Social Aspects for Opportunistic Communication

Radu Ioan Ciobanu*, Ciprian Dobre*, Valentin Cristea*, Dhiya Al-Jumeily†

*University Politehnica of Bucharest
Faculty of Automatic Control and Computers
Bucharest, Romania
E-mails: radu.ciobanu@cti.pub.ro, ciprian.dobre@cs.pub.ro, valentin.cristea@cs.pub.ro

†School of Computing and Mathematical Sciences
Liverpool John Moores University
Liverpool , UK
E-mail: D.Aljumeily@ljmu.ac.uk

***Abstract*—As wireless and 3G networks become more crowded, users with mobile devices experience difficulties in accessing the network. Opportunistic networks, created between mobile phones using local peer-to-peer connections, have the potential to solve such problems by dispersing some of the traffic to neighbouