PeopleRank: Social Opportunistic Forwarding

Abderrahmen Mtibaa Martin May Christophe Diot
Thomson, Paris, France
firstname.lastname@thomson.net

Mostafa Ammar
Georgia Institute of Technology
ammar@cc.gatech.edu

***Abstract*—**
In opportunistic networks, end-to-end paths between two communicating nodes are rarely available. In such situations, the nodes might still copy and forward messages to nodes that are more likely to meet the destination. The question is which forwarding algorithm offers the best trade off between cost (number of message replicas) and rate of successful message delivery. We address this challenge by developing the *PeopleRank* approach in which nodes are ranked using a tunable weighted social information. Similar to the PageRank idea, *PeopleRank* gives higher weight to nodes if they are socially connected to other important nodes of the network. We develop centralized and distributed variants for the computation of *PeopleRank*. We present an evaluation using real mobility traces of nodes and their social interactions to show that *PeopleRank* manages to deliver messages with near optimal *success rate* (*i.e.,* close to Epidemic Routing) while reducing the number of message retransmissions by 50% compared to Epidemic Routing.

I. Introduction

For many years, communication in mobile scenarios was focusing on communication between mobile devices and fixed access points. With the emergence of a new generation of powerful mobile devices, novel communication paradigms were possible: ad hoc data transfer between the mobile devices. In such ad hoc network settings, the devices are often disconnected from each other and use Bluetooth, Wifi, or any other wireless connectivity to exchange and forward data in an opportunistic hop by hop manner.

In such opportunistic ad hoc networks [5], a device has to decide whether or not to forward data to an intermediate node that it encounters. Such forwarding decisions are typically guided by the desire to reduce the number of replicas of data items in the network to conserve bandwidth while reducing the end-to-end delay.

The more additional information about the network are available, the more likely the optimal forwarding decision can be made. Such additional information include historical contacts with other devices, information about the device mobility, or most important for this work, information about the social interaction between people carrying the devices.

Previous work studied how the availability of various types of information about the network can be used to guide and improve forwarding decisions. Such information includes historical contacts with other devices [3], [4], [1], [12], [10], information about device mobility patterns [16], or, most relevant to this work, information about the social interaction between the participants [7], [11], [14]. Those latter ones implicitly assume that the opportunistic contacts relate with the social property used to design the algorithms.

In this paper, we develop a systematic approach to the use of social interaction as a means to guide forwarding decisions in an opportunistic ad-hoc network. We introduce an opportunistic forwarding algorithm that uses *PeopleRank* which ranks the "importance" of a node in the social graph. *PeopleRank* is inspired by the PageRank algorithm [2] used in Google's search engine to measure the relative importance of a Web page within a set of pages. Analog to the PageRank, *PeopleRank* identifies the most popular nodes (in a social context) to forward the message to, given that popular nodes are more likely to meet other nodes in the Networks [14], [13], [11]. There is an important difference between the PageRank algorithm which imposes a centralized data collection and *PeopleRank* which also can be implemented in a centralized way but in a completely distributed fashion.

In the rest of this paper, we present our contributions but start with related work. In section II, we present the *PeopleRank* algorithm. Section III evaluates the performance of the *PeopleRank* algorithm; we compare the PeopleRank performance to other existing approaches. We show that PeopleRank outperforms all other forwarding rules studied. Then, we conclude the paper with a discussion of our results.

II. The PeopleRank Algorithm

Using social interaction between people for forwarding has already been proven to reduce the number of message replicas (*cost*) while increasing the likelihood of a message to reach its destination (*success rate*) [7], [14]. However, studying exhaustively the social interaction as a good predictor of future contact opportunities remains largely unexplored. In the following, we present and compare different methods that use additional information such as historical contact or social patterns to make "better" forwarding decisions.

A. The idea

In general, global knowledge of network topology can make for very efficient routing and forwarding decisions. Collecting and exchanging topology information in opportunistic networks is cumbersome because of their intermittent connectivity and unpredictable mobility. Routing schemes for such networks typically rely on partial knowledge and on prediction of future contacts which results in degraded routing performance.

With the emergence of Online Social Network platforms and applications such as Facebook, Orkut, or MySpace, information about the social interaction of users has became readily available. Moreover, while opportunistic contact information is changing constantly, the links and nodes in a social network remain rather stable.

