Bootstrapping Opportunistic Networks Using Social Roles

Greg Bigwood and Tristan Henderson
School of Computer Science, University of St Andrews, St Andrews, Fife, UK
{gjb4,tnhh}@st-andrews.ac.uk

Abstract

Opportunistic routing protocols can enable message delivery in disconnected networks of mobile devices. To conserve energy in mobile environments, such routing protocols must minimise unnecessary message-forwarding.

This paper presents an opportunistic routing protocol that leverages social role information. We compute node roles from a social network graph to identify nodes with similar contact relationships, and use these roles to determine routing decisions. By using pre-existing social network information, such as online social network friends, to determine roles, we show that our protocol can bootstrap a new opportunistic network without the delay incurred by encounter-history-based routing protocols such as SimbetTS. Simulations with four real-world datasets show improved performance over SimbetTS, with performance approaching Epidemic routing in some scenarios.

I. Introduction

The prevalence of mobile devices carried by users, such as smartphones, has led to an interest in new network architectures that exploit these devices in different scenarios. One such proposal is opportunistic networking, which attempts to enable the forwarding of data via encounters between devices, while dealing with the problems of large delays and a lack of end-to-end connectivity.

A main challenge in opportunistic networks is routing: how can one find the best device to relay a message? Any messages that are passed to a device that is unlikely to be on the path to the destination device are a waste of bandwidth, storage and energy. Thus, an opportunistic network routing protocol should aim to provide high delivery ratios, while reducing the number of unnecessary messages.

Many researchers have conducted social network analysis of encounters between nodes to create opportunistic routing protocols, exploiting the fact that a node that has encountered a node in the past may be likely to encounter this node in the future. Such protocols can suffer from two problems. First, nodes that are best-connected in the social network (i.e., nodes with the most "friends") might be selected very frequently for forwarding messages. This may lead to those nodes running out of energy or storage. It has been shown as much as 63% of traffic is routed through only 10% of nodes when using protocols which rely on only the most preferred nodes for forwarding [1]. It may therefore be beneficial to consider alternative nodes than only using the most obviously suitable.

Second, collecting encounter information on which to analyse social networks may require time and a mechanism for aggregating these collected data. This leaves a window of time between network start-up and the point in time at which the protocol has built a model reliable enough to make accurate inferences for routing. During this period nodes may experience sub-optimal routing behaviour, reducing network performance. Using an alternative mechanism for bootstrapping the network, before the encounter information can be gathered, may improve routing performance.

This paper introduces a new opportunistic routing protocol, Social Role Routing (SRR), which attempts to solve these two challenges. Rather than depend on social networks derived from collected encounter patterns, we use information from pre-existing social networks to drive routing decisions. To derive alternative nodes for forwarding, we determine the social *roles* in the social network to find classes of nodes that may be useful for forwarding. Our hypothesis is that self-reported social networks are sufficient to bootstrap the opportunistic network, even if those self-reported networks may differ from actual encounter patterns. Our proposed protocol enables an opportunistic network to be bootstrapped while providing high performance, before encounter-history-based routing protocols' forwarding decisions have stabilised.

This paper is structured as follows. We next discuss existing opportunistic network protocols, and introduce our social-role-based routing protocol in Section III. Section IV describes a set of trace-driven simulations using four real-world mobile traces, and the results in Section V demonstrate that our protocol is viable in general and bootstrapping scenarios. Finally we conclude and discuss our current work.

II. Related Work

Opportunistic networking [2] targets environments where mobile nodes wish to communicate with each other in the absence of any static connecting routes. All nodes in the network act as forwarding gateways, and nodes forward packets *opportunistically* between nodes as they encounter each other, using whatever mechanism may be available. For instance, if Alice wishes to send a message to Bob, Alice may use the Bluetooth radio on her mobile phone to forward a message to Charlie's phone, who then uses the phone's 802.11 interface to forward the message to a passing train, and so on until the message may or may not reach Bob. Choosing the intermediate nodes that are likely to result in a message reaching its destination is one of the primary challenges for an opportunistic routing protocol.

