Exploiting self-reported social networks for routing in ubiquitous computing environments

Greg Bigwood, Devan Rehunathan, Martin Bateman, Tristan Henderson, Saleem Bhatti
School of Computer Science,
University of St Andrews,
Fife KY16 9SX, UK
{gjb,dr,mb,tristan,saleem}@cs.st-andrews.ac.uk

***Abstract*—**

Mobile, delay-tolerant, ad hoc and pocket-switched networks may form an important part of future ubiquitous computing environments. Understanding how to efficiently and effectively route information through such networks is an important research challenge, and much recent work has looked at detecting communities and cliques to determine forwarding paths.

Such detected communities, however, may miss important aspects. For instance, a user may have strong social ties to another user that they seldom encounter; a detected social network may omit this tie and so produce sub-optimal forwarding paths. Moreover, the delay in detecting communities may slow the bootstrapping of a new delay-tolerant network.

This paper explores the use of *self-reported* social networks for routing in mobile networks in comparison with *detected* social networks discovered through encounters. Using encounter records from a group of participants carrying sensor motes, we generate detected social networks from these records. We use these networks for routing, and compare these to the social networks which the users have self-reported on a popular social networking website.

Using techniques from social network analysis, we find that the two social networks are different. These differences, however, do not lead to a significant impact on delivery ratio, while the self-reported social network leads to a significantly lower cost.

I. Introduction

There is one mobile phone for every two people in the world [1], and in many countries, such as the UK[1], there are more mobile phones than people. New network architectures have been proposed that exploit these mobile devices. In particular, so-called delay-tolerant, opportunistic, or pocket-switched network architectures attempt to provide application support in scenarios with: high levels of node mobility that may lead to disruptions in connectivity; asymmetric or unidirectional links; links with high propagation delay; and links with high error rates.

Example scenarios and applications for such mobile delay-tolerant networks (DTNs) might include information propagation in disaster scenarios, network connectivity for rural areas, sensor data gathering, or simply the use of existing mobile applications without incurring additional charges from an infrastructure mobile network provider.

[1]http://www.ofcom.org.uk/research/cm/tables/q2_2007/q2_2007.pdf

A fundamental issue in these DTNs is how to effectively and efficiently route information. If network nodes (users) are mobile, then static routing tables are inappropriate, and we require some mechanism for finding the best node to which to forward a message in order for the message to effectively reach its destination. Since the destination of many messages will be to a node known to the source node, and many of the intermediate nodes may also be nodes known to the source (e.g., co-located nodes in a home or workplace), many researchers have explored the use of *social network* information for building DTN routing tables. By examining the social network of the nodes visited by a particular node, it may be possible to optimise routing by forwarding messages to nodes that are encountered more often.

To build the social networks for each node, however, data, such as encounters, must be collected so that social networks can be *detected* or discovered. This can lead to delays in bootstrapping the DTN which may impede its effectiveness. We observe that many networked applications involve communicating with a recipient that is known to the sender, i.e., a node that is part of the sender's existing social network. Such a social network is already known to a node, and so requires little time to generate compared to the social network collected from an encounter trace.

It is well known amongst social science researchers [2], [3], that self-reported and detected social networks will differ. A recent study by Mtibaa et al. [4], however, looked at self-reported and detected social networks amongst a group of conference attendees, finding that the two social networks are similar.

This paper thus attempts to answer two questions:

1) Are detected social networks and self-reported social networks similar?
2) If they are not similar, how does this affect network routing?

We structure the paper as follows. Section II outlines our network and experimental setup. In Section III we compare the two types of social network, and in Section IV we demonstrate the effect of these two networks on delay-tolerant networks. Section V describes the growing body of related work in this area, and Section VI concludes with our plans for future work.

II. Experimental setup

As part of a research project, we have set up a mobile sensor network comprising mobile IEEE 802.15.4 sensors (T-mote invent devices) carried by human users, and Linux-based basestations that bridge the 802.15.4 sensors to the wired network. T-mote invent devices can detect each other within a radius of $\sim$10m. These encounters are stored in the invent devices and are uploaded through the basestations to a central database.

For the experiments described in this paper, we deployed 27 invent devices amongst 22 undergraduate students, 3 postgraduate students and 2 members of staff. To upload encounters, we deployed three basestations across the two Computer Science buildings in our institution. Participants were asked to carry the devices whenever possible over a period of 79 days. We could detect invent-to-invent encounters anywhere throughout the town of St Andrews and beyond; they are not limited by basestation placement.

