SCPS: A Social-aware Distributed Cyber-Physical Human-centric Search Engine

Jianwei Liu, Haiying Shen, Ze Li
Department of Electrical and Computer Engineering
Clemson University Clemson, SC 29634
{jianwel, shenh, zel}@clemson.edu

Shoshana Loeb, Stanley Moyer
Telcordia Technologies
Piscataway, NJ 08854
{shoshi, stanm}@research.telcordia.com

***Abstract*—One critical function of cyber-physical systems (CPS) is object search in the physical world through the cyber sphere that enables interaction between the cyber and physical spheres. Some of the previously proposed physical object search engines use RFID tracking, and others collect the information of object locations into a hierarchical centralized server. The difficulty of widely deploying RFID devices, the centralized search, and the need for periodical location information collection prevent CPS from achieving higher scalability and efficiency. To deal with this problem, we propose a Social-aware distributed Cyber-Physical human-centric Search engine (SCPS) that leverages the social network formed by wireless device users for object search. Without requiring periodical location information collection, SCPS locates objects held by users based on the routine user movement pattern. Moreover, using a social-aware Bayesian network, it can accurately predict the users' locations when exceptional events (e.g., inclement weather) occur, which breaks user movement pattern. Thus, SCPS is more advantageous than all previous social network based works which assume that user behaviors always follow a certain pattern. Further, SCPS conducts the search in a fully distributed manner by relying on a DHT structure. As a result, SCPS achieves high scalability, efficiency and location accuracy. Extensive real-trace driven simulation results show the superior performance of SCPS compared to other representative search methods.**

I. Introduction

Advances in ubiquitous sensing, computing and wireless-communication technologies are leading to the development of cyber-physical systems (CPS), which promise to revolutionize the way we interact with the physical world. CPS are computer systems that monitor and interact with a constantly changing physical environment. While many technologies are important to achieving high performance CPS, perhaps one of the most essential challenges is object search in the physical world through the cyber sphere that enables interaction between the cyber and physical spheres. The problem dealt in this paper is human-centric object search. That is, *how to efficiently search objects carried by people (such as documents, keys and electronic files) in the physical world through a computer system?* CPS applications, such as healthcare monitoring, are becoming increasingly prevalent, and involve ubiquitous users and objects scattered over a wide area. This requires that a search engine can provide *scalable, efficient, low-latency* search service with *high location accuracy*.

The number of mobile devices with ad hoc wireless communication capacities (e.g., WiFi and Bluetooth) has been increasing rapidly. The mobile device users constitute a social network, in which human mobility exhibits certain patterns, and is predictable to a large extent [1]. Also, individuals are tied by one or more specific types of relationship, such as friendship, kinship or trade. In addition, wireless sensors are widely deployed for monitoring the environment, like weather and traffic. The increasing number of mobile users, wireless sensor nodes, and base stations (BSs) now creates opportunities for innovations in human-centric object search.

By leveraging these opportunities, in this paper, we propose a Social-aware distributed Cyber-Physical human-centric Search engine (SCPS), which does not specifically depend on additional RFIDs or sensor devices. Without periodical location information collection, SCPS locates objects held by users based on the user movement patterns in the social network. Moreover, using a Bayesian network (BN) combined with a social network (i.e., social-aware BN), it can find the users' locations when exceptional (i.e., non-routine) events (e.g., bad weather, traffic jam and meeting a friend) occur, which breaks routine user movement patterns. For example, a person does not play football as planned when it is raining but goes to the gym instead. If two friends[1] bump into each other, they may stay together for a time period. Further, SCPS conducts the search in a fully distributed manner by relying on a Distributed Hash Table (DHT) structure constituted by BSs, which search objects for their nearby mobile users. SCPS is distinguished by its high scalability, efficiency, and accuracy and low latency in object searching.

II. Related Work

Physical object search. In recent years, a number of methods for physical object search or localization have been proposed. MAX [2] is perhaps the first search engine for physical objects. It has a centralized hierarchy formed by station and substation for object searching. Substations can sense the RFIDs attached to the objects nearby, and they are responsible for building an inverted index[2] of their nearby objects for local search. The central station stores the inverted index of substations. Nodes send queries to the substations,

[1]Two persons with a direct relationship (e.g., co-workers, family members, classmates and business partners) in a social network are called friends.