The SimbetTS [3] protocol decides whether a node should forward a message to an encountered node using a combination of three social-network-analysis measures: betweenness [4], similarity (the number of ties that two nodes share) and tie-strength (the recency, duration and number of encounters between two nodes). As nodes encounter each other, they share encounter histories, update their utility scores (the normalised sum of the three measures), and use this utility to decide whether to forward a message.

BubbleRap [5] also uses social network information, by grouping nodes into communities. By ranking nodes within a community, and having hierarchical communities, BubbleRap allows nodes to "bubble up" a message to a node if it has a higher rank within the same community or is a member of a community that is closer to the destination.

These, and other existing protocols such as Habit [6], use metrics based on locally-available information. Such information is useful since it can be calculated from a node's own encounters, without requiring any global knowledge. Using an encounter history to make an informed decision, however, means that a number of encounters must take place before the routing protocol works as intended. Thus, the SimBetTS authors propose that some time (15% in their simulations) be reserved as a "warm-up" period, during which nodes gather encounter information, but do not forward messages.

Other protocols, like ours, use pre-existing social network information, or what we hereafter refer to as self-reported social networks (SRSNs). Mtibaa et al. [7] use SRSN information from online social networks to compute node rankings. The rationale is that nodes with a higher rank will generally be more central, and therefore good candidates for exchanging messages. Pietiläinen et al. [8] describe an architecture that uses SRSNs to bootstrap an opportunistic network. Nodes use their SRSN as a list of permitted intermediaries, and can decide to add nodes to this list over the network lifetime. Boldrini et al [9] also consider roles, but these are based on users' behavioural history and not on graph structure.

Our proposed protocol builds on these existing protocols in two ways. First, we demonstrate that SRSN information can be used to build a protocol that operates effectively without the need to accumulate node encounter histories, making it useful for bootstrapping a network. Second, unlike existing protocols, we make inferences from the SRSN information, determining social roles in an attempt to avoid overloading popular nodes. Our protocol uses information gathered from knowledge of the entire network, but during network operation nodes only need knowledge of a small network graph to work effectively.

III. Role analysis for routing

In society we are members of various social networks, from work relationships to sports clubs to connections with our neighbours. Our goal is to use a record of these social network data to create SRSNs, such as our Facebook "friends", that can aid opportunistic routing. By constraining the devices that can use a particular node as an intermediary for message delivery to only those members of a node's SRSN, we may save energy (due to the reduced number of forwarded messages) at little expense (as a node's SRSN members may be more likely to aid message delivery).

