. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing.* ACM, 325–334.
[26] John Ortberg. 2009. *When the Game is Over, it All Goes Back in the Box.* Zondervan.
[27] Mana Sasagawa, Kaori Ikematsu, and Itiro Siio. 2016. Simply tag and find: finding indoor items by using detection history of RFID tags. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct.* ACM, 365–368.
[28] David Schwarz, Max Schwarz, Jörg Stückler, and Sven Behnke. 2014. Cosero, find my keys! Object localization and retrieval using Bluetooth Low Energy tags. In *Robot Soccer World Cup.* Springer, 195–206.
[29] Sejun Song, Sunae Shin, Younghwan Jang, Seoungjin Lee, and Baek-Young Choi. 2015. Effective Opportunistic Crowd Sensing IoT System for Restoring Missing Objects. In *Services Computing (SCC), 2015 IEEE International Conference on.* IEEE, 293–300.
[30] The Tile. 2016. . https://www.thetileapp.com/.
[31] News Week. 2016. . http://www.newsweek.com/.
[32] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. 2013. Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)* 45, 4 (2013), 43.
[33] Jingjing Yang, Zhihui Wang, and Xiao Zhang. 2015. An ibeacon-based indoor positioning systems for hospitals. *International Journal of Smart Home* 9, 7 (2015), 161–168.