[2]Inverted index is a data structure, which shows the mapping of objects to their owners.

which send back the IDs of nodes containing the searched keywords. Queries that cannot be met locally are directed to the central station, which returns the best matched substations. Snoogle [3] extends MAX by adding schemes for supporting multiple-keyword search and top-k query. Microsearch [4] further details the design and implementation of the top-k search and also presents a memory efficient algorithm and a theoretical model of the search. However, the centralized structure in MAX, Snoogle and Microsearch make them vulnerable to congestion when many global queries occur, leading to low scalability. MASCAL [5] is an operational hospital asset management system. It utilizes active Wi-Fi RFID tags to enable the real-time tracking of patients and assets inside a hospital. Other research[6], [7], [8] focuses on the localization of sensors based on coordination schemes, while others [9], [10] rely on radio frequency or ultrasound to support sensor location.

Social network based routing. Social networks recently have been utilized for routing in Delay Tolerant Networks (DTNs) [11], [12], [13] and Mobile Ad hoc Networks (MANETs) [14], [15], [16]. The schemes exploit the history of contacts with the assumption that the meeting probability between nodes remains approximately the same, and cluster nodes with high meeting frequency choose the node which has a high probability of meeting the destination at the next hop. However, these methods only consider node movement patterns or routines, i.e., the places people regularly visit or friends they usually meet, but neglect exceptions in people movement. SCPS addresses this deficiency by capturing the exceptions in routine human mobility in location prediction.

III. The Design of the SCPS System

SCPS builds the nodes into a hierarchical structure, as shown in Figure 1, to efficiently manage object data for the search service. In the upper layer of the structure, the base stations (BSs) form a Chord DHT. In the lower layer, mobile nodes (MNs) and sensors communicate with their closest BSs.

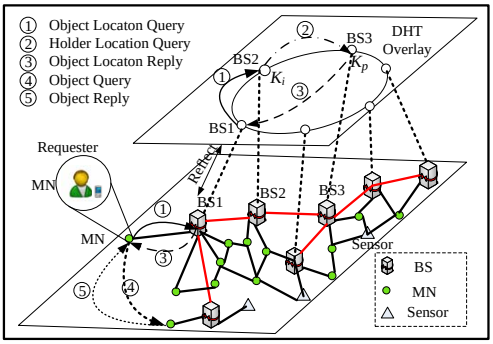


Fig. 1. SCPS structure and query.

A. Data Collection in the Distributed Hierarchical Structure

1) Object data collection and storage: SCPS uses a DHT-based overlay for data collection and storage. In a DHT, each object (or file) has a key, which is the consistent hash function of the object name. A DHT has a function *Insert(key, value)* to store the value to a node. We call the node the value's owner. Each node has an ID which is the consistent hash value of its IP address (note this implies that nodes must use public IP addresses or must all use the same private IP address space). An object is stored in a node whose ID is the closest (or first succeeding) the object's key. Any participating node can efficiently retrieve the value associated with a given key through function *Lookup(key)*.

The *holder indexer* of an object is the owner of the object's key, and the *locator* of a MN is the owner of the node's ID. The object data is either input by users manually using MNs or reported by the sensors when they detect the object. A node sends an index report that contains the user name and its objects to its nearest BS. For example, user A sends its index report "*User A has object 1, 2 and 3*" to BS1. After receiving an index report, a BS executes *Insert(key, objHolder)* for each object, where key is the hash value of the object's name. Finally, this message arrives at the holder indexer of the object, which stores the holder information of the object such as "*obj1: A, B and C*".

B. Social-aware Bayesian Network Prediction Model

A MN locator builds a social-aware BN prediction model for locating the MN. To enable each locator to collect the required data for building the BN, each MN reports its movement routine to its locator at the initial stage, SCPS stores the social relationship information of each MN to its locator initially, and the social events data of a MN is promptly sent to its locator. In the DHT-based overlay, *Insert(K,D)* is used to send data to a MN's locator, where K is the ID of the MN and D is the data. Environmental event data is sent to the locators of MNs that are influenced by the events.

A BN is a probabilistic graphical model that represents a set of random variables and their conditional dependencies, which can be used to model complex event driven casual relationships. We use BN to model and infer the probability of an event according to other observed events. Figure 2 shows an example of a BN for one person. It has variables including time, weather, health, nearby friends, and location. The location has variables such as football field, gym, home, and basketball court. The events determine location values. Football field is the value of the routine event when none of the exceptional events happen. Exceptional events such as rain, illness and meeting a friend break the person's routine. For example, at time 5:00pm, if it is raining, the probability that the person is in the gym is 1, and the probability that the person is at the football field is 0. If the person is ill, then it is more likely that he will be at home.

