

# Friendship Based Routing in Delay Tolerant Mobile Social Networks

Eyuphan Bulut and Boleslaw K. Szymanski

Department of Computer Science and Center for Pervasive Computing and Networking  
Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180, USA  
{bulute, szymansk}@cs.rpi.edu

**Abstract**—Routing in delay tolerant networks (DTN) have attracted a great interest recently. Increasingly popular type of DTNs are mobile social networks (MSN) also called pocket switched networks. Hence, analyzing accurately social network properties has become an important issue in designing efficient routing protocols for MSNs. In this paper, we first introduce a new metric for detecting the quality of friendships accurately. Using the introduced metric, each node defines its friendship community as the set of nodes having close friendship with itself either directly or indirectly. Then, we present *Friendship Based Routing* in which temporally differentiated friendships are used to make the forwarding decisions of messages. Real trace-driven simulation results show that the introduced algorithm achieves better delivery rate while forwarding fewer messages than the existing algorithms.

## I. INTRODUCTION

Delay Tolerant Networks (DTN) [1][2], are sparse mobile wireless networks in which the connection between nodes changes over time and it is usually not possible to find a path from source to destination at any time instance. Among many real life examples of such networks, mobile social networks (MSN) (called also pocket switched networks) are of growing significance as a result of the rapid and wide spread usage of different kinds of devices (e.g., cell phones, GPS devices) with wireless capabilities among people and their surroundings.

Since mobile social networks has the potential of collaborative data gathering via already deployed and human maintained devices, opportunistic routing of messages in these networks has attracted a great interest recently. However, due to the intermittent connectivity and lack of continuous end-to-end path between the nodes, routing is a challenging problem in these networks. To ease these difficulties and enable nodes to give better forwarding decisions through routing, inherent social network properties of these networks have been utilized. The connectivity (opportunity for message transfers) between human-carried devices is achieved when they get into the range of each other. Thus, the relationship defining the frequency and

duration of the connectivity between nodes has to be analyzed to route messages efficiently. For example, consider a high school network. Students in the same class have higher chance to see (so also to transfer data to) each other than the students from other classes that can meet only during breaks.

In this paper, utilizing the social network features of an MSN, we present *Friendship Based Routing*. To analyze social relations between nodes (i.e. people), we need to define their *friendships* in terms of their behavior. For this purpose, we define a new metric measuring different aspects of friendship behavior recorded in the history of their encounters with other nodes. We consider both direct and indirect friendship. We also differentiate friendships according to time of day and propose to use different friendship communities in different time periods.

The rest of the paper is organized as follows. In Section II we present a brief overview of previous work. In Section III we give the detailed design of proposed algorithm. In Section IV, we talk about our simulation model and its results. In Section V, we discuss the presented algorithm and outline the future work. Finally, we offer conclusion in Section VI.

## II. RELATED WORK

In the previous studies, several routing algorithms have been introduced for DTNs based on different techniques (multi-copy based [3]-[5], single-copy based [6]-[8], erasure coding based [9]-[11]). Besides these studies, most of which assume simplistic random mobility models (such as random walk) for the movement of nodes, many recent studies have focused on DTNs consisting of human-carried devices (we call them MSN) and tried to analyze the social network properties of these networks to aid the design of efficient routing algorithms.

In [12], Daly et al. use both the betweenness and the similarity metric to increase the performance of routing. In each contact of two nodes, the utility function containing these two metrics is calculated for each destination, then the node having higher utility value for a destination is given the messages. In [13], each node is assumed to have two rankings: global and local. While the former denotes the popularity (i.e. connectivity) of the node in the entire society, the latter denotes its popularity within its own community. Messages are forwarded to nodes having higher global ranking until a node in the destination's community is found. Then, the messages are forwarded to nodes having higher local ranking

This research was sponsored by US Army Research laboratory and the UK Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001 and under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

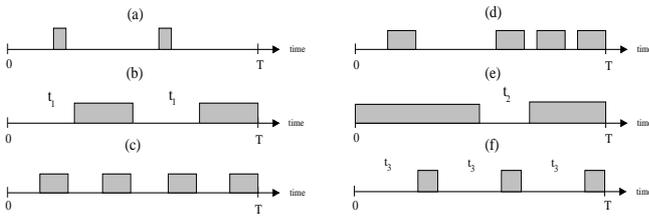


Fig. 1. Six different encounter history between nodes  $i$  and  $j$  in the time interval  $[0, T]$ . Shaded boxes show the encounter durations during which the nodes are within their communication ranges.

within destination's community. A distinction between local community members and others is also made in [14] and the forwarding is optimally balanced between these two kinds of encountered nodes. In [15], a community-based epidemic forwarding scheme is presented. First, it efficiently detects the community structure using local information of nodes. Then, it forwards the message to each community through gateways.