The idea of this paper is to use this more stable social information to augment available partial contact information in order to provide efficient data routing in opportunistic networks. The intuition behind this idea is that socially well connected nodes are better suited to forward messages towards any given destination. More specifically, we propose a ranking algorithm that is inspired by more famous web page ranking. We apply it to rank nodes based on their position in a social graph. We then use this ranking as a guide for forwarding decisions. More specifically, a node u forwards data to a node v that it meets if the rank of v is higher than the rank of u.

PeopleRank is inspired by the PageRank [2] algorithm employed by Google to rank web pages. By crawling the entire web, this algorithm measures the relative importance of a page within a graph (web). Motivated by the success of this algorithm, we propose to apply a similar technique, which we call *PeopleRank* to rank the nodes in a social graph. The main idea is that nodes with a higher *PeopleRank* value will generally be more "central" in the social graph.

We present, next, the centralized and the distributed implementation of our *PeopleRank* algorithm.

B. *Centralized* PeopleRank

Before going into the details of our *PeopleRank* algorithm, we revisit the properties of the PageRank algorithm. PageRank performs a random walk on the World Wide Web graph, where the nodes are pages, and the edges are links between pages. It gives the probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank is given by the following equation:

$$PR(p_i) = \frac{1-d}{n} + d \sum_{p_j \in M(i)} \frac{PR(p_j)}{L(j)} \tag{1}$$

where $p_1, p_2, ..., p_n$ are the pages, $M(i)$ is the set of pages that link to p_i, $L(j)$ is the number of outbound links on page p_j, and d ($0 < d \leq 1$) is a *damping factor* which is defined as the probability, at any step, that the person will continue clicking on links instead of opening another random page.

We apply the same idea in *PeopleRank* to tag people as "important" when they are linked (in a social context) to many other "important" people. We assume that only neighbors in the social graph have an impact of the popularity (*i.e.,* the ranking).

In the same way web pages are hyperlinked, we establish a social graph between persons when they are socially related to each other. Such social relationships can be based on explicit friendships (as defined in online social networks for example), on personal communication (for example extracted through the communication patterns available in cell phones), or can be based on common interests. We denote such a social graph $G_s = (V_s, E_s)$ as a finite undirected graph with a vertex set V and an edge set E_s. An edge $(u, v) \in E_s$ if, and only if, there is a social relation between nodes u and v. In this paper, we define a social relationship between two nodes u and v either (i) if they are declared friends, or (ii) if they are sharing k common interests.

Consequently, the *PeopleRank* value is given by the following equation:

$$PeR(N_i) = (1-d) + d \sum_{N_j \in F(N_i)} \frac{PeR(N_j)}{|F(N_j)|} \tag{2}$$

where $N_1, N_2, ..., N_n$ are the nodes (devices), $F(N_i)$ is the set of neighbors that links to N_i, and d is the damping factor which is defined as the probability, at any encounter, that the social relation between the nodes helps to improve the rank of these nodes. This means that, the higher the value of d, the more the algorithm accounts for the social relation between the nodes. As a result, the damping factor is a very useful in controlling the weight given to the social relations for the forwarding decision. Such a mechanisms is very important since social graphs are built on different types of information. One could expect that a "friendship" between two individuals defines a stronger social relation than one defined by one or multiple common interests. When using *PeopleRank* for message forwarding, the damping factor should then be set to a value close to one for strong social relations and smaller for more loosely defined social graphs. In the next section, we address this issue in more detail and examine the impact of the damping factor on the *PeopleRank* performance.

The techniques we described thus far are suitable for centralized implementations where the social graph is known a-priori. Clearly this may not be feasible in the mobile wireless environment we are considering. We describe the distributed version of our algorithm in the next section.

C. *Distributed* PeopleRank

The distributed version of *PeopleRank* is shown in Algorithm 1. In this version, whenever two neighbor nodes in the social graph meet, they exchange two pieces of information: (i) their current *PeopleRank* values; and (ii) the number of social graph neighbors they have. Then, the two neighbors update their *PeopleRank* values using the formula given in Eq. 2. Implicitly, this distributed version of the algorithm exploits the mobility and contact behavior of the nodes since the *PeopleRank* value is updated every time the nodes meet.