The invent devices were programmed to broadcast a beacon every 6.67 seconds[2]. When other devices detect these beacons, they record a timestamp (and other information such as signal strength) for this beacon. This timestamp, the sending device's ID and other information form a Sensor Encounter Record (SER). When these SERs are uploaded to a basestation, the basestation adds the ID of the uploading device and the basestation's ID, and records these in a central database.

We used the participants' *Facebook*[3] social network information to generate a topology . We refer to this as the ***self-reported social network (SRSN)*** following the terminology of [5]. We also generate a topology using the SERs to create a social network, similar to [6]. We refer to this as the ***detected social network (DSN)*** following the terminology of [7], [8].

III. Self-reported and detected social networks

Before we can examine whether the use of SRSNs instead of DSNs has an impact on DTN performance, we must first answer our first question: *Are detected social networks and self-reported social networks similar?* Figure 1(a) and Figure 1(b) show the topologies for the SRSN and DSN. We can observe differences between the two networks, but in order to better understand these differences, we employ techniques from traditional social network analysis.

A. Structural equivalence

The notion of *structural equivalence* allows us to compare the ties between nodes (or *actors* in social network analysis terminology) in social networks. Actors who have identical relationship ties to the same group of actors are *structurally equivalent* and are referred to as being in the same *equivalence class*.

To calculate structural equivalence we create a matrix of the ties between actors (a *sociomatrix*); if an actor i has a social tie to actor j, then the element (i, j) has a value of 1; otherwise the value is 0. If actors i and j are structurally equivalent, the entries in their respective rows and columns of the sociomatrix will be identical (i.e., the Euclidean distance between them is 0). By computing distances between all n actors in the network, we create an $n \times n$ matrix that shows the structural equivalence of each actor.

Using these Euclidean distances as a metric, we plot *dendrograms* for the SRSN (Figure 2(a)) and DSN (Figure 2(b)). Dendrograms can be used to understand clustering; we say the nodes are clustered, where each mutual cluster is a set of nodes whose largest intra-group distance is smaller than the distance to the nearest point outside the set. For us, the important feature to note is that nodes on the same 'branch' of the dendrogram are considered to have shorter distances between them, and are said to be clustered.

B. Role equivalence

Closely related to the concept of structural equivalence is *role equivalence*. This allows us to examine clusters in a social network and also compare the clusters between different social networks. Two actors i and j are role equivalent if the collection of ways in which i relates to other actors is the same as the collection of ways in which j relates to other actors [9]. To examine role equivalence graphically, we use *blockmodels* following [10]. The blockmodel can be thought of as a graphical abstraction over the sets of ties between nodes, and the function to produce it takes the clustered nodes from the dendrogram as input. Each block in the blockmodel indicates whether or not the column actor can be though of as having equivalent ties to other nodes as the row actor. A position is assigned to each actor in the network. For each pair of structural equivalence relationships (node ties), we determine whether there exists a tie between the positions of the relationships, i.e., do the ties from actor j match actor i? If there is sufficient overlap to satisfy the equivalence criteria (in our case: is there at least one match in every row and column of i and j's role sets), then a block is added to the blockmodel diagram in the jth column for the ith row.

C. Network analysis

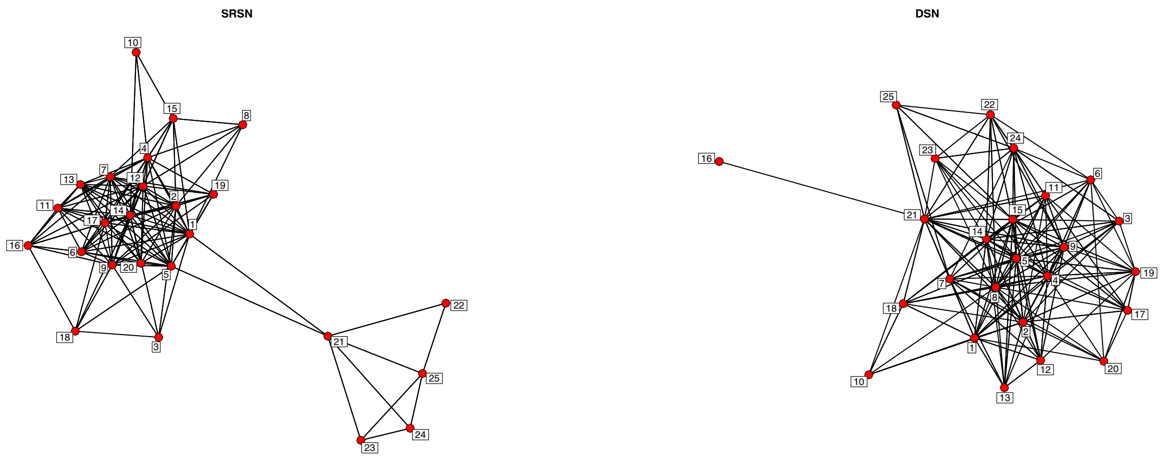
From the SRSN blockmodels we found three roles, each of which can be seen as a cluster in the dendrogram, or as a section of the blockmodel.