In some other studies several interesting properties of social networks are considered. In [16], irregular deviations from the habitual activities of nodes are addressed and it is shown that worst-case performance of routing can be improved by scattering multiple copies of a message in the network such that even deviant (less frequently encountered) nodes will be close to at least one of these copies. In [17], the effect of socially selfish behavior of nodes on routing is studied.

In this paper, we introduce a new routing algorithm different from all above studies. First, we define a new metric to understand relations between nodes more accurately. Second, we propose a local community formation based on this new friendship detection metric. We use not only direct relations but also indirect ones in a different way than it was considered previously. Third, we introduce a new approach to handle temporal differentiations of node relations. Throughout the presentation of all these features of our design, we show in detail how they differ from the previous work.

### III. THE PROPOSED SCHEME

#### A. Analysis of Node Relations

Since the nodes in an MSN encounter intermittently, the link quality between each pair of nodes needs to be estimated to learn about the possible forwarding opportunity between nodes. Then, the temporal encounter information between nodes can be condensed to a single link weight and the neighboring graphs of nodes can be constructed.

Previously, several metrics, including encounter frequency, total or average contact period and average separation period [15] were used to extract the quality of links between pairs of nodes. However, all these metrics have some deficiencies in the accurate representation of forwarding opportunity between nodes. For example, consider the six different encounter histories of two nodes,  $i$  and  $j$  in Figure 1. Shaded boxes show the encounter durations between these nodes in the time

interval  $T$ . In cases  $a$  and  $b$ , the encounter frequencies are the same but the contact between the nodes lasts longer in case  $b$  than in case  $a$ . Therefore encounter pattern  $b$  offers better forwarding opportunities than  $a$  does. Comparing cases  $b$  and  $c$ , we notice that the contact durations are the same but the encounter frequencies are different. Since frequent encounters enable nodes to exchange messages more frequently, case  $c$  is preferable to  $b$  for opportunistic forwarding. Among the previously proposed metrics, encounter frequency fails (to represent the stronger link) in the comparison of cases  $a$  and  $b$ , and total contact duration fails in the comparison of cases  $b$  and  $c$ . Although average separation period can assign correct link weights representing the forwarding opportunity in cases  $a$ ,  $b$  and  $c$ , it fails in other cases. When we compare cases  $c$  and  $d$ , both the contact durations and encounter frequencies are the same. However, case  $c$  is preferred to  $d$  due to the even distribution of contacts. In [15], preference of case  $c$  is achieved by utilizing irregularities in separation period as penalty factor. However, deciding on how much it will affect the link quality in different cases is still difficult. Moreover, for the cases such as  $e$  and  $f$ , average separation period fails to assign accurate link weights. If  $t_1 = t_2$ , average separation period can not differentiate cases  $b$  and  $e$  but case  $e$  is preferable due to its longer contact duration (average separation period can even give preference to case  $b$  if  $t_1$  is slightly less than  $t_2$ ). Similarly, if  $t_1 = t_3$ , average separation period can not differentiate cases  $b$  and  $f$  even though case  $b$  offers better forwarding opportunity.

To find a better link metric that reflects the node relations more accurately, we have considered the following three behavioral features of close friendship: high frequency, longevity, regularity. That is, to be considered friends of each other, two nodes need to contact frequently, their contacts must be long-lasting and regular. Note that frequency and regularity are different. Two nodes may meet infrequently but regularly (for example, once a week) and still be considered friends. This is of course a weaker friendship than the one with contacts both frequent and regular. The previous metrics take into account some of these features but not all of them at the same time. To account these properties in one metric, we introduce a new metric called *social pressure metric* (SPM) that may be interpreted as a measure of a social pressure that motivates friends to meet to share their experiences. In our setting, this amounts to answering the question 'what would be the average message forwarding delay to node  $j$  if node  $i$  has a new message destined to node  $j$  at each time unit?'. Then, we define the link quality ( $w_{i,j}$ ) between each pair as the inverse of this computed value. More formally:

$$SPM_{i,j} = \frac{\int_{t=0}^T f(t)dt}{T} \text{ and } w_{i,j} = \frac{1}{SPM_{i,j}}$$

where  $f(t)$  returns the remaining time to the first encounter of these nodes after time  $t$  (if they are currently in contact  $f(t) = 0$ ). The bigger the value of  $w_{i,j}$ , the closer the friendship (the higher the forwarding opportunities) between the nodes  $i$  and