Algorithm 1 Distributed *PeopleRank* (i)

```
Require: |F(i)| ≥ 0
  PeR(i) ← 0
  while 1 do
    while i is in contact with j do
      if j ∈ F(i) then
        send(PeR(i), |F(i)|)
        receive(PeR(j), |F(j)|)
        update(PeR(i))                    (Eq. 2)
      end if
      while ∃ m ∈ buffer(i) do
        if PeR(j) ≥ PeR(i) OR j = destination(m) then
          Forward(m, j)
        end if
      end while
    end while
  end while
```

Note that in the distributed algorithm frequently seen nodes update their *PeopleRank* more often. In fact, the more often two nodes meet, the faster their rank increases. This will tend to "inflate" the social ranking (*PeopleRank*) for frequently seen nodes.

III. PEOPLERANK EVALUATION

The performance of a forwarding algorithm like *PeopleRank* is determined by two conflicting factors: (i) the average message delivery delay; and (ii) the overhead (or cost) induced by the forwarding mechanism, *i.e.,* the number of message replicas in the system. In the following, we assess the *PeopleRank* performance with regard to these two performance indicators.

We have chosen to evaluate our forwarding algorithm using analysis on real traces. Specifically, we used the following experimental datasets: MobiClique, SecondLife, Infocom06, and Hope. Each dataset includes both, a mobility or contact trace and a social interaction graph. Table I summarizes the characteristics of these datasets.

A. Methodology

The trace collection is one part of the challenge. It is also difficult to prepare the data set for later analysis. Key here is to establish a benchmark, *i.e.,* a baseline performance used to compare the different methods. In our case, we have to determine the "optimal" forwarding paths given the mobility patterns and the connectivity properties. We establish this benchmark as follows: We compute offline, for all starting times (the time when the message is ready to be sent) and all source-destination pairs, the set of delay-optimal paths. Paths may use either a single contact, or a sequence of contacts in a time-respecting manner. In other words, we compute the optimal delivery time of a message sent by a node A to a node B ($A \neq B$) for all possible starting times. Then, we compute the success rate as the delivery probability with a given end-to-end delay bound following the same construction rules as described in [6].

In a first step, we study the impact of the damping factor and the social patterns on the *PeopleRank* performance. Then, we compare the *PeopleRank* performance (*success rate* and *cost*) to the most well-known social- and contact-based forwarding algorithms using the construction rules described above.

B. Impact of damping factor

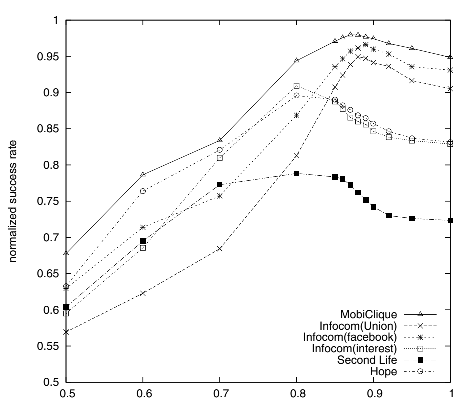


Fig. 1. Impact of damping factors

As described in section II, d can be used to control the amount of randomness in *PeopleRank*. For d equal to 0, all nodes are used for forwarding with the same probability, values close to 1 will prefer the socially best connected nodes. Fig. 1 plots a normalized *success rate* of *PeopleRank* for a TTL of ten minutes with different damping factors (*i.e.,* we measure the *success rate* of *PeopleRank* normalized by the *success rate* of flooding within a delay of ten minutes). It can be seen from this plot that optimal damping factor values change with the different traces (and the underlying social information). In fact, three traces (MobiClique, Infocom(facebook), and Infocom(union)) show the best performance with an optimal damping factor around 0.87 while the others traces (Hope, SecondLife, and Infocom(interest)) perform best for $d \approx 0.8$. The first three traces are based on explicitly defined connections, the three latter traces are built on implicit social connections. We conjecture that the reason for this difference lies in the way the social interactions are defined in each dataset. Implicit connections are defined as contacts between persons that share some common interests; we define a connection as explicit only when the two nodes declared a direct connection (for example links in applications like Facebook). Obviously, in traces with an explicit social pattern, the social graph information is more likely to be suitable for forwarding, and hence, damping factors of close to 1 perform best.

Fig. 1 also shows that the curves decrease for a damping factor close to 1. This effect can be explained by the fact that a damping factor d equal to 1 considers exclusively socially connected nodes. Since also socially disconnected nodes are potentially able to deliver messages, the performance decreases when those nodes are not considered in the forwarding path. That means that even in networks that exhibit a high correlation between the social and contact graphs, some randomized forwarding is beneficial.