To provide a fine-grained view of node movement for more accurate location prediction of people, we consider the routine of a person as a chain through time. Specifically, we find the place each person usually stays during a certain time period. For example, Tom stays in his research lab during 8:00am-12:00pm and stays in the cafeteria during 12:00pm-1:00pm. We further rely on the event-driven BN to capture the influence of exceptional events. For example, the BN correctly predict that Tom is in the gym instead of the basketball court

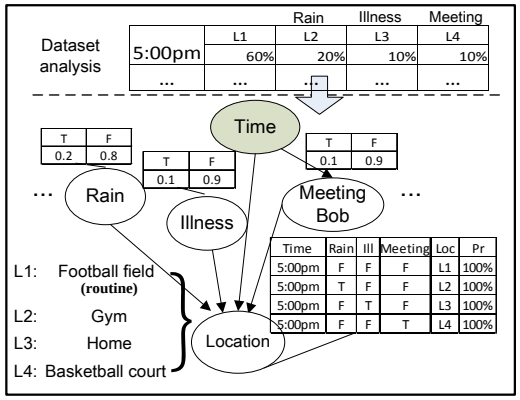


Fig. 2. A Bayesian network for one person.

by considering the rain event by adjusting the probability of variables in the BN. We further consider the event of meeting a friend and social network information in the process of BN prediction. The social network information includes the social relationship or common interests between friends. The intuition behind this scheme is the fact that people with certain social relationships tend to meet at certain places determined by their relationship and common interests in our everyday life. If friends in a basketball club are meeting each other, they are likely to play basketball or watch a basketball game at the basketball court. After training, the BN can predict the location of people when they meet each other according to their social relation information. In Section IV-A, details for building a BN using a real trace of node mobility are presented.

C. Cyber-Physical Searching

Figure 1 shows the process of object querying in SCPS. When one requester wants to search for an object named obj1, it sends an *object location query (OLQ)* containing the name of the object to its nearest BS, say BS1. If the requester is not in range of any BS, it uses geographic routing to send the query to BS1. After receiving the query, BS1 forwards the *OLQ* to the holder indexer of obj1 using *Lookup(*$key(obj1)$*,OLQ)*. Using the DHT routing algorithm, the request arrives at the BS, say BS2, which has the inverted index of obj1. Assume that the holders of obj1 are mobile users A, C and D. Then, BS2 hashes the holder names and sends out the *holder location queries (HLQ)* using the functions of *Lookup(*K_A*,(HLQ,B1))*, *Lookup(*K_C*,(HLQ,B1))* and *Lookup(*K_D*,(HLQ,B1))*. The three requests arrive at the locators of nodes A, C and D, which predict the locations of A, C and D, respectively. Assume BS3 is the locator of user A. Using the prediction model, BS3 predicts the location of A and responds to BS1 with *object location reply*. BS1 then replies to the requester with the locations. After receiving the locations, the requester sends *object query* to each object holder using geographic routing, and each holder responds to the requester with *object reply*.

IV. Performance Evaluation

We conducted real-trace driven simulations to evaluate the prediction accuracy and system effectiveness of the proposed SCPS system based on the Reality [17] dataset.

A. Bayesian Network Construction Using the Real Trace

Reality [17] is a dataset collected at the MIT Reality Group, which contains (1) survey data of 94 users in 10 months, (2) cell phone trace, and (3) Bluetooth trace. We derived the friend social relationship from the survey data. To make the Reality dataset suitable to our design and test scenarios, we processed the dataset information.

We need location records with a certain granularity, say for every minute, in order to simulate node movement. However, many missing records for days with "no signal" occur in the dataset. We went through all records and identified 19 such users with a relatively long period of records of 11 days. We used the records of the first 10 days for training of the prediction model, and those of the last day for prediction simulation. To generate events for meeting friends, we found the location of the longest accumulated meeting time of two nodes during the 11 days, and changed all meeting locations for them to that location.

We now describe how the BN of each person, (in Figure 2), is generated. We identify 3 variables: time, event and location. The time variable has 7 different values (14 hours from 8am to 10pm are divided into 7 intervals with a 2-hour granularity). The events have two values: true and false. The location has several values. The parents of *location* are *time* and *event* because the human mobility is influenced by both factors. Accordingly, we construct the BN using all retrieved events, locations and time variables from the records in the real trace. In the BN, there are links from: 1) time to the location (routine), and 2) events to the locations (exceptional events).