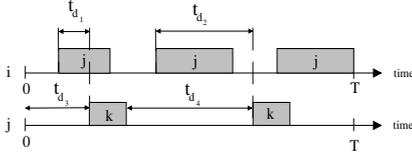


Fig. 2. Encounter history between node  $i$  and  $j$  (upper) and between node  $j$  and  $k$  (lower) in the same time interval  $[0, T]$ .

$j$ . Note that, when we evaluate all cases in Figure 1,  $w_{i,j}$  gives preference to cases which offer more forwarding opportunities.

### B. Friendship Community Formation

Each node can compute its link qualities ( $w_{i,j}$ ) with other nodes from its contact history. Then, it can define its friendship community as a set of nodes having a link quality larger than a threshold ( $\tau$ ) with itself. But this set will include only direct friends. However, two nodes that are not close friends directly (they even may not have contacts at all) still can be close indirect friends. This happens if they have a very close friend in common so that they can contact frequently through this common friend. To find such indirect friendships between nodes, we propose to use *conditional SPM* (or simply *CSPM*) between nodes. Consider the sample encounter history in Figure 2. While the upper one shows the contacts between nodes  $i$  and  $j$ , the lower one shows the contacts between nodes  $j$  and  $k$ . We define  $CSPM_{j,k|i}$  as the average time it takes node  $j$  to give node  $k$  the message received from node  $i$ . That is, for the contact history in Figure 2, node  $j$  computes  $CSPM_{j,k|i} = (\int_{s=0}^{t_{d_1}} s + \int_{s=0}^{t_{d_2}} s) / T$  instead of  $SPM_{j,k} = (\int_{s=0}^{t_{d_3}} s + \int_{s=0}^{t_{d_4}} s) / T$ .

Each node can detect its direct friendships from its own history. To detect indirect friendships, a node needs *CSPM* values of its friends for its non-contact nodes. Once such *CSPM* values are received periodically, each node can form its community using the following definition:

$$F_i = \{j \mid w_{i,j} > \tau \text{ and } i \neq j\} \cup \{k \mid w_{i,j,k} > \tau \text{ and } w_{i,j} > \tau \text{ and } i \neq j \neq k\}$$

where, to simplify the data collection and computation, we approximate<sup>1</sup> the indirect link weight as  $w_{i,j,k} = 1 / (SPM_{i,j} + CSPM_{j,k|i})$ . The above equation enables nodes to detect their one-hop direct and two-hop indirect friends. Indirect friendship can easily be generalized to friends more than two hops away. We have not included such extension because [15] demonstrated that nodes in the same community are usually at most two hops away from each other.

Note that the introduced method for detecting the indirect strong links between nodes is different than previous approaches (based on transitivity) which basically consider the

<sup>1</sup>For lack of space, a more complex but precise formula based on computing the delay of messages accumulated during the indirect transfer is omitted here; it is included in the extended version of the paper.

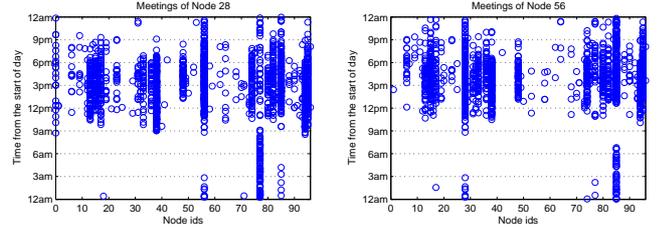


Fig. 3. Encounter distributions of node 28 (left) and 56 (right) in MIT traces.

links between node pairs separately and assume a virtual link between node  $i$  and  $k$  if  $w_{i,j}w_{j,k} > \tau$ . However, in our model we can detect indirect relations more accurately. For example, if node  $j$  has a weak link with node  $k$ ,  $w_{i,j}w_{j,k}$  may be less than  $\tau$ . However, if node  $j$  usually meets node  $k$  in a short time right after its meeting with node  $i$ , our metric can still consider node  $k$  as a friend of node  $i$ .