Nodes $21, 22, 23, 24, 25$ *form role 1:* the postgraduate and staff group. They are found on the left of the dendrogram (Figure 2(a)) and are the small outlying cluster of nodes in the social network graph (Figure 1(a)). We observe that nodes in this group mainly only have ties to each other, and the only ties to the rest of the network are through node 21.

Nodes $1, 2, 4, 5, 7, 9, 12, 14, 17$ *form role 2.* These nodes are almost entirely connected to themselves and role 3 (see below). These nodes are the most popular, with lots of ties and to the widest variety of nodes. In Figure 1(a), these nodes are located close to the centre. These nodes represent a *clique* — they all have ties to each other, but there are no other nodes that are directly connected to all the members of the group [9].

[2]Found experimentally using two walking users each with a device; details omitted due to space restrictions.

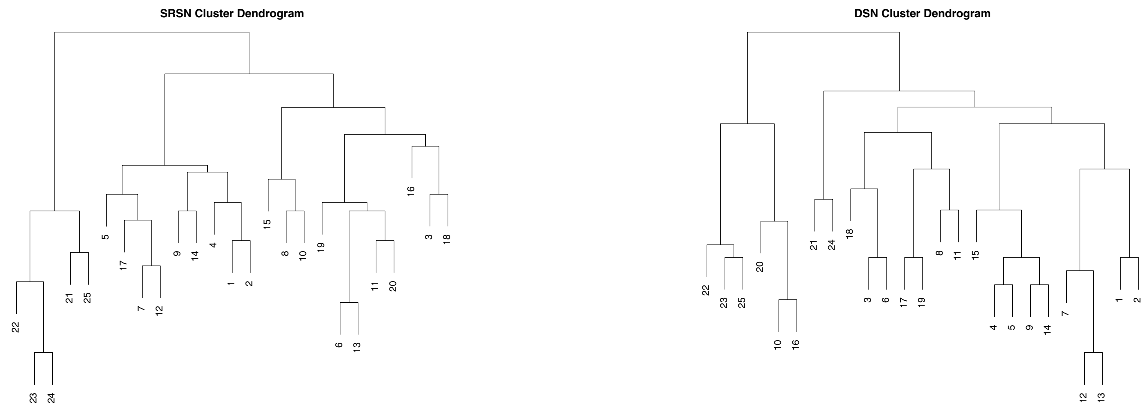
[3]http://www.facebook.com/



(a) The SRSN graph. There are two groups of nodes, the small group is the staff and postgraduates, and the large group is the undergradaute student group.

(b) The DSN graph. At first glance it appears all participants bar one seem to be in the same group. On average nodes in the DSN have more ties.

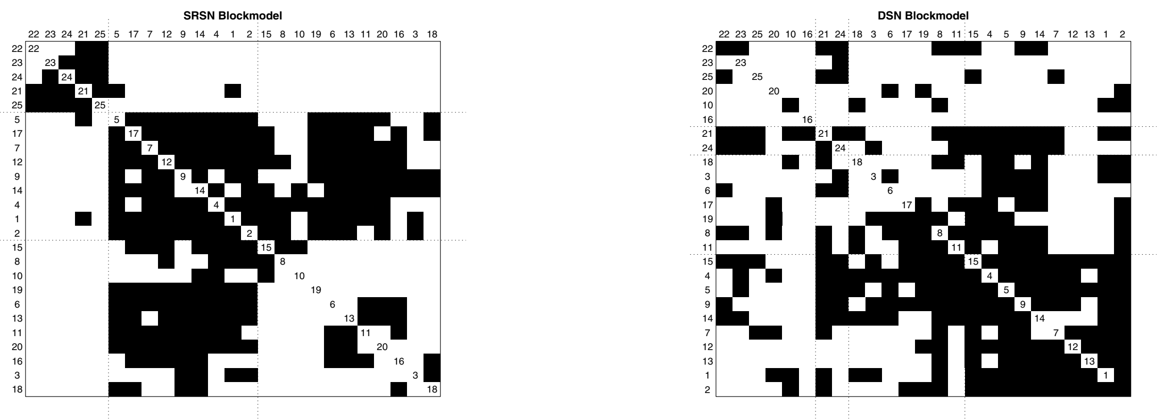
Fig. 1. The topologies of the SRSN and DSN. The numbers are consistent across both plots, i.e. node 1 in the SRSN is also node 1 in the DSN.