We calculate the historical frequency of a user visiting different locations at different time intervals in the dataset. For example, if there are 100 records at 5:00pm, in which 60 records are for Loc1, 20 records for Loc2, 10 records for Loc3, and 10 for Loc3, respectively. We infer that at 5:00pm, the probability of the person staying in Loc1 is 60%, in Loc2 is 20%, in Loc3 is 10% and in Loc4 is 10%. Since the real trace does not provide data for non-routine events, we assume that a person visits an infrequently visited place due to an exceptional event. When an exceptional event happens, the probability of the person staying at the corresponding location is 100%. When no exceptional events happens, the probability that the person staying at the routine location is 100%.

B. Experiment Settings

We conducted experiments on *NS-2* [18]. We used PlanetSim [19] to test the overhead and delay in the DHT-based overlay formed by BSs, and used the Bayesian Network Tools [20] for BN inference to test the BN prediction delay. GPSR [21] is used when a requester sends a query to a nearby BS not within its transmission range and when a requester sends an object query to the object holder. If the location that the requester received is incorrect and the packet arriving at the location cannot find the object holder, GPSR uses perimeter mode routing.

We compared our SCPS system with SCPS without social-aware prediction (SCPSw/oS), SCPS without BN prediction

(SCPSw/oP), Snoogle [3], statistics-based method (Statistic) [14], and MOPS [12]. SCPSw/oS is SCPS without considering social events in BN prediction. SCPSw/oP is SCPS without using the location prediction mechanism. Instead, it uses the periodic reporting from nodes to BSs to keep track of the locations of every user. Snoogle [3] is a centralized search engine with a two-layer hierarchical structure over the sensor nodes. The higher layer is a central server called KeyIP and the sensors in the lower layer are called IPs. Queries for arbitrary objects are firstly directed to the KeyIP, which returns the best matched IPs, which send back the IDs of nodes containing the searched objects by checking their inverted indexes. In the simulation, we used BSs to function as the IPs, and used the BS in the center of the simulation area as the keyIP. To be comparable to SCPS, Snoogle also adopts geographic routing, and relies on periodic location reporting to IPs for destination locations. In Snoogle and SCPSw/oP, mobile nodes report their locations to BSs every 5 seconds.

The Statistic method is similar to SCPS, but only uses a person's routine for location prediction. MOPS is a publish/subscribe system based on social network information. It clusters nodes with frequent communications into a community. It uses nodes having frequent contact with other communities as brokers for inter-community communication and direct contact between nodes in the same community for intra-community communication. In MOPS, a node carries the message until meeting the destination.

Table I summarizes the default parameters used in the simulation unless otherwise specified. Of the 69 total nodes, 19 nodes move according to the real trace and 50 nodes move randomly with a speed randomly chosen from [0,2]m/s. All results over 2000 queries are averaged for the final results.

We randomly assigned 95 items (5 copies of 19 items) to 19 people. Each node starts to query after a 20s initialization time period at a certain query rate for 580s. The simulation then ran for another 20s before stopping. A query rate of $1/x$ means a query is sent out every x seconds. The item queried is randomly chosen from the 19 items. A requester randomly selects 1 from the 5 object holders in the object location reply for an object query. The length of a query packet is 28 bytes. We consider the following metrics:

TABLE I
SIMULATION PARAMETERS

Environment Parameters	**Default Value**
Simulation area	$600m \times 600m$
Node Parameters	**Default Value**
Total number of mobile nodes	69
Total number of base stations	7
Physical layer	IEEE 802.11
Communication range	200m
Movement speed	$0m/s - 2m/s$
The length of a query	28 bytes
Query Parameters	**Default Value**
Query rate	$1/10$ (one query every 10 seconds)
Initialization period	20s
Query time	580s

1) Object hit rate: the average ratio between the number of *object replies* and the number of queries sent from requesters to object holders using geographic routing. This metric can reflect the accuracy of location prediction since a wrong location in geographical routing leads to routing failure.

2) Overhead: the total number of hops passed by all messages in object searching. The messages do not include hello messages.

3) Query delay: the average delay time of all successfully resolved object queries. The query delay time is the time elapsed from the time a requester sends an *object location query* to the time the requester receives the *object reply*.