When we analyzed a long-period and commonly used social network data (MIT Reality dataset [21], see Section IV for details), we have also noticed temporal but regular [18] differentiations in node relations. Consider the Figure 3 illustrating the distribution of contact times of two different nodes (28 and 56)<sup>2</sup> with other nodes (with ids [0-96]) in MIT traces. Clearly, nodes encounter other nodes in some specific periods of the day. For example, node 28 meets with node 38 usually between 9am to 7pm while it meets with node 48 usually between 1pm to 7pm.

It is reasonable to expect similar behavior in other mobile social networks. For example,  $i$  can be a school, work or home friend of a node  $j$  and the encounter times can differ accordingly. Moreover,  $i$  can be both school and home friend of  $j$  so that they stay together during the day. Previously some aging mechanisms [6] [20] were used to reflect this feature of node relations. However, the most significant drawback of such models is their slow reaction for periodically changing quality of node's links. For example, around 7pm, the link quality of node 56 with node 38 (see Figure 3) starts to decrease with aging effect<sup>3</sup> and still keeps a high value for some time, however node 56 usually does not meet with node 38 until 10am next day. Therefore, forwarding a message considering an aged but still strong link quality may cause high delays when the link is already in its periodic low.

To reflect this differentiation in the strength of friendships properly, we propose to use temporal friendship communities. That is, each node  $i$  will compute its  $F_i$  for different periods of the day and will have different friendship communities in different periods. For example, if we divide a day into periods of three hour duration, according to Figure 3 node 85 can be the only friend of node 56 in period 3am-6am, whereas nodes 28, 85 and 95 can be friends of node 56 in period 9pm-12am.

<sup>2</sup>We selected these nodes since they are the top two nodes having the most encounters with other nodes.

<sup>3</sup> $w_{i,j} = w_{i,j} \alpha^t$  where  $t$  is the time since the last encounter and  $0 < \alpha < 1$  is aging parameter.

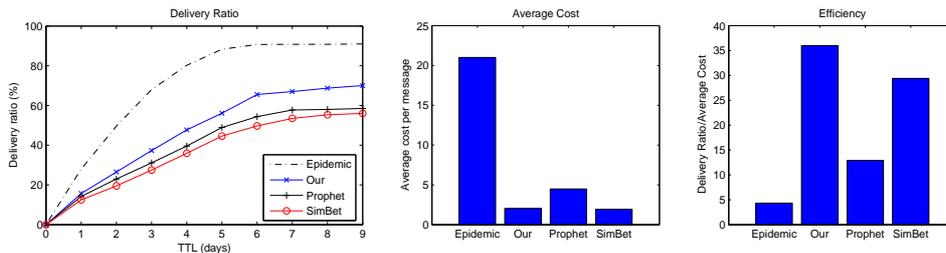


Fig. 4. Simulation Results: Delivery Ratio (Left), Average Cost (Middle), Efficiency (Right)

### C. Forwarding Strategy

Once each node constructs its friendship community for each period, the forwarding algorithm works as follows. If a node  $i$  having a message destined to  $d$  meets with node  $j$ , it forwards the message to  $j$  if and only if node  $j$ 's current friendship community (in the current period) includes node  $d$  and node  $j$  is a stronger friend of node  $d$  than node  $i$  is. It should be noted that even if node  $j$  has a better link with node  $d$  than node  $i$ 's has, if node  $j$  does not include  $d$  in its current friendship community, node  $i$  will not forward the message to node  $j$ .

We also need to handle period boundary cases which arise when the encounter of two nodes is close to the end of the current period. In such a case, nodes use their friendship communities in the next period. For example, if we use three hour periods for community formation and node  $i$  meets node  $j$  at 2:45pm, it would be better if the nodes use their communities in the next three hour period (3pm – 6pm) to check whether the destination is included. Since the time remaining in the current period is very limited, using the current communities may lead to make wrong forwarding decisions. In our algorithm, we use threshold  $t_b$  and let the nodes use next period's community information if remaining time to the end of current period is less than  $t_b$ .

## IV. EVALUATIONS

### A. Simulation Setup

To evaluate our algorithm, we used real trace-driven simulations based on MIT Reality data [21]. This data consists of the traces of 97 Nokia 6600 smart phones which were carried by students and staff at MIT over nine months. The phones run Bluetooth device discovery every five minutes and log contacts (i.e. start and end time) with other nodes in their proximity. In our simulations, we used the contacts logged during a three month period from the beginning of February to the end of April. This is the time of the second academic semester, so human relationships are relatively stable and the participants are active on the campus [18].