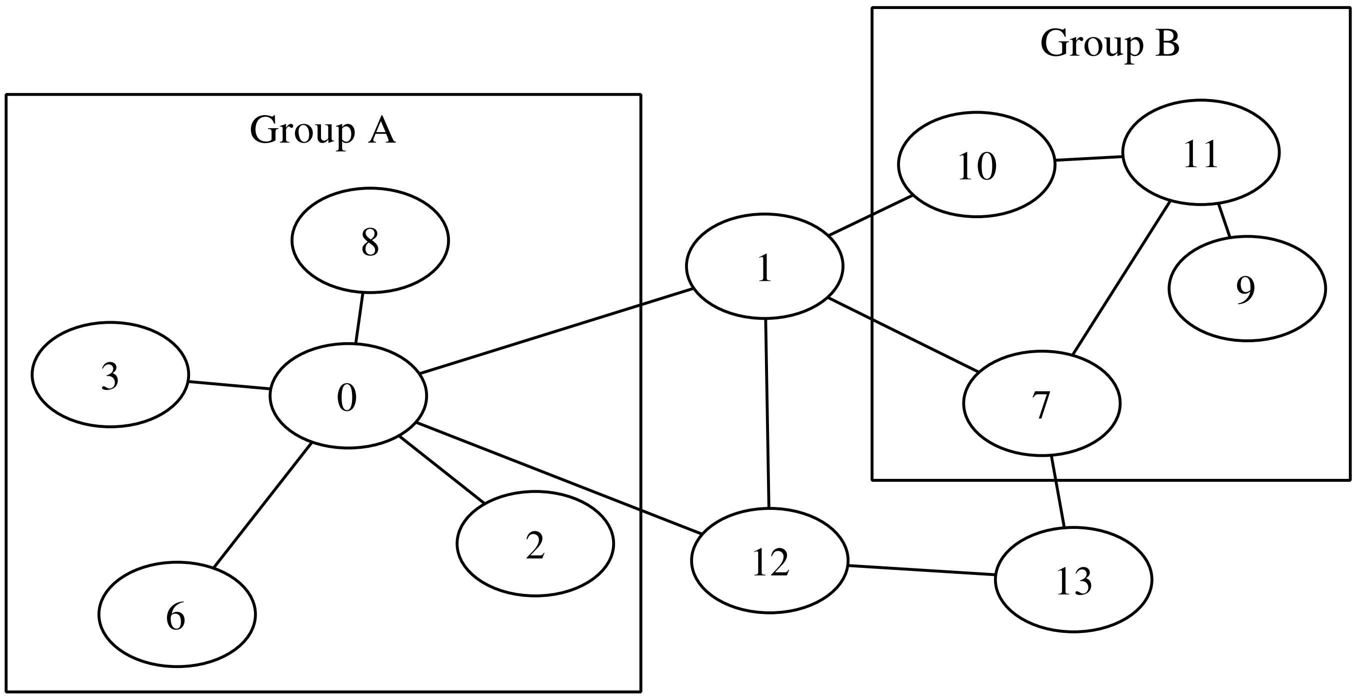


Fig. 1. An example node contact graph. Routing messages from group *A* to *B* via the shortest path (node 1) may overburden the device's resources. Utilising an alternative path (nodes 12 and 13) for some messages may be more appropriate.

Certain nodes may be more central in an SRSN, however, and therefore required to do a larger share of message-forwarding on behalf of other devices. This may lead to the device being overloaded. For example in Figure 1, all messages sent by nodes in group *A* to group *B* would likely go through node 1 (as it is on the shortest path). Node 1 might therefore run out of battery quickly, or be less likely to participate in an opportunistic network if its resources were to be heavily-utilised. We thus need to find nodes that can function similarly to node 1 (in that they connect group A to group B). In this case messages could be routed via nodes 12 and 13, which can be used as an alternative forwarding path. This is similar to using alternative path forwarding in MANETs [10] to avoid overloading nodes.

As well as being members of social networks, we function as members of various roles. For instance, we can view a business as a network of employees, all of whom have particular roles defining their connections to the other employees. Assistant managers connect managers to non-managerial employees. If an assistant manager is unavailable to pass on instructions to employees, a manager may select another assistant manager to do so, as all members in the assistant manager role perform the same task (giving instructions).

We extend this to opportunistic routing. By passing messages to nodes performing a similar role in an opportunistic network, we can find alternative nodes: to avoid overloading preferred nodes, or when preferred nodes are unavailable.

We focus on roles derived from the graph structure of the SRSN, where we expect that the explicit declaration of contacts is sufficient to provide roles that will be useful in opportunistic forwarding. While it is possible to derive roles from node encounters, it is outside the scope of this paper.