C. Performance with Different Query Rates

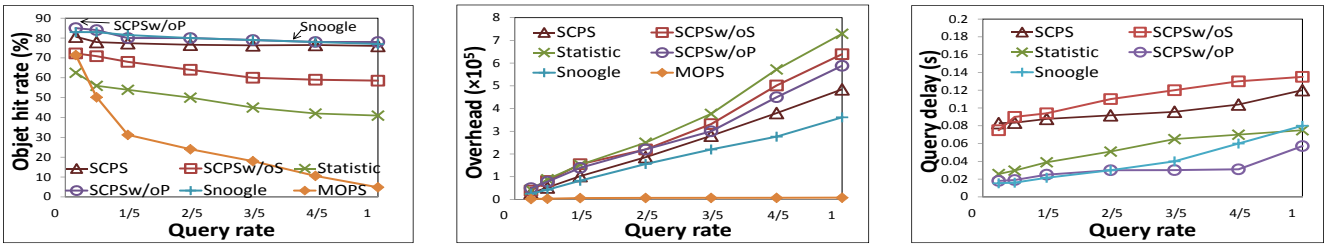
In this test, we examined the performance of SCPS and other methods at different query rates. We varied the query rate of nodes over 7 different rates from 1/20 to 1.

1) **Object hit rate:** Figure 3(a) shows the average object hit rate of the six methods with different query rates. First, we find that the object hit rate of SCPS is higher than the other two prediction based methods (SCPSw/oS and Statistic). This is because it can produce more accurate locations than the other two methods due to its higher prediction accuracy, so that a query can be sent to the correct destination with higher probability. The higher object hit rate of SCPS compared with SCPSw/oS verifies the effectiveness of considering social events in prediction. Similarly, the object hit rate of SCPSw/oS is higher than Statistic, which verifies the importance of considering exceptional environmental events in prediction.

Relying on periodic location reporting, SCPSw/oP and Snoogle can always provide accurate object locations, thus producing the highest object hit rate. It is interesting to see that the object hit rate of SCPS is comparable to SCPSw/oP and Snoogle, which can always provide correct locations due to periodic reporting. This is caused by two reasons. First, SCPS provides relatively highly accurate object locations. Second, SCPSw/oP and Snoogle send many location reporting messages in periodic reporting, which greatly increases the channel contention, thereby lowering their hit rates.

In addition, we see that MOPS exhibits a sharp decrease in the object hit rate as the query rate increases. When the brokers of different communities meet, they only can exchange a limited number of messages due to limited meeting time. Thus, a higher query rate produce more undelivered messages, which are dropped when the test completes, leading to a lower object hit rate. These experimental results confirm that SCPS has a very high object hit rate even at high query rates, which is comparable to reporting based methods.

2) **Overhead:** Figure 3(b) plots the total overhead of the six methods. We find that the overhead of SCPSw/oP is higher than SCPS. This is because nodes in SCPSw/oP need to periodically report their locations to nearby BSs, while SCPS does not have this requirement due to its location prediction capacity. It is interesting to see that, though also using location reporting, Snoogle has lower overhead than SCPSw/oP and SCPS. This is due to two reasons. 1) Snoogle has a high location query drop rate, which reduces the number of hops traversed by messages in multi-hop transmission. 2) Compared



(a) Object hit rate. (b) Overhead. (c) Query delay.

Fig. 3. Performance of object search with different query rates.

with Snoogle, SCPSw/oP and SCPS need extra overhead in the DHT layer for looking up the locations of object owners.

We observe that the overhead of MOPS is much lower than the other methods. This is because the packets in MOPS are mainly transmitted by brokers carrying the messages instead of using hop-by-hop transmission. We also see that the overhead follows: Statistic>SCPSw/oS>SCPS. This is because the location accuracy of the methods follows Statistic<SCPSw/oS<SCPS. Packets with wrong locations result in many "hanging packets", and thereby increasing overhead. These observations verify that SCPS has a relatively lower overhead compared with methods with similar location query drop rate and object hit rate.

3) **Query delay:** Figure 3(c) illustrates the average query delay of the six methods with different query rates. We find that the average delay of MOPS is much higher than the other methods (about 50s). Thus, we do not include MOPS in the figure. The reason for the high delay is because the inter-community communication is conducted only when brokers meet each other, which results in a long delay.

