# SPRINT: Social Prediction-Based Opportunistic Routing

Radu Ioan Ciobanu, Ciprian Dobre, Valentin Cristea
*Faculty of Automatic Control and Computers*
*University Politehnica of Bucharest, Romania*
*radu.ciobanu@cti.pub.ro, {ciprian.dobre, valentin.cristea}@cs.pub.ro*

## Abstract

*Opportunistic networks are mobile networks that rely on the store-carry-and-forward paradigm, using contacts between nodes to opportunistically transfer data. For this reason, traditional routing mechanisms are no longer suitable. To increase the probability of successfull message delivery, we propose SPRINT, an opportunistic routing algorithm that introduces an additional routing criterion: online social information about nodes. Furthermore, previous results show that, for particular environments, contacts between devices in opportunistic networks are highly predictable. When users follow rare events-based mobility patterns, we show that human mobility can be approximated as a Poisson distribution. Based on this result, we add an additional prediction component into our routing algorithm. Our solution delivers better results compared to traditional social-based routing approaches, for different real-world and synthetic mobility scenarios.*

## 1. Introduction

The emergence and wide spread of new-generation mobile devices create the premises for new means of communication and interaction using alternative, ad-hoc networks such as opportunistic networks (ONs). An ON [1] is a type of delay-tolerant network (DTN) established in environments where human-carried mobile devices act as network nodes and are able to exchange data while in proximity. Whenever a destination is not directly accessible, a source opportunistically forwards data to its neighbours (nodes in wireless range). The latter act as carriers and relay the data until the destination is reached or the messages expire. Routing algorithms for ONs are either mobility-aware or social-aware. The former (which includes protocols such as PRoPHET [2]) takes routing decisions based on the number and duration of node encounters; the latter (which includes BUBBLE Rap [3]) relies on the knowledge that the nodes of an ON are people carrying mobile devices. Since it is generally believed that individuals of a certain type are more likely to interact with members of the same type, one can exploit the social relationships between participants in the ON in order to develop effective routing methods, which are able to bring messages close to a destination with high probability.

Traditionally, social-aware ONs rely on the history of contacts to statically determine social relationships between nodes. Here we extend this and propose an ON algorithm that introduces an additional social routing criterion: online social information about nodes. We note that nowadays it is feasible to assume that such information is available when forwarding, due to the integration of social networks such as Facebook, Twitter of LinkedIn in various mobile applications. Moreover, previous results show that, in certain types of environments, contacts between opportunistic devices are highly predictable. When users follow rare events-based mobility patterns, we show that human mobility can be approximated as a Poisson distribution. Based on this result, we add an additional prediction component into the proposed routing algorithm. We show that our approach produces better results when compared to traditional social-based routing approaches, for both real-world traces, as well as for synthetic mobility scenarios.

The contribution of this paper is twofold. Firstly, we introduce a novel ON routing algorithm entitled SPRINT (Social PRedIction-based routing in opportunistic NeTworks) that takes advantage all these aspects, while enhancing their functionality. It not only uses social information about the ON participants learned from the history of contacts, but also from social networks. In addition, it includes the possible

978-1-4673-5828-6/13/$31.00 

Poisson-based prediction of a node's future behaviour in the routing decision. Secondly, we demonstrate through extensive experiments, using both real-world mobility traces and synthetic models, that SPRINT optimizes previously proposed social-aware opportunistic routing solutions. It is capable, in fact, to improve hit rate, latency, delivery cost and hop count.