C. Evaluation Results for PeopleRank

C.1 Comparison to social algorithms

We selected the following two well-known social forwarding algorithms for comparison with *PeopleRank*:

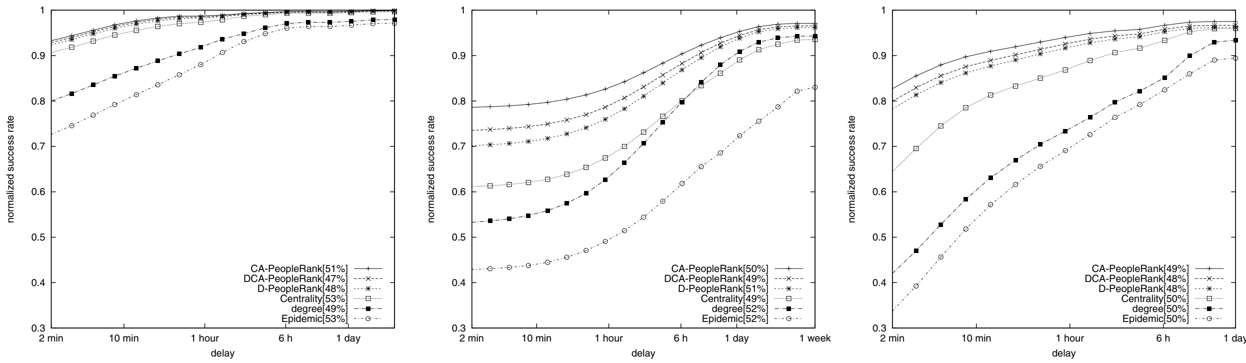
- centrality: u forwards a message to v if, and only if, $C(u) \leq C(v)$. Where $C(u)$ denotes the betweenness centrality of node u measured as the occurrence of this node in all shortest paths connecting all other pairs of nodes.
- degree: u forwards a message to v if, and only if, $d(u) \leq d(v)$. $d(u)$ denotes the degree of node u in the social graph (in a friendship graph, the degree is the number of friends of node u).

In Fig. 2, we plot the *PeopleRank* delay and cost performances for the six datasets presented in Table I. We compare the delay distribution (CDF) of *PeopleRank* , centrality, and the degree-based algorithm with an epidemic approach performing the same number of retransmission (cost). The plot shows that our *PeopleRank* algorithm outperforms all the other forwarding schemes on all six datasets. *PeopleRank* and the centrality-based algorithm perform with around 90% of the maximum success rate within the 10 minutes timescale using the three datasets (*II*) MobiClique, Infocom(facebook) and Infocom(union). In theses data sets, *PeopleRank* achieves the best success rates while using only 50% of contacts. Note again that the centrality-based algorithm required centralized computation (as defined in [14]) and is way more complex to compute than the other rules under consideration.

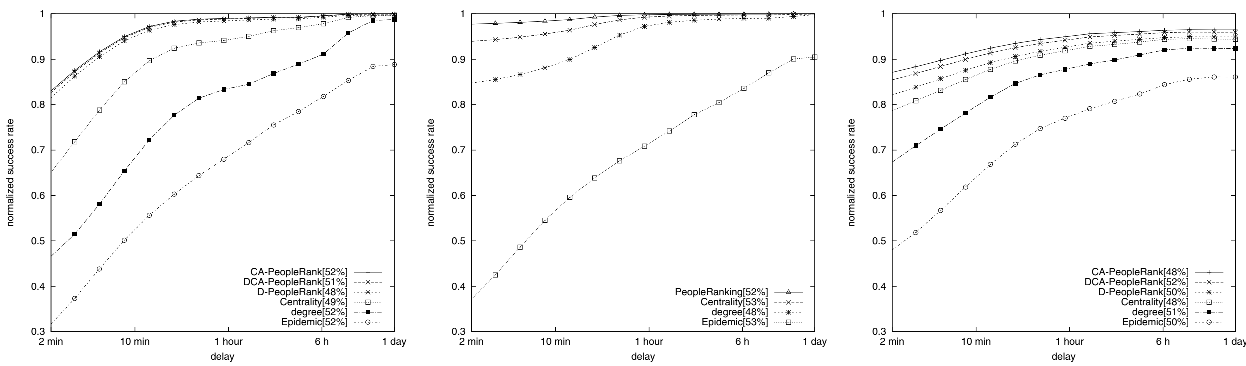
In the SecondLife dataset (see Fig. 2(b)), the non-decreasing-degree rule outperforms the non-decreasing-centrality rule for larger timescales (up to six hours). Indeed, when a message spend long time in the network before