We observe that Snoogle, SCPSw/oP and Statistic generate lower query delay than SCPS and SCPSw/oS because the first three methods do not use BN location prediction, which takes about 0.0675s per prediction. We also see that Statistic produces a slightly higher query delay than Snoogle and SCPSw/oP. Some packets in Statistic have much larger delay due to the incorrect location used in routing. Though most of the wrong location packets are dropped, a few can arrive at the destination after a long network detour, leading to larger average query delay. Without considering social events, SCPSw/oS has lower location prediction accuracy than SCPS, thus it generates higher average query delay than SCPS.

As the query rate increases, the query delay of Snoogle increases faster than SCPSw/oP, because centralized KeyIP has a long queue when the query rate is high. We observe the query delays of all methods grow as the query rate increases due to network congestion generated by more queries.

V. Conclusion

In this paper, we proposed a Social-aware distributed Cyber-Physical human-centric Search engine (SCPS) that provides a scalable, efficient and accurate search service for physical objects carried by moving people. SCPS consists of three components: *distributed hierarchical structure*, *social-aware Bayesian network prediction model* and *cyber-physical searching algorithm*. The three components cooperate to predict the locations of queried objects without the need of periodical location reporting. Trace-driven simulation results show the high efficiency and location prediction accuracy of SCPS in comparison with existing methods.

Acknowledgements

This research was supported in part by U.S. NSF grants OCI-1064230, CNS-1049947, CNS-1025652, CNS-1025649, CNS-1057530 and CNS-0917056, Microsoft Research Faculty Fellowship 8300751, and Sandia National Laboratories grant 10002282.

References

[1] M. C. Gonzalez, C. A. Hidalgo, and A. Barabasi. Understanding individual human mobility patterns. *Nature*, 453:779–782, 2008.
[2] K. K. Yap, V. Srinivasan, and M. Motani. MAX: human-centric search of the physical world. In *Proc. of SenSys*, page 179, 2005.
[3] H. Wang, C. C. Tan, and Q. Li. Snoogle: A search engine for the physical world. In *Proc. of IEEE INFOCOM*, 2008.
[4] C. Tan, B. Sheng, H. Wang, and Q. Li. Microsearch: When search engines meet small devices. *Pervasive Computing*, pages 93–110, 2008.
[5] E. A. Fry and L. A. Lenert. MASCAL: RFID tracking of patients, staff and equipment to enhance hospital response to mass casualty events. In *Proc. of AMIA Annual Symposium*, 2005.
[6] G. Mao, B. Fidan, and B. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, 2007.
[7] R. Peng and M. L. Sichitiu. Probabilistic localization for outdoor wireless sensor networks. *Mobile Comp. and Comm. Review*, 2007.
[8] H. C. Chu and R. H. Jan. A GPS-less, outdoor, self-positioning method for wireless sensor networks. *Ad Hoc Networks*, 5(5):547–557, 2007.
[9] J. Bruck, J. Gao, and A. Jiang. Localization and routing in sensor networks by local angle information. *ACM Transactions on Sensor Network*, 5:1–31, 2009.
[10] Y. M. Kwon, K. Mechitov, S. Sundresh, W. Kim, and G. Agha. Resilient localization for sensor networks in outdoor environments. *TOSN*, 7(1):1–30, 2010.
[11] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proc. of MobiHoc*, 2007.
[12] F. Li and J. Wu. MOPS: Providing Content-based Service in Disruption-tolerant Networks. In *Proc. of ICDCS*, 2009.
[13] A. Mtibaa, M. May, C. Diot, and M. Ammar. Peoplerank: social opportunistic forwarding. In *Proc. of INFOCOM*, 2010.
[14] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *Service Assurance with Partial and Intermittent Resources*, 2004.
[15] J. Ghosh, S. J. Philip, and C. Qiao. Sociological orbit aware location approximation and routing (SOLAR) in MANET. *Ad Hoc Networks*, 5(2):189–209, 2007.
[16] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proc. of MobiHoc*, 2008.
[17] N. Eagle, A. Pentland, and D. Lazer. Inferring social network structure using mobile phone data. In *Proc. of PNAS*, 2007.
[18] The Network Simulator ns-2. http://www.isi.edu/nsnam/ns/.
[19] PlanetSim. http://projects-deim.urv.cat/trac/planetsim/.
[20] BNT, Bayes Net Toolbox for Matlab. http://code.google.com/p/bnt/.
[21] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proc. of MobiCom*, pages 243–254, 2000.