## 2. Related Work

A detailed review of opportunistic networking is performed in [4]. Relevant forwarding algorithms, such as BUBBLE Rap [3] or PROPICMAN [5], are analyzed. Other popular ONs routing algorithms include dLife [6], PRoPHET [2], or CiPRO [7]. We compare our solutions to BUBBLE Rap, one of the most well-known and effective socially-aware opportunistic routing techniques. We don't focus on comparisons with history-based algorithms such as PRoPHET in this paper because we are also dealing with less predictable environments, but we will attempt to do so in future work. BUBBLE Rap uses social knowledge about the nodes in the network to deliver messages. As such, it forwards data to nodes that are more popular than the carrier node, bubbling it up a hierarchical ranking tree using a global popularity level. A node's popularity is the value of its betweenness centrality, i.e. the number of times the node is on the shortest path between any other two nodes in the network. Communities in BUBBLE Rap are dynamically discovered using $k$-CLIQUE [8]. Another socially-aware middleware that learns information about the nodes in the network and then uses it to predict their future movement is proposed in [9].

The prediction of the future behaviour of nodes in DTNs is treated in several papers. In [10], the behaviour of a time series is modelled as a Poisson distribution, which in turn is modulated using a hidden Markov process. The authors show that using a Poisson model is significantly more accurate at detecting future behaviour and known events than a traditional threshold-based technique. Since contact information in an ON is also a time series, we believe that it can also be approximated as a Poisson distribution for certain situations (like an academic environment). These results are based on the assumptions that human behaviour and mobility are predictable, an aspect previously demonstrated in [11] and [12]. The type of distribution that can approximate this behaviour may vary depending on the environment and the situation, therefore our proposed algorithm has a modular prediction component (as shown in Sect. 4). This means that it can be replaced with any prediction method, depending on the current ON characteristics.

For testing the results of our implementation, we use five publically-available mobility traces. UPB 2011 [13] and UPB 2012 [14] are two traces taken in an academic environment at the University Politehnica of Bucharest, where the participants were students and teachers at the faculty. UPB 2011 includes Bluetooth data collected for a period of 25 days by 22 participants, while UPB 2012 contains Bluetooth and WiFi data and had a duration of 64 days, with 66 participants. St. Andrews [15] is a real-world mobility trace taken on the premises of the University of St. Andrews and around the surrounding town. It lasted for 79 days and involved 27 participants that used T-mote Invent devices with Bluetooth capabilities. The main advantage of these three traces is that they also include social information about participating nodes, gathered from social network information (e.g. Facebook) about the users in the experiments, in the form of a graph of social connections. As shown in Sect. 4, we use this information in the routing process. Moreover, these three traces were all taken in an academic environment, which is a scenario where we have a high degree of predictability (as demonstrated in [16]). The Content trace [17] contains Bluetooth sightings recorded in various locations around the city of Cambridge that were likely to be visited by many people. Finally, Infocom 2006 [17] was collected during the IEEE Infocom scientific conference in Barcelona, Spain. Its participants were 78 students and researchers attending the student workshop, as well as 20 stationary Bluetooth-capable iMotes. We also tested our algorithm using HCMM [18], a synthetic mobility model that tries to replicate the behaviour of nodes in a DTN. It assumes that nodes are driven not only by the social relationships between them, but also by the attractions of physical locations. HCMM is based on the caveman model and assumes that each node is attracted to its home cell according to the social attraction exerted on that node by all nodes that are part of its community.

## 3. Contact Prediction

In a previous paper [16], we analyzed the predictability of nodes' mobility considering contact distribution of nodes over time under several real-world data traces, which contain data collected in academic environments, where we expected to see evidence that users tend to follow mobility patterns (as previously

shown in [12]). We showed that, by using contact history for these three traces, we are able to predict how many contacts would a node have in a given one-hour interval, by using a Poisson distribution. Although these results are mainly applicable to environments where there are regular contacts at given times, our algorithm still behaves well for other types of environments because it also includes other components that adapt the routing decision using extra social and message-state information. Additionally, the contact prediction mechanism is implemented as a SPRINT module, which can easily be replaced with prediction modules suitable for different environments.