	MobiClique	SecondLife	Infocom(interest)	Infocom(facebook)	Infocom(union)	Hope
duration	3d	10d	3d	3d	3d	3d
mobility patterns	bluetooth contacts	cartesian coord.	bluetooth contacts	bluetooth contacts	bluetooth contacts	cartesian coord.
social patterns	*explicit*	*implicit*	*implicit*	*explicit*	*explicit*	*implicit*
# connected nodes	27	150	65	47	62	414
# edges	115	2452	835	219	423	6541
median inter-contact	10mn	25mn	15mn	15mn	15mn	30mn
median contact time	240s	180s	150s	150s	150s	90s

TABLE I
DATASET PROPERTIES



(a) MobiClique (b) SecondLife (c) Infocom(Interest)



(d) Infocom(Facebook) (e) Infocom(Union) (f) Hope

Fig. 2. *PeopleRank* vs. social algorithms

reaching its destination, it is better to forward it to nodes that are socially well connected than nodes that are central in the social graph. SecondLife consists of a virtual environment, where avatars interact with each other. However, Varvello et al. [17] observed that above 30% of regions do not attract any visitors, and in few popular regions some avatars spend more than 12 hours. Those people frequently meet their friends and join many groups of interest to interact with other people. As result, those avatars become more and more connected in the social graph (large degree) and are likely to meet additional people and discuss with them.

In Fig. 2, we plot the *PeopleRank* performance using the same contact trace between the preselected Infocom participants and three different social graphs connecting those participants (their Facebook profiles, common interests, and other social information). In the Infocom06 experiment, users were asked to fill their interest as a mandatory field in the questionnaire, additional fields such as name, nationality, affiliation, and city were optional. Thus, the Infocom(Facebook) and the Infocom(Union) trace use a subset of the nodes in the global Infocom(interest) dataset (see Tab. I for more details). In Fig. 2(c), (d), and (e), we compare how the social properties used by the forwarding rules reduce the number of message retransmissions. Using the friendship graph extracted from Facebook, the *PeopleRank* algorithm performs with a success rate above 95% normalized by the optimal flooding delay. It improves the performance by 45% of the random distribution using the same number of retransmission. Indeed, with Facebook we are able to build an explicitly defined social graph; a link is established if, and only if, the two users accept this friend relationship. Similar results were shown using the union social graph which is based on geographic location as defined by academic affiliation or cities. Geographic location help user to socialize more often and meet with each other more frequently. The social graph based on common interests (Fig. 2(c)) use a common topics to link users. However, this information is not efficient to identify people that are likely to meet and socialize with each other. In Fig. 2(c), the *PeopleRank* algorithm performs with a 86% success rate while the degree-based algorithm performs with only 58% within a 10 minutes timescale. We conclude that people sharing the same interest may not good targets for opportunistic social message forwarding: shared interest is not a good estimation to model a social network.

C.2 Comparison to contact-based algorithms

In order to compare the performance of our algorithm to a non-social based algorithm, we implemented four selected forwarding algorithms. All these algorithms are well-known in the literature, they use a local information to decide whether to forward a message when node i meets node j.

- **Last Contact** [8]: Node i forwards messages to node j if j has contacted any other node more recently than node i.
- **Destination Last contact** [8]: Node i forwards messages to node j if j has contacted the message's destination more recently than has node i.
- **Frequency** [9]: Node i forwards the message to node j if j has more total contacts than node i.
- **spray&wait** [15]: the source node creates R (we use $R = 8$) replicas of the message. If node i has $k > 1$ replicas of the message and j has no replicas, i forwards $k/2$ replicas to j. Otherwise ($k = 1$) i wait the destination.
- **Wait_destination**: the message is forwarded only if the source node met directly its destination. Obviously, this algorithm consists of using the minimum cost which causes higher delay and lower success delivery message.