(a) Dendrogram for SRSN, clustered by Euclidean distance. We observe three different clusters of nodes.

(b) Dendrogram for DSN, clustered by Euclidean distance. We observe four different clusters of nodes.

Fig. 2. The equivalence clustering Euclidean distance dendrograms.



(a) Blockmodel for the SRSN. There are three clearly-defined roles within the social network.

(b) Blockmodel for the DSN. There are four weakly-defined roles.

Fig. 3. Blockmodels of role equivalence for the SRSN and DSN. Dotted lines indicate role divisions.

Nodes $3, 6, 8, 10, 11, 13, 15, 16, 18.19.20$ *form role 3.* These nodes are tied mainly to members of role 2, and so rely on members of this role to communicate with each other. In Figure 1(a) these nodes are located around the edge of the largest group.

We found four weakly-defined roles in the DSN blockmodel.

Nodes $10, 16, 22, 23, 25$ *form role 1.* These nodes are outlying on the graph(Figure 1(b)) and they have the fewest connections that are useful to other nodes.

Nodes 21 *and* 24 *form the second cluster and role 2.* These nodes have connections to most of the nodes in roles 1 and 4, and only a few connections to role 3. Crucially, they provide links to the more central nodes for those in the outlying cluster, and hence do not fall into the outlying cluster themselves.

Nodes $3, 6, 8, 11, 17, 18, 19$ *form role 3.* They lie near to (but not in) the centre of the network. They can also be considered edge nodes, but they have more connections into the centre than role 1, and are also highly connected to each other.

Nodes $1, 2, 4, 7, 9, 12, 13, 14, 15$ *form role 4.* This role has the most ties to other groups. They have few connections to role 1 but are largely interconnected to each other and a variety of nodes across clusters.

On average, nodes in the DSN have a greater number of ties than in the SRSN. In both cases the roles indicated in the blockmodel confirm the clusters described in the respective dendrograms, and also help us to distinguish the roles more clearly. The roles are less well defined in the DSN, since the blockmodel does not show as obvious divisions as in the SRSN. The SRSNs roles seem to form more blocky structures with similar relations to each other, and with clear boundaries. In the DSN, however, divisions seem to be distinguished by number of ties to the center of the network. This is a feature of the blockmodel that would not have been obvious from simply inspecting the topology diagrams.

D. Hypotheses

Nodes in the DSN have a greater number of ties than the SRSN, and those ties are more frequently to nodes from a different role. So we infer that the nodes in the DSN may be more easily reached. We see in Figure 1(a) and Figure 3(a) that there are fewer paths and more clearly defined roles than the DSN equivalents Figure 1(b) and Figure 3(b).

We use similar metrics to those listed in [11], namely:

- delivery ratio — the proportion of messages that have been delivered to the total number of messages created.
- delivery cost — the total number of medium accesses, normalised by the total number of messages created.[4]

We make the following hypotheses:

1) A DTN application using a SRSN as a routing table would have a lower delivery ratio than one using a DSN, since there are fewer links available, and nodes are more restricted in the ways they relate to each other.
2) A DTN application using a SRSN as a routing table would have a lower delivery cost than one using a DSN, since there are fewer links available, and so nodes are likely to generate fewer transmission.

[4]Results for delivery delay are omitted because of space restrictions

IV. Experimental results

We now test the hypotheses of Section III-D. We use the SRSN and DSN as inputs to a simulated DTN and calculate delivery ratio and delivery cost metrics to determine message-passing performance.

A. Simulation setup

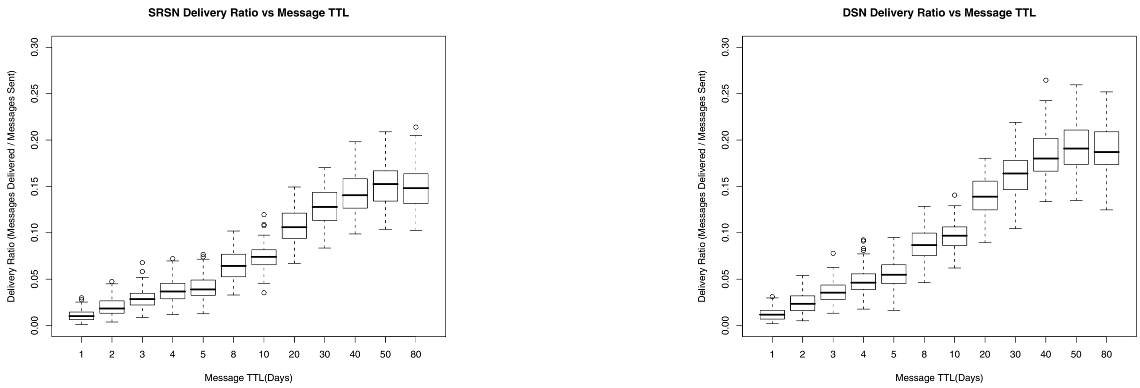
An important factor in the design of a DTN is the amount of time that it takes to pass a message from one node to another – the *bundle transfer time*. If messages are large and take a long time to transfer between nodes, then we are reliant on nodes encountering each other and staying within range of each other for a long time in order for the DTN to be effective. We assume a bundle transfer time of 30 seconds, which we believe to be a reasonable minimum encounter time. We hold this bundle transfer time constant and experiment with different time to live (TTL) values: this allows us to determine DTN performance under different application scenarios. For instance, a disaster scenario application might require a low message TTL, whereas a holiday maker sending photos to their friends may not worry about timely delivery and so be content with a high TTL.

The SER data were used to create a trace-driven simulation. For each day of the simulation, we randomly selected 20 nodes to send 20 messages to 20 randomly selected destinations. At each encounter, a message was passed to the encountered node if the encountered node was in the source's SRSN or DSN, respectively. The encounters were condensed as follows: bundles of messages were assumed to be passed within 30 seconds, so any encounters less than this period of time were not included in the simulations. We performed 100 runs of the experiments at each TTL (TTL values used: $1, 2, 3, 4, 5, 8, 10, 20, 30, 40, 50, 80$ days).

B. Delivery ratio

Figure 4(a) and Figure 4(b) indicate similar trends for both DSN and SRSN. As might be expected, the delivery ratio increases along with the TTL. Both DSN and SRSN result in a jump in delivery ratio between a TTL of 10 and 20 days — this was due to sudden increase in activity. Both social networks result in somewhat low delivery ratios overall, with a maximum of 26% of the messages being delivered. This is an artefact of our trace; it turns out that the participants in our experiment did not encounter each other sufficiently frequently to create a very effective DTN.

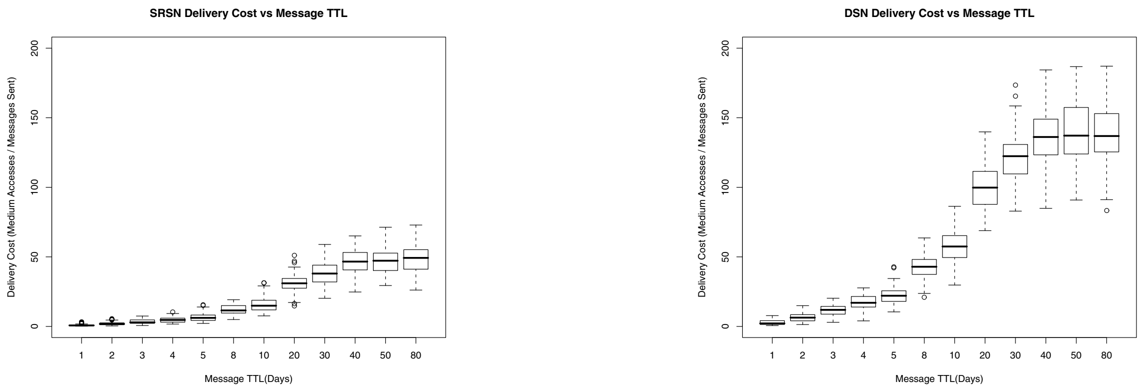
The DSN results in a slightly higher delivery ratio than the SRSN (at most 5%). This difference was found to be statistically significant using a t-test ($p < 0.01$).