## 4. The SPRINT Algorithm

Each SPRINT node has a data memory and a cache memory. The former is used to store data objects (messages), while the cache memory contains the node's previous encounters with other ON participants. When two nodes meet, they exchange information about each message in their data memory, which includes a hash of the message's content (its unique ID), the source and destination, the generation time, and the number of hops traversed so far. Generally, we assume that the ON nodes are aware of the IDs of the members of their communities in advance. If this is not the case, a distributed community detection algorithm such as $k$-CLIQUE [8] is used. Based on this information, each node computes utility values for the messages in its own memory, as well as for the ones advertised by the encountered node. The formula used by a SPRINT node $A$ to compute the utility of a message $M$ is:

$$u(M, A) = w_1 * U_1(M, A) + w_2 * U_2(M, A)$$

Messages from both nodes are sorted according to their utility values. If some of the messages with high utility values belong to the encountered node, a download request is sent for each of them. The node then starts transferring these messages in utility order, until it has finished downloading all the required messages or the two encountering nodes are not in range anymore. In the formula, $w_1$ and $w_2$ are weight values which follow the conditions that $w_1 + w_2 = 1$ and $w_1 > w_2$. $U_1$ and $U_2$ are utility components computed according to the following formulas:

$$U_1(M, A) = freshness(M) + p(M, A) * (1 - \frac{enc(M, A)}{24})$$

$$U_2(M, A) = c_e(M, A) * \frac{s_n(M) + hop(M) + pop(A) + t(M, A)}{4}$$

The $freshness(M)$ component of $U_1$ favours new messages. The value is set to 0.5 if the message has been created less than a day ago, and to 0 otherwise. This means that in the beginning, when new messages are created, they are spread to several nodes (the utility value is high). As messages travel along the network, they are forwarded only to other nodes that maximize the changes of succesful delivery. $p(M, A)$ is the probability of node $A$ being able to deliver a message $M$ closer to its destination. The term is based on predicting a node's behaviour, combined with the idea that a node has a higher chance of interacting with nodes it is socially connected with and/or has encountered before. It is an environment-specific function that can be configured according to the contact distribution of the network SPRINT is used in. For the academic environments presented in Sect. 2, we compute it based on the knowledge that the node contacts follow a Poisson distribution [16]. We start by analyzing the cache memory and counting how many times node $A$ encountered each of the other nodes. If a node has been previously met in the same day of the week or in the same two-hour interval as the current time, the total encounters value is increased by 1. For the nodes encountered in the past that are in the same social community as node $A$, the total number of contacts is doubled. The reason we add 1 to the total number of encounters for nodes that have been met in the same day of the week or in the same two-hour interval is that there is a certain regularity in the behaviour of nodes in an academic environment. Therefore, there should be a higher probability of encountering nodes that have been seen in the same intervals. There is a similar reason for doubling the total number of contacts when the past nodes are in the same community as node $A$, since a social opportunistic node tends to interact more with other members of its own social community. We then compute the probabilities of encountering nodes based on past contacts by performing a ratio between the number of encounters per node and the total number of encounters. The next step consists of computing the number of encounters $N$ that node $A$ will have for each of the next 24 hours by using the Poisson distribution probabilities [16] and choosing the value with the highest probability as $N$. We then pick the first $N$ nodes as potential future contacts for each of the next 24 hours (sorted by probability), and for the rest of them $p(M, A)$ is set to 0. $U_1$ also uses $enc(M, A)$, which is the time (in hours) until the destination of message $M$ will be met by $A$ according to the probabilities previously computed. If the destination will never be encountered, then $enc(M, A)$ is set to 24 (so the product is 0). We are

interested in forwarding the message to a node that will encounter a message's destination sooner rather than later. We multiply $p(M, A)$ by $1 - \frac{enc(M,A)}{24}$ because the sooner a good target for a message is met, the sooner the node can delete the message.

The second component of the utility function is $U_2$. $c_e(M, A)$ is set to 1 if node $A$ is in the same community as the destination of message $M$ or if it will encounter a node that has a social relationship with $M$, and 0 otherwise. The prediction information computed for $U_1$ is used to analyze the potential future encounters of a node. The $s_n(M)$ component is set to 1 (and 0 otherwise) if the source and destination of $M$ do not have a social connection, because if a message doesn't have the source and destination in the same community, the chance of it being delivered by the source is low (since it will mostly meet community nodes). Therefore, the messages should be given to a different node that has the chance of reaching the destination community. $hop(M)$ represents the normalized number of nodes that $M$ has visited, $pop(A)$ is the popularity value of $A$ according to its social network information (i.e. number of Facebook friends in the opportunistic network), and finally $t(M, A)$ is the total time spent by node $A$ in contact with $M$'s destination.

## 5. Results

We compare SPRINT to a distributed BUBBLE Rap implementation that employs $k$-CLIQUE for community detection and a C-window algorithm for centrality computation [3]. If a node $A$ delivers a message to node $B$, it still keeps a copy of this message in memory until its buffer is full and the message is replaced by a new one. Furthermore, we also compare SPRINT's hit rate to the Epidemic routing algorithm [19]. We use four metrics for testing SPRINT's performance. The first one is *hit rate*, defined as the ratio between successfully delivered messages and the total number of generated messages. Next, the *delivery latency* is defined as the time passed between the generation of a message and its delivery. Another metric we use is the *delivery cost*, defined as the ratio between the total number of messages exchanged and the number of generated messages, which shows the congestion of the network. We only count the actual data messages, and not the control messages sent when two nodes meet each other. The *hop count* is the number of nodes that carried a message until its destination on the shortest path, and should be low in order to avoid node congestion. Another metric related to congestion that we can measure is the amount of buffer overflow events, but due to space limitations, we refer you to [20], where we treated this problem.

We ran BUBBLE Rap, Epidemic and SPRINT on five traces, attempting to simulate an academic environment. Thus, every weekday, each node generates 30 messages with destinations chosen based on its social relationships using a per-node Zipf distribution with an exponent of 1. We chose Zipf because it has been shown that data requests and sends follow power law distributions [21]. Therefore, a node has a higher chance of sending a message to another member of its community than to other nodes. Where social information is not available (Content and Infocom 2006), the destinations are selected randomly. For every trace, we vary the size of the data memory from 20 to 4500, but we consider it unlimited for the Epidemic algorithm. The cache memory size is set empirically to 40. Each of the three algorithms was run five times on each trace, with varying random seed values, for a confidence level of 95%. As shown in Sect. 4, SPRINT uses information obtained from a node's social network when computing the utility values. However, Content and Infocom 2006 do not offer such information, so in this case we use the communities obtained by $k$-CLIQUE instead. These two traces also differ from the others because they do not represent an academic environment with a rare events-based mobility pattern, so the contact distribution can't be approximated as Poisson (thus, utility computation only relies on $U_2$).

Figure 1 shows that, for UPB 2011, SPRINT's hit rate is better than BUBBLE Rap's for most of the cases. The maximum improvement is of about 2.3%, which may not seem so important at first, but we can see that SPRINT can achieve maximum hit rate (Epidemic's hit rate), whereas BUBBLE Rap can't. SPRINT also outperforms BUBBLE Rap in terms of delivery cost and hop count. Distributed BUBBLE Rap has a limitation regarding hop count, since it computes node centrality and communities on the fly. Thus, situations can arise where messages get passed between the same nodes again and again. Centrality is computed using a C-window algorithm with 6-hour windows, so a node's centrality can differ between two such 6-hour intervals. This can lead to a message being delivered from a node $A$ to a node $B$ at one time interval, and then delivered back from $B$ to $A$ at a subsequent 6-hour period. Finally, SPRINT manages to improve latency by up to seven hours. Although ONs are DTNs, an improvement in response of seven hours is not negligible and is a step forward towards a real-life implementation of ONs, since users

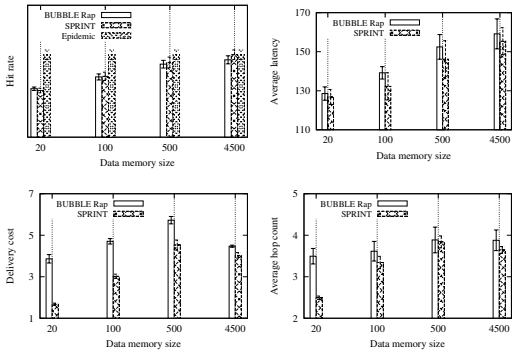


Figure 1. UPB 2011 trace.

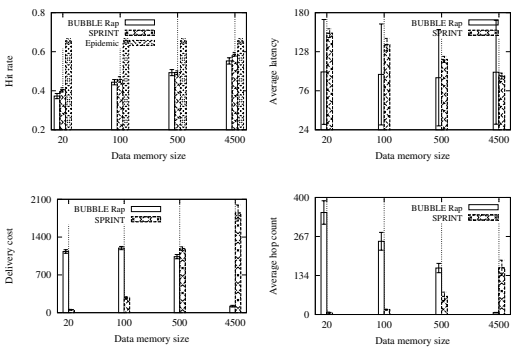


Figure 2. St. Andrews trace.

would probably not accept high values of latency so easily. The situation is similar for UPB 2012, as the traces were performed in similar conditions. The only differences were in the trace duration and the number of participants. Again, the hit rate values are better at SPRINT, which reaches the maximum hit rate for the given test scenario. The delivery cost and average hop count are drastically improved by SPRINT, except when the data memory is 4500. This happens because SPRINT manages to deliver more messages than BUBBLE Rap, which are destined for remote nodes that have few contacts with the other nodes. When messages are finally delivered to those nodes, the overall delivery cost and hop count are increased. The improvement in delivery latency is even better than for UPB 2011, having a maximum value of 41 hours. Another trace where SPRINT uses contact prediction when performing routing decisions is St. Andrews, shown in Fig. 2. This time, SPRINT doesn't manage to achieve maximum hit rate. This possibly happens because approximating the contact distribution as Poisson may not be so optimal. Nonetheless, SPRINT's hit rate is better than BUBBLE Rap's, irrespective of the data memory size. The delivery cost and average hop count situation is similar to the one observed at UPB 2012: for small data memory values, they are improved. However, because a larger data memory helps the algorithm distribute more messages to remote nodes, the overall delivery cost and hop count are increased. For large data memory values, SPRINT also improves the average latency.

Although Content and Infocom 2006 don't contain social information and $U_1$ is 0, SPRINT still behaves well, as it still uses social and context information to assign utility values to messages. Fig. 3 shows that, although SPRINT doesn't manage to achieve maximum hit rate for the Content trace, it still outperforms BUBBLE Rap in terms of hit rate by as much as 3%, because it doesn't forward messages only to nodes with higher centrality, which can easily get congested and have to start dropping old (but undelivered) messages. However, the important results for this trace concern delivery cost and hop count. SPRINT manages to decrease the delivery cost by as much as 146, so the total number of messages exchanged in the ON is five times lower when using SPRINT. The hop count is also reduced dramatically by SPRINT, by as much as 51. These improvements come courtesy of SPRINT taking into consideration the communities of the nodes encountered, as well as the age and hop count of a message, as opposed to BUBBLE Rap, which uses a C-window algorithm for centrality computation. As stated previously, a message can be exchanged between two nodes repeatedly, if the centralities differ between time windows. Most importantly, delivery latency is also decreased by as much as 8 hours when using SPRINT. The Infocom 2006 results look very much like Content: SPRINT outperforms BUBBLE Rap in terms of all four metrics, thanks to the $U_2$ component of the utility function, which uses social information about nodes, as well as message-related knowledge such as age or hop count.