A. *Determining roles for forwarding*

To categorise nodes in an opportunistic network into roles, we employ the social science technique of regular equivalence [11]. This partitions nodes into classes, where all nodes in a class are connected to the same classes of nodes. In the aforementioned business example, all nodes in the manager role connect to at least one node in the assistant manager role, who in turn connect to at least one node in the non-managerial employee role. These roles can then be used to drive routing decisions. Consider three roles $(0,1,2)$, connected in the following manner: $\{0,1\}$,$\{1,2\}$. If node a in role 0 intends to pass a message destined for node d in role 2 to intermediate node b in role 1, but instead meets node c from role 1, it may be beneficial for a to pass the message to c, since a and b may never meet. Note that nodes b and c do not need to ever have contact, despite being members of the same role. Connections between roles (and those within roles) are taken to be an indicator of expected interaction.

Using blockmodelling to break down the nodes into regularly equivalent classes [12] is common method in sociology. However we would need to decide on the number or roles using inspection. Instead we use an approach that does not require human involvement in the calculation of roles. Following [13], we use betweenness centrality to create the initial partition, place nodes with matching betweenness in the same classes, and use the Kanellakis-Smolka algorithm [14] to assign roles. This results in a Role Connectivity Graph (RCG) of connections between roles.

Strict regular equivalence results in many roles of size one, and a number of roles similar to the number of nodes in the input graph. Such roles are impractical for forwarding, since if this sole node in a role is unavailable, there would be no alternative node available for forwarding. Thus, we relax this constraint and enforce a minimum role size of two.

B. Social Role Routing

Our routing protocol, Social Role Routing (SRR), utilises role analysis of pre-existing social contact information.

As reducing message duplication can prolong battery lifetime we only allow forwarding of messages to intermediate nodes that are in the same role, or in a role adjacent to, the destination's role in the RCG. Since they are likely to see nodes in the destination's role, and therefore by extension, the destination, nodes in or adjacent to the destination's role are good candidates for intermediary nodes.

Before the network starts up we analyse the SRSN of the participating nodes using the method described in Section III-A. Each node stores a copy of the resulting RCG, allowing them to compute the geodesic distance between roles. Each node has a unique identifier (ID) and stores the identifier of the role to which it belongs (RoleID).

When nodes encounter one another, each provides its ID, RoleID and identifiers of each of the messages that it is carrying. For each message in its message buffer not present in the other node's message buffer, the node will check the geodesic distance of the encountered node's role from the destination node's role. If this distance is less than or equal to 1 (i.e., it is in the same role or in a role adjacent to the destination's role), the node duplicates the message and passes it on to the encountered node.

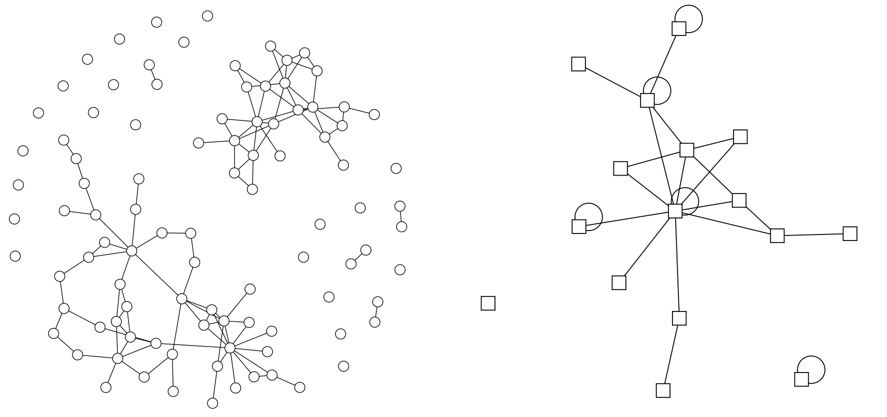
IV. Evaluation

Since the performance of an opportunistic network may vary depending on the connectivity patterns of the nodes, we use four different real-world traces in our analysis:

1) Our "SASSY" connectivity trace [15]. 24 individuals carried T-mote sensor nodes for 3 months. We use the motes' ZigBee radios to detect co-location and collected the participants' *Facebook* "friends" as SRSNs.
2) The MIT Reality Mining trace [16]. 99 individuals carried mobile phones using Bluetooth to detect co-location. We use users' phone contact lists as SRSNs.
3) The NUS student schedule trace [17], which comprises $22,341$ student timetables ($4,885$ sessions, with an average of 40.25 students per session). Students in the same lecture are assumed to be in contact with one another, and provides the encounter trace. The timetable is presented as contiguous hours without breaks overnight. To improve realism we assume a 6 day work-week of 11.5-hour days. As this is a dense dataset, to reduce simulation complexity the method from [18] is used to reduce the number of nodes in the dataset. Students attending the same courses are assumed to be in the same SRSN. To increase scenario realism, we randomly remove 34% of nodes each day from the encounter trace, assuming that these students did not attend classes [19].

4) The Hope dataset [20] of the movements of 767 attendees at the Hackers On Planet Earth conference. Participants registered their interest in topics and specific sessions before attending the conference, where participants were given RFID tags to carry. We use matching interests as the SRSN connections, and the encounters are taken from the provided RFID readings. As the dataset is very dense, and participants and most of the timetable is taken up by talks, we merge all contacts between pairs of nodes within one hour.

We believe these traces capture a wide variety of possible network scenarios. Table I shows the number



(a) The SRSN features two disconnected groups of nodes, four groups of outlying pairs, and 22 disconnected nodes.

(b) The RCG has a highly important central role, and all the outlying nodes (Figure 2(a)) have been grouped together.

Fig. 2. Plots of the Reality Mining self-reported social network and the resultant role connectivity graph. Nodes are represented as circles, roles as squares, loops indicate members of a role connect to one another.

of nodes varies in each trace, with a similar variation in the number of connections in each trace. The SASSY SRSN is a large component, the Reality Mining SRSN: two large components, four pairs of nodes and 23 disconnected nodes. The NUS trace has two components, all nodes are highly connected. The Hope trace is a highly connected component.

A. Role Analysis

Figure 2(b) shows the RCG for the Reality Mining trace. Each vertex represents one role, and may have two or more members. Summary statistics of each trace's SRSN graph and RCG can be seen in Table I.

For each trace, the graph density is higher in the RCG, which can lead to a wider distribution of messages, as each node can consider more potential contacts than they would in the SRSN. With a large number of relays available, messages can be routed around overloaded nodes if required. In all cases the number of vertices has been reduced from the input partition, meaning that storing the RCG requires less memory than storing the whole SRSN graph.

Central roles can be seen in both the SASSY and Reality Mining RCGs. Notice that in the Reality Mining SRSN several nodes have no ties to other nodes, and using a forwarding protocol based purely on the SRSN data might lead to the assumption that these nodes make no contacts during the trace. In the RCG graph they are now members of a role. We also see that the two large disconnected groups are now connected together. This shows that role analysis can find similarities between nodes that are not in the same components, as all nodes in a role are expected to connect to members of the same roles as the each other.

In all of the RCGs the central role/roles connect to themselves, showing that nodes in this role form a group/cluster (note that they do not necessarily connect to all members of their role). Many roles, however, do not connect to themselves, showing that these roles act as intermediaries between other roles. Potentially these roles could be useful to drive a source-routing protocol, paths for the messages could

TABLE I
DATASET GRAPH STATISTICS

Property	SRSN	RCG	SRSN	RCG
	SASSY		Reality Mining	
Number of Vertices	25	5	97	16
Number of Edges	127	7	107	23
Clustering Coefficient	0.748	0	0.255	0.246
Graph Density	0.406	0.560	0.023	0.180
Graph Components	1	1	28	3
	NUS		Hope	
Number of Vertices	500	26	414	27
Number of Edges	29743	198	67836	329
Clustering Coefficient	0.605	0.808	0.834	0.885
Graph Density	0.238	0.586	0.792	0.903
Graph Components	2	1	1	1

be specified in the role network rather than in the device network. Nodes could then opportunistically forward using whichever members of the required roles are available for forwarding, rather than relying on specific nodes (as they would in a standard source-routing protocol).