(a) Delivery ratio compared to message TTL, when the SRSN is passed at each hop.

(b) Delivery ratio compared to message TTL, when the DSN is passed at each hop.

Fig. 4. The delivery ratios for SRSN versus DSN



(a) Delivery cost compared to message TTL when the SRSN is passed at each hop.

(b) Delivery cost compared to message TTL, when the DSN is passed at each hop.

Fig. 5. The delivery costs for SRSN versus DSN.

C. *Delivery cost*

Figure 5(a) and Figure 5(b) show the delivery costs for the SRSN and DSN simulations. We see that the SRSN results in a dramatically lower delivery cost, with a maximum of 73 medium accesses per message for the SRSN compared to a maximum of 187 for the DSN. Like the delivery ratio, the difference in delivery cost is found to be statistically significant via a t-test ($p < 0.01$). This is because the SRSN exploits fewer links, since it forces the intermediate nodes to ignore encounters not in the source social networks that may prove useful, and also provides links to the intermediate node to try and use that it may never meet.

V. Related work

Several researchers look at detecting social networks and using these for routing in MANETs and DTNs. Eagle et al. [5] analyse nine months of mobile phone traces from 94 users (the "Reality Mining" dataset). They find a mildly positive correlation between the self-reported and detected social networks. Perkio et al. [12] also use Bluetooth connections between mobile phones to detect social networks. Hui et al. [13] attempt to find *communities* within the Reality Mining dataset and three traces from the Haggle DTN project.[5] Messages are forwarded within a source's community, and if the destination does not exist within this community, the message is passed up to a hierarchy of communities. Daly and Haahr [6] use betweenness centrality (a measure of a nodes control over information flow) for routing in DTNs, and find that detected social networks may provide useful routing metrics.

The use of encounters for studying mobility in MANETs and DTNs has been examined elsewhere, e.g., by collecting encounters from Bluetooth devices [14], [15], mobile phone records [16], or student class records [17].

In addition to use for routing, social networks can also be used for improving mobile applications. Miklas et al. [18] use social networks to improve three different mobile applications: DTN routing, mitigating the spread of worms, and mobile

[5]http://www.haggleproject.org/

peer-to-peer file-sharing. The CenceMe [19] and Participatory Sensing [20] projects involve users carrying sensors to create new sensor-aware applications. Messages from such applications may be forwarded using existing mobile infrastructure networks or DTNs. Social networks and communities have also been used to build improved publish/subscribe architectures [21], [22].

Contrary to the literature, we noted in Section I, that Mtibaa et al. [4], found that SRSNs and DSNs were similar. This may be because of the short duration of their study (3 days) our study was 79 days. Moreover, their use of conference attendees may not be as representative a sample as our study. Of course, our study has similar limitations in that we only examine one cohort of students, but we hope to conduct further and more wide-ranging experiments in the future.

VI. Conclusions and Future Work

This paper has explored the use of self-reported social networks (SRSNs) in comparison with detected social networks (DSNs) for routing in mobile delay-tolerant networks (DTNs). We make two contributions:

- We find that social network analysis techniques can be used to compare self-reported and detected social networks. Based on our encounter records for 27 people over 79 days, we show that the two social networks differ in terms of structural equivalence and role equivalence.
- In our trace-driven simulation, we show that using a SRSN for routing in a delay-tolerant network scenario produces comparable delivery ratio to a DSN (although our experiments did show a statistically significant difference, the volume of traffic was only 5%), but SRSNs have a much lower delivery cost than DSNs.

Our immediate plans include exploring the use of SRSNs to bootstrap routing and determining appropriate thresholds for switching between the use of SRSNs, DSNs, or a combination of the two. We hope that the use of role equivalence may play a part here: it may be more effective to forward to nodes in certain roles. We also plan larger and more comprehensive experiments to collect further SRSN and encounter information, since at the moment we cannot make any predictions on how an increased number of users would affect results.