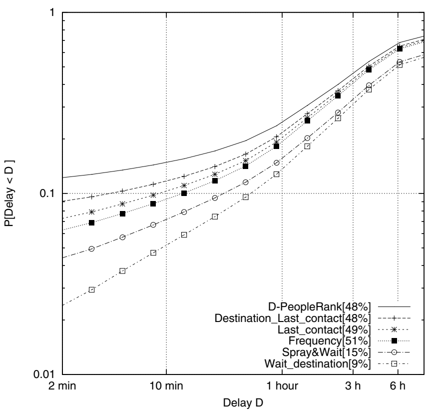


Fig. 3. Social versus contact based forwarding

Figure 3 plots the shortest-delay distribution and the cost of PeopleRank for the four selected forwarding schemes. PeopleRank outperforms the Frequency, Last contact and destination Last contact algorithms. In fact, it uses almost the same number of message transmission (around 50% of contacts) more efficiently (10% to 15% additional success delivery within 1 hour timescale) than the three others algorithms; i.e., the social aspect of the algorithm delivers the messages with higher probability to the destination. The Spray&Wait algorithm reduces dramatically the number of retransmissions which impacts in return the success rate of the algorithm. Note that Spray&Wait does not use any utility function to select forwarding nodes, it just send the message to the first node encountered. Indeed, the success delivery ratio with Spray&Wait is only 2% higher than the lower bound, i.e., Wait_destination.

IV. Conclusion

The emergence of Online Social Networks applications on the new powerful mobile devices makes social information between people easily accessible. In this paper, we have presented *PeopleRank* , a forwarding algorithm exploiting social properties to reduce the number of message retransmissions in Mobile Opportunistic Networks. *PeopleRank* is a social distributed algorithm measuring opportunistically the importance of a node in a social graph based on the social interaction between nodes and their contact frequency. Our results which are derived from three real human mobility and one virtual human mobility (SecondLife) trace, highlight the better performances given by the *PeopleRank* algorithm; it achieves an end-to-end delay and a success rate close to those given by flooding while reducing the number of retransmission by 50%.

References

[1] A. Balasubramanian, B. Levine, and A. Venkataramani. Dtn routing as a resource allocation problem. *SIGCOMM Comput. Commun. Rev.*, 37(4):373–384, 2007.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107–117, 1998.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networking. In *Proceedings of IEEE Infocom*, 2006.

[4] B. Burns, O. Brock, and B. Levine. MV routing and capacity building in disruption tolerant networks. 2005.

[5] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket switched networks: Real-world mobility and its consequences for opportunistic forwarding. Technical Report UCAM-CL-TR-617, University of Cambridge, Computer Lab, Feb 2005.

[6] A. Chaintreau, A. Mtibaa, L. Massoulié, and C. Diot. The diameter of opportunistic mobile networks. In *Proceedings of ACM CoNext*, 2007.

[7] E. M. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *MobiHoc '07: Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40, New York, NY, USA, 2007. ACM.

[8] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 257–266, 2003.

[9] V. Erramilli, A. Chaintreau, M. Crovella, and C. Diot. Diversity of forwarding paths in pocket switched networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 161–174, New York, NY, USA, 2007. ACM.

[10] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 251–260, New York, NY, USA, 2008. ACM.

[11] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: Social-based forwarding in delay tolerant networks. In *to appear in Proceedings of ACM MobiHoc '08*, New York, NY, USA, 2008. ACM. (also appeared as Cambridge University TR: UCAM-CL-TR-684).

[12] A. Lindgren, A. Doria, and O. Schelén. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

[13] A. Mtibaa, A. Chaintreau, and C. Diot. Popularity of nodes in pocket-switched networks. In *Proceedings of ACM SIGCOMM (Extended abstract)*, 2007.

[14] A. Mtibaa, A. Chaintreau, J. LeBrun, E. Oliver, A.-K. Pietilainen, and C. Diot. Are you moved by your social network application? In *WOSN'08: Proceedings of the first workshop on Online social networks*, pages 67–72, New York, NY, USA, 2008. ACM.

[15] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, New York, NY, USA, 2005. ACM.

[16] M. M. B. Tariq, M. Ammar, and E. Zegura. Message ferry route design for sparse ad hoc networks with mobile nodes. In *MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing*, pages 37–48, New York, NY, USA, 2006. ACM Press.

[17] M. Varvello, E. Biersack, and C. Diot. Is there life in second life? In *Proceedings of ACM CoNext*. ACM, 2007.