In the simulations, we generate 5000 messages, each from a random source node to a random destination node<sup>4</sup> at every

<sup>4</sup>Since some nodes do not have any contacts with others in the selected three month period, we did not assign those nodes as either source or destination to prevent meaningless messages.

two minutes. All messages are assigned a TTL value representing the delay requirement. To form friendship communities, we used three hour periods and set  $\tau = 1/80 \text{ min}^{-1}$  and  $t_b = 15 \text{ min}$ . We repeated each simulation 10 times with different random seeds and took the average of each run as result.

### B. Algorithms in Comparison and Performance Metrics

In simulations, we compare our algorithm with two other benchmark algorithms, PROPHET [6] and SimBet [12]. In the former, each node calculates its delivery predictability using its contact history along with transitivity and aging features and relays a packet to a node with higher delivery predictability. In the latter, each node calculates a simbet metric using two social measures (similarity and betweenness) and the messages are forwarded to nodes with higher simbet metric. For both algorithms we use the same parameters presented in [6] and [12]. To show the highest delivery ratio that could be achieved with current setting in the network, we also present the results of epidemic routing [3].

For the comparison of our algorithm to these existing algorithms, we use the following three metrics: message delivery ratio, average cost, and routing efficiency. Delivery ratio is the proportion of messages that are delivered to their destinations among the total messages generated. Average cost is the average number of forwards done per message during the simulation. Finally, routing efficiency [19] is defined as the ratio of delivery ratio to the average cost.

### C. Simulation Results

In Figure 4, we show comparison of all algorithms in terms of the three aforementioned metrics. As it is seen in the left graph, while our algorithm achieves 72% of delivery ratio, Prophet and SimBet could only deliver 60% and 58% of all messages, respectively. Moreover, as the middle graph shows, our algorithm achieves this higher delivery ratio with almost the same average cost as SimBet, and with less than half of Prophet's average cost. As a result, the routing efficiency achieved by our algorithm is 20% higher than the efficiency of SimBet and 170% higher than the efficiency of Prophet. Note that epidemic routing achieves the optimal delivery but it achieves the highest cost and the lowest efficiency.

## V. DISCUSSIONS AND FUTURE WORK

### A. Complexity of Introduced Algorithm

Since in the introduced algorithm each node determines its friendship community in each period using mainly its own history, there is no much control message or system maintenance transfers occurring between nodes. The only information (to decide its indirect close friends) that a node needs from its contacts is their CSPM values with its non-contact nodes. However, this information is requested from only close friends of nodes and performed with small size messages compared to data messages. On the other hand, the control message overhead in Prophet and SimBet is significantly higher than in our algorithm, because nodes change their summary vectors during contact times.

### B. Number of Periods vs. Performance

Obviously, increasing the number of periods that a day is divided into (thus the local friendship communities each node has) will enable the node to make better forwarding decisions. On the other hand, the cost of computing the friendship communities in each period and also the space required to hold different communities will increase as well. However, as long as this cost could be handled and there is enough space at nodes, better results could be achieved.

### C. The Effect of Thresholds

Each node forms its friendship community from nodes with strong links (i.e., with weights larger than a threshold) with itself. Clearly, as the threshold increases (decreases), friend lists of nodes get smaller (bigger) and routing performance of our algorithm changes. Similarly using different  $t_b$  values can change the routing performance. Therefore, in future work we will look at this issue and try to find optimum values of  $\tau$  and  $t_b$ .

### D. Possible Extensions

We believe that the performance of the introduced algorithm can be improved by using transitive friendship behaviors of different nodes in consecutive periods of the day. For example, when node  $i$  has a close friend  $j$  in period  $12pm - 3pm$  of the day, and node  $j$  has a close friend  $k$  in  $3pm - 6pm$  period, then when node  $s$  meets node  $i$  and has a message destined to node  $k$  in period  $12pm - 3pm$ , it can forward this message to node  $i$  (even though node  $i$  has no direct or indirect close friendship with node  $k$  in the current period). This is because with high probability the message will be forwarded from  $i$  to  $j$  and then from  $j$  to  $k$ . However, such a solution will increase the maintenance cost of the algorithm. We will study this issue in our future work and analyze the cost-benefit tradeoff.

destination node in their friendship communities. We differentiate those friendship communities depending on the period of the day in which forwarding is done. We evaluated the introduced algorithm through trace driven simulations using MIT data and demonstrated that it performs better than two benchmark algorithms proposed previously.