B. Routing Protocols

We evaluate the following routing protocols:

- Epidemic [21]: Nodes forward messages to any encountered nodes that do not already have a copy.
- SimbetTS [3]: This is an example of a well-known existing history-based opportunistic routing protocol.
- Social Role Routing (SRR): as described in Section III-B.
- Social Role Routing SimbetTS Hybrid (HySimbR): Many routing protocols use a warm-up period (such as SimbetTS [3]) in which to gather enough data to make accurate routing decisions. Here we switch from SRR to SimbetTS after 15% of encounters. This simulates the performance of a system using SRR to bootstrap a network before switching to a history-based protocol.

C. Routing Evaluation

TABLE II
SIMULATION PARAMETERS

Parameter	Value
TTL of messages	10 days / 1 day / 2 hours
Message frequency	1 per node per day
Simulation length	30 days / 7 days / 1 day
Message size (MB)	1
Buffer size (MB)	2000
Loss per second (mAh)	1.9×10^{-6}
Time to send bundle (s)	34
Max energy (mAh)	1200
Energy per send (mAh)	0.4
Charge time (h)	8

To evaluate SRR against existing protocols, we use trace-driven simulations of smartphones running a message-passing application. This allows us to compare several opportunistic routing protocols. In our scenario we use the simulation parameters described in Table II, which are in line with other work in this area, e.g., [22].

To reduce the effect of specific behaviour in segments of the trace (for instance, an outlying period such as a holiday), we split each trace into segments and analyse each independently. As the SASSY trace lasts over two months we split it into two 30-day segments and discard the rest. The nine-month Reality Mining trace is divided into three 30-day periods, from the beginning, middle and end of the trace respectively. As the NUS data are cyclical, we simulate using only one week of trace information. For the Hope trace we use the second day of three. We perform 30 runs of each protocol in each segment.

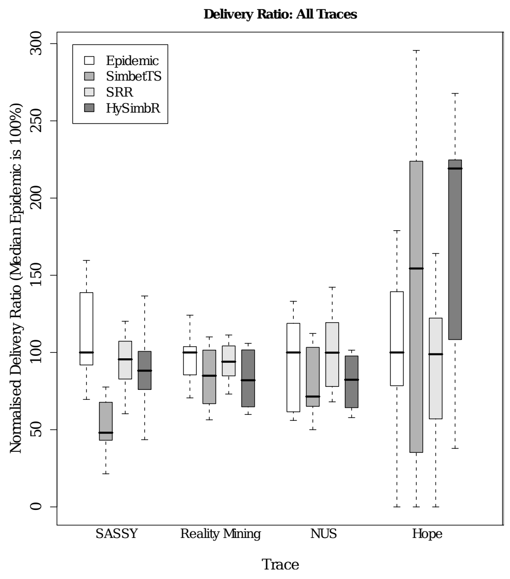
D. Messaging and Battery Model

Each node is assumed to be a smartphone, with message buffer and battery parameters as in Table II. We only consider encounters over 30 seconds in length, assuming that shorter encounters are insufficient for exchanging data. Message sources and destinations are chosen from a randomly-generated exponential distribution, to model real-world messaging patterns.