For the HCMM test, we simulated an academic environment in order to have a predictable distribution of contacts. Therefore, we split the physical space into a 400x400-metre grid, with 10x10-metre cells, in order to simulate the campus of a university. The speed of the nodes was chosen between 1.25 and 1.5 metres per second (the average human speed), while the transmission radius of the nodes was 10 metres (the regular Bluetooth range). There were 33 nodes in the network, split into seven communities. The duration

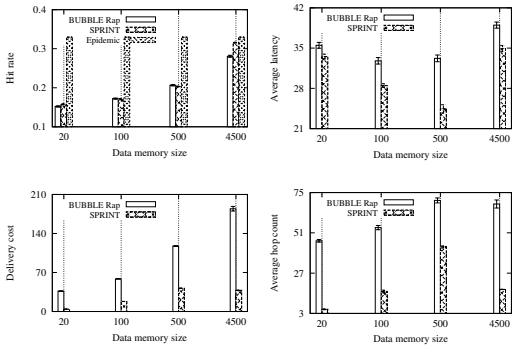


Figure 3. Content trace.

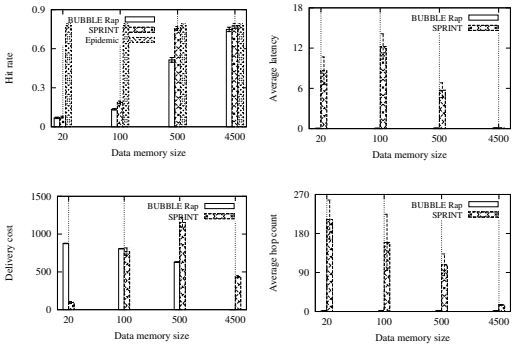


Figure 4. HCMM model.

of the scenario was three days, and the generation of messages was done in the same way as in the previous tests. We used the HCMM community grouping to create the social network used for routing decisions. The rewiring probability was 0.7, while the remaining probability was 0.8. We ran the same three algorithms (BUBBLE Rap, Epidemic and SPRINT) as for the trace scenarios, analyzing the results using the same four metrics (hit rate, delivery latency, delivery cost and hop count). The experimental results are shown in Fig. 4. For small data memory sizes, the hit rates are small, because there are a lot more messages in the network than a node is capable of carrying, and the duration of the trace is not so high as to allow a recirculation of all the messages. For every data memory size, SPRINT outperforms BUBBLE Rap in terms of hit rate, with an increase of up to 24%. Even for a data memory of 500 messages, SPRINT gets close to the best attainable hit rate. Both delivery cost and hop count are higher for SPRINT, but this happens for the same reasons stated before: more messages are delivered, even to remote nodes. Since contacts with these nodes are rare, the messages are delivered late. The same argument is also valid for latency.

## 6. Conclusions

We have presented SPRINT, a routing algorithm for opportunistic networks that uses information about a node's social connections, contacts history, and predictions of future encounters, when performing routing decisions. We have shown that it outperforms existing algorithms for various mobility traces, as well as in a synthetic simulation environment. Our approach manages to improve such metrics as hit rate, delivery latency, delivery cost, or hop count. These results have also proven that the distribution of contacts in certain scenarios (when users follow rare events-based mobility patterns) is highly predictable and can be approximated as a Poisson distribution.

## Acknowledgment

This work was partially supported by the project "ERRIC - Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1", ID: 264207. The work has been cofounded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

## References

[1] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Comm. Magazine*, vol. 44, no. 11, pp. 134–141, Nov. 2006. [Online]. Available: http://dx.doi.org/10.1109/MCOM.2006.248176

[2] A. Lindgren, A. Doria, and O. Schelén, "Probabilistic routing in intermittently connected networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 3, pp. 19–20, Jul. 2003. [Online]. Available: http://doi.acm.org/10.1145/961268.961272

[3] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: social-based forwarding in delay tolerant networks," in *Proc. of the 9th ACM int. symp. on Mobile ad hoc networking and computing*, ser. MobiHoc '08. New York, USA:

ACM, 2008, pp. 241–250. [Online]. Available: http://doi.acm.org/10.1145/1374618.1374652

[4] M. Conti, S. Giordano, M. May, and A. Passarella, "From opportunistic networks to opportunistic computing," *Comm. Mag.*, vol. 48, pp. 126–139, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1866991.1867009

[5] H. A. Nguyen, S. Giordano, and A. Puiatti, "Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network (PROPICMAN)," *2007 IEEE Int. Symp. on a World of Wireless Mobile and Multimedia Networks*, pp. 1–6, 2007.