## REFERENCES

- [1] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, *Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebraNet*, Proc. ACM ASPLOS, 2002.
- [2] J. Ott and D. Kutscher, *A disconnection-tolerant transport for drive-thru internet environments*, Proc. IEEE INFOCOM, 2005.
- [3] A. Vahdat and D. Becker, *Epidemic routing for partially connected ad hoc networks*, Duke University, Tech. Rep. CS-200006, 2000.
- [4] T. Spyropoulos, K. Psounis, C. S. Raghavendra, *Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-copy Case*, IEEE/ACM Transactions on Networking, 2008.
- [5] E. Bulut, Z. Wang, B. Szymanski *Cost Effective Multi-Period Spraying for Routing in Delay Tolerant Networks*, to appear in IEEE/ACM Transactions on Networking, 2010.
- [6] A. Lindgren, A. Doria, and O. Schelen, *Probabilistic routing in intermittently connected networks*, SIGMOBILE Mobile Computing and Communication Review, vol. 7, no. 3, 2003.
- [7] S. Jain, K. Fall, and R. Patra, *Routing in a delay tolerant network*, Proc. Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 145-158, 2004.
- [8] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, *MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks*, Proc. IEEE Infocom, April 2006.
- [9] Y. Wang, S. Jain, M. Martonosi, and K. Fall, *Erasure-coding based routing for opportunistic networks*, Proc. ACM SIGCOMM Workshop on Delay-Tolerant Networking, 2005.
- [10] Y. Wang and H. Wu, *Delay/Fault-Tolerant Mobile Sensor Network (DFT-MSN): A New Paradigm for Pervasive Information Gathering*, IEEE Transactions on Mobile Computing, vol. 6, no. 9, pp. 1021-1034, 2007.
- [11] E. Bulut, Z. Wang, B. Szymanski, *Cost Efficient Erasure Coding based Routing in Delay Tolerant Networks*, Proc. ICC 2010.
- [12] E. Daly and M. Haahr, *Social network analysis for routing in disconnected delay-tolerant manets*, Proc. ACM MobiHoc, 2007.
- [13] P. Hui, J. Crowcroft, and E. Yoneki, *BUBBLE Rap: Social Based Forwarding in Delay Tolerant Networks*, Proc. ACM MobiHoc, 2008.
- [14] E. Bulut, Z. Wang and B.K. Szymanski, *Impact of Social Networks in Delay Tolerant Routing*, Proc. IEEE GLOBECOM, 2009.
- [15] F. Li and J. Wu, *LocalCom: a community-based epidemic forwarding scheme in disruption-tolerant networks*, Proc. IEEE Conf. Sensor, Mesh and Ad Hoc Communications and Networks, p.574-582, June 22-26, 2009.
- [16] T. Zhou, R. R. Choudhury, K. Chakrabarty, *Diverse Routing: Exploiting Social Behavior for Routing in Delay-Tolerant Networks*, Pro. Conf. Computational Science and Engineering, Canada, 2009.
- [17] Q. Li, S. Zhu and G. Cao, *Routing in Selfish Delay Tolerant Networks*, Proc. IEEE Infocom 2010.
- [18] P. Hui and J. Crowcroft, *Predictability of Human Mobility and Its Impact on Forwarding*, Communications and Networking in China, 2008.
- [19] J. M. Pujol, A. L. Toledo, and P. Rodriguez, *Fair routing in delay tolerant networks*, Proc. IEEE INFOCOM, 2009.
- [20] J. Link, N. Viol, A. Goliath and K. Wehrle, *SimBetAge: utilizing temporal changes in social networks for pocket switched networks*, Proc ACM Workshop on User-provided Networking, Rome, Italy, 2009.
- [21] N. Eagle, A. Pentland, and D. Lazer, *Inferring Social Network Structure using Mobile Phone Data*, Proc. National Academy of Sciences, 106(36), pp. 15274-15278, 2009.

## VI. CONCLUSION

In this paper, we studied the routing problem in mobile social networks. First, we introduced a new metric to detect contact relations between nodes accurately based on friendships. Then, we introduced a new routing algorithm in which each node forwards their messages to nodes that contain the