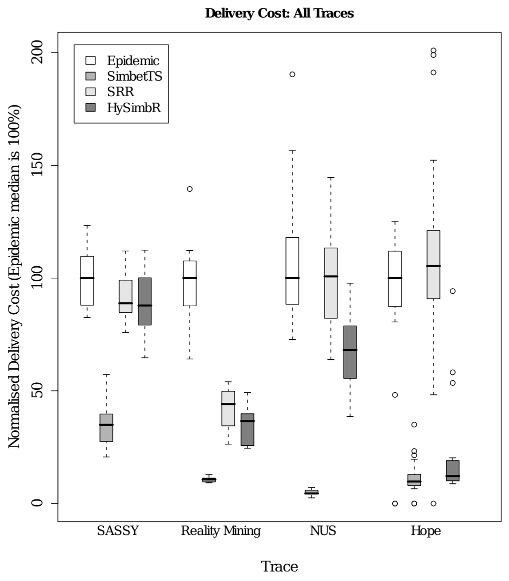
To determine an energy model for our nodes, we enabled the Bluetooth radio on a Nokia N95 smartphone with a 1200 mAh battery and measured energy depletion using the Nokia Energy Profiler v1.2. Our simulated nodes' batteries deplete according to this observed behaviour (1.9×10^{-6}mAh). When a node sends a message, it decrements its battery level by 0.4mAh. When a node runs out of battery it charges for 8 hours, and is not present on the network during this time.

E. Metrics

To compare protocol performance, we analyse two commonly-used metrics. *Delivery ratio* is the number of messages arriving at destinations divided by the total number of messages. *Delivery cost* is the average number of medium accesses required to deliver a message to a destination.



(a) No one protocol performs best in all traces. SRR and HySimbR perform as well as or better than SimbetTS in all traces.



(b) SimbetTS always has a low cost, due to its reduced message duplication. HySimbR performs quite well, whereas the high duplication protocols do not.

Fig. 3. Box plots of protocol delivery ratio and delivery cost (normalised using median Epidemic routing value as 100).

V. Results

We now present results of the trace-driven simulations performed to evaluate our routing protocol. For brevity we have only included graphs for the starting segments of each trace. All observed relative performance results remain the same in the other trace segments.

Figures 3(a) and 3(b) show that no one protocol performs the best in all traces. One of SRR and HySimbR always does as well as Epidemic, and HySimbR performs better in the Hope trace. Due to the large number of forwards it produces Epidemic frequently performs worse than the other protocols as it drains the battery and memory resources of nodes. SRR does well in all traces, but the density of the Hope trace increases the cost of routing, causing both Epidemic and SRR to perform worse than SimbetTS and HySimbR.

SimbetTS has a low cost in all traces. This is due to the small number of message duplications it produces. In the very dense Hope trace, SimbetTS and HySimbR perform better than SRR and Epidemic, and have a much lower cost. In this trace, many message duplications can be seen to negatively impact performance. Even here HySimbR performs slightly better than SimbetTS, showing that the extra duplications were useful in delivering messages, without a negative impact from the increased cost. Therefore a higher cost does not necessarily lead to worse performance, as the other protocols frequently perform better than SimbetTS.

Overall, SRR performs similarly or better than SimbetTS in most traces, and HySimbR does as well as SimbetTS or better in all traces. This shows that roles are useful for routing, but in particular at helping during network startup, where bootstrapping the network using roles provides an advantage over having to create an encounter history.

VI. Conclusions

We have presented a novel opportunistic routing protocol, Social Role Routing, that uses self-reported social networks and applies the social network analysis technique of role analysis to select nodes to act as message relays. This allows the protocol to route effectively when expected nodes are not present, and to provide good performance while encounter-history-based protocols are still warming up.

We find that SRR performs as well as existing protocols in most scenarios. Moreover, it always performs better than existing protocols when combined with SimbetTS to bootstrap an opportunistic network, with a delivery ratio similar to Epidemic routing, at a much lower cost.

We see that minimising delivery cost is not necessarily an indicator that a routing protocol will perform better than a protocol that does not. SRR performs well despite having a high cost, as it aims to avoid adversely overloading nodes that perform a large bulk of the forwarding. We intend to investigate metrics that capture routing protocol effects on individual nodes (rather than the network level), as we believe our results show understanding these effects is important for building effective opportunistic routing protocols.

We evaluated protocol performance using four traces each with differing structure and connectivity patterns. No one protocol consistently performs best across all of these traces. Which trace, if any, is indicative of an opportunistic network in the general population, is an open question. Even then, it appears that a system designer's choice of routing protocol may depend on the nature of the system's users.

Our plans for future work include analysing a network's connectivity and structural properties during the warm-up period to determine which routing protocol is the most appropriate to use after the warm-up period, investigating the affect of varying the minimum role size, and further comparison of role-graphs and self-reported social networks.

References

[1] K. Xu *et al.*, "Exploring centrality for message forwarding in opportunistic networks," in *IEEE WCNC*, Apr. 2010.

[2] L. Pelusi *et al.*, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, pp. 134–141, Nov. 2006.

[3] E. M. Daly *et al.*, "Social network analysis for information flow in disconnected delay-tolerant MANETs," *IEEE Transactions on Mobile Computing*, vol. 8, pp. 606–621, May 2009.

[4] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, pp. 35–41, Mar. 1977.

[5] P. Hui *et al.*, "Bubble rap: social-based forwarding in delay tolerant networks," in *MobiHoc*, May 2008, pp. 241–250.

[6] A. J. Mashhadi *et al.*, "Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks," in *WoWMoM '09*, Jun. 2009.

[7] A. Mtibaa *et al.*, "PeopleRank: Social opportunistic forwarding," in *IEEE INFOCOM*, Mar. 2010.

[8] A. K. Pietiläinen *et al.*, "MobiClique: middleware for mobile social networking," in *Proc. WOSN*, Aug. 2009, pp. 49–54.

[9] C. Boldrini *et al.*, "Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution," *Pervasive and Mobile Computing*, vol. 4, pp. 633–657, Oct. 2008.

[10] M. R. Pearlman *et al.*, "On the impact of alternate path routing for load balancing in mobile ad hoc networks," in *MobiHoc*, Aug. 2000.

[11] D. R. White *et al.*, "Graph and semigroup homomorphisms on networks of relations," *Social Networks*, vol. 5, Jun. 1983.

[12] H. C. White *et al.*, "Social structure from multiple networks. i. blockmodels of roles and positions," *The American Journal of Sociology*, vol. 81, pp. 730–780, Jan. 1976.

[13] S. Borgatti, "Two algorithms for computing regular equivalence," *Social Networks*, vol. 15, pp. 361–376, Dec. 1993.

[14] P. C. Kanellakis *et al.*, "CCS expressions, finite state processes, and three problems of equivalence," in *PODC '83*, Aug. 1983, pp. 228–240.

[15] G. Bigwood *et al.*, "Exploiting self-reported social networks for routing in ubiquitous computing environments," in *Proc. SAUCE*. IEEE, Oct. 2008, pp. 484–489.

[16] N. Eagle *et al.*, "Reality mining: sensing complex social systems," *Personal Ubiquitous Computing*, vol. 10, pp. 255–268, May 2006.

[17] V. Srinivasan *et al.*, "CRAWDAD data set nus/contact (v. 2006-08-01)," Downloaded from http://crawdad.org/nus/contact, Aug. 2006.

[18] C. Liu *et al.*, "Routing in a cyclic mobispace," in *MobiHoc '08*, May 2008, pp. 351–360.

[19] V. Srinivasan *et al.*, "Analysis and implications of student contact patterns derived from campus schedules," in *MobiCom*. ACM, 2006, pp. 86–97.

[20] Aestetix *et al.*, "CRAWDAD data set hope/amd (v. 2008-08-07)," Downloaded from http://crawdad.org/hope/amd, Aug. 2008.

[21] A. Vahdat *et al.*, "Epidemic routing for partially-connected ad hoc networks," CS-200006, Duke University, Tech. Rep., Apr. 2000.

[22] W. Gao *et al.*, "Multicasting in delay tolerant networks: a social network perspective," in *MobiHoc '09*, May 2009, pp. 299–308.