[6] W. Moreira, M. de Souza, P. Mendes, and S. Sargento, "Study on the effect of network dynamics on opportunistic routing," in *Proceedings of the 11th international conference on Ad-hoc, Mobile, and Wireless Networks*, ser. ADHOC-NOW'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 98–111. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31638-8_8

[7] H. A. Nguyen and S. Giordano, "Context information prediction for social-based routing in opportunistic networks," *Ad Hoc Networks*, vol. 10, no. 8, pp. 1557–1569, Nov. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2011.05.007

[8] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proc. of 2nd ACM/IEEE inter. workshop on Mobility in the evolving internet architecture*, ser. MobiArch '07. New York, NY, USA: ACM, 2007, pp. 7:1–7:8. [Online]. Available: http://doi.acm.org/10.1145/1366919.1366929

[9] C. Boldrini, M. Conti, F. Delmastro, and A. Passarella, "Context- and social-aware middleware for opportunistic networks," *J. Netw. Comput. Appl.*, vol. 33, no. 5, pp. 525–541, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2010.03.017

[10] A. Ihler, J. Hutchins, and P. Smyth, "Learning to detect events with markov-modulated poisson processes," *ACM Trans. Knowl. Discov. Data*, vol. 1, no. 3, Dec. 2007. [Online]. Available: http://doi.acm.org/10.1145/1297332.1297337

[11] P. Hui and J. Crowcroft, "Predictability of human mobility and its impact on forwarding," in *Int. Conference on Communications and Networking in China*, 2008.

[12] C. Song, Y. Qu, N. Blumm, and A.-L. Barabási, "Limits of Predictability in Human Mobility," *Science*, vol. 327, pp. 1018–1021, 2010.

[13] R. I. Ciobanu, C. Dobre, and V. Cristea, "Social aspects to support opportunistic networks in an academic environment," in *Proceedings of the 11th international conference on Ad-hoc, Mobile, and Wireless Networks*, ser. ADHOC-NOW'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 69–82. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31638-8_6

[14] R.-C. Marin, C. Dobre, and F. Xhafa, "Exploring Predictability in Mobile Interaction," in *Emerging Intelligent Data and Web Technologies (EIDWT), 2012 Third International Conference on*. IEEE, 2012, pp. 133–139. [Online]. Available: http://dx.doi.org/10.1109/EIDWT.2012.29

[15] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti, "Exploiting self-reported social networks for routing in ubiquitous computing environments," in *Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 484–489. [Online]. Available: http://dl.acm.org/citation.cfm?id=1475703.1476536

[16] R.-I. Ciobanu and C. Dobre, "Predicting encounters in opportunistic networks." in *HP-MOSys*, C. X. Mavromoustakis, L. Shu, and T. Dagiuklas, Eds. ACM, 2012, pp. 9–14.

[17] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set Cambridge/Haggle (v. 2009-05-29)," Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009.

[18] C. Boldrini and A. Passarella, "HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships," *Comput. Commun.*, vol. 33, pp. 1056–1074, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2010.01.013

[19] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," 2000.

[20] R. I. Ciobanu, C. Dobre, and V. Cristea, "Reducing Congestion for Routing Algorithms in Opportunistic Networks with Socially-Aware Node Behavior Prediction," Mar. 2013, accepted at the 27th IEEE International Conference on Advanced Information Networking and Applications (AINA-2013).

[21] L. A. Adamic and B. A. Huberman, "Power-law distribution of the world wide web," *Science*, vol. 287, no. 5461, p. 2115, Mar. 2000. [Online]. Available: http://dx.doi.org/10.1126/science.287.5461.2115a