References

[1] International Telecommunications Union, "Report on the world summit on the information society stocktaking," ITU, Tech. Rep., May 2008. [Online]. Available: http://www.itu.int/wsis/stocktaking/docs/2008/WSIS-Stocktaking2008-e.pdf

[2] H. R. Bernard, P. D. Killworth, and L. Sailer, "Informant accuracy in social-network data V. an experimental attempt to predict actual communication from recall data," *Social Science Research*, vol. 11, no. 1, pp. 30–66, March 1982.

[3] S. R. Corman and L. Bradford, "Situational effects on the accuracy of self-reported organizational communication behavior," *Communication Research*, vol. 20, no. 6, pp. 822–840, December 1993.

[4] A. Mtibaa, A. Chaintreau, J. Lebrun, E. Oliver, A. K. Pietilainen, and C. Diot, "Are you moved by your social network application?" in *Proc. the First ACM SIGCOMM Workshop on Online Social Networks (WOSN)*, Seattle, WA, USA, Aug. 2008.

[5] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring social network structure using mobile phone data," 2008, in submission. [Online]. Available: http://www.socialsciences.cornell.edu/0508/sciencereport_formatted_10.12.pdf

[6] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETs," in *Proc. ACM MobiHoc 2007*. New York, NY, USA: ACM Press, 2007, pp. 32–40.

[7] M. E. J. Newman, "Detecting community structure in networks," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 38, no. 2, pp. 321–330, 2004.

[8] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proc. ACM MobiArch Workshop 2007*. New York, NY, USA: ACM, 2007, pp. 1–8.

[9] S. Wasserman, *Social network analysis: methods and applications*. Cambridge University Press, 1997.

[10] H. C. White, S. A. Boorman, and R. L. Breiger, "Social Structure from Multiple Networks. I. Blockmodels of Roles and Positions," *The American Journal of Sociology*, vol. 81, no. 4, pp. 730–780, 1976.

[11] P. Hui and J. Crowcroft, "How small labels create big improvements," in *Proc. the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*. Washington, DC, USA: IEEE Computer Society, Mar. 2007, pp. 65–70.

[12] J. Perkio, V. Tuulos, M. Hermersdorf, H. Nyholm, J. Salminen, and H. Tirri, "Utilizing rich Bluetooth environments for identity prediction and exploring social networks as techniques for ubiquitous computing," in *Proc. the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, Hong Kong, China, Dec. 2006, pp. 137–144.

[13] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *Proc. ACM MobiHoc 2008*, Hong Kong, China, May 2008.

[14] E. Nordström, C. Diot, R. Gass, and P. Gunningberg, "Experiences from measuring human mobility using Bluetooth inquiring devices," in *Proc. the 1st international workshop on System evaluation for mobile platforms*, San Juan, Puerto Rico, Jun. 2007, pp. 15–20.

[15] E. O'Neill, V. Kostakos, T. Kindberg, A. F. gen. Schiek, A. Penn, D. S. Fraser, and T. Jones, "Instrumenting the city: developing methods for observing and understanding the digital cityscape," in *Proc. UbiComp 2006*, Irvine, CA, USA, Sep. 2006, pp. 315–332.

[16] M. C. Gonzalez, C. A. Hidalgo, and A. L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, June 2008.

[17] V. Srinivasan, M. Motani, and W. T. Ooi, "Analysis and implications of student contact patterns derived from campus schedules," in *Proc. ACM MobiCom 2006*. New York, NY, USA: ACM, 2006, pp. 86–97.

[18] A. Miklas, K. Gollu, K. Chan, S. Saroiu, K. Gummadi, and E. de Lara, "Exploiting social interactions in mobile systems," in *Proc. UbiComp 2007*, Innsbruck, Austria, 2007, pp. 409–428.

[19] E. Miluzzo, N. D. Lane, S. B. Eisenman, and A. T. Campbell, "CenceMe – injecting sensing presence into social networking applications," in *Proc. EuroSSCS 2007*, Lake District, UK, Oct. 2007, pp. 1–28.

[20] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *Proc. Workshop on World-Sensor-Web (WSW'2006)*, Boulder, CO, USA, Oct. 2006.

[21] E. Yoneki, P. Hui, S.-Y. Chan, and J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks," in *Proc. MSWiM 2007*, Crete, Greece, Oct. 2007, pp. 225–234.

[22] W.-J. Hsu, D. Dutta, and A. Helmy, "Profile-cast: Behavior-aware mobile networking," in *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, Las Vegas, NV, USA, Mar. 2008, pp. 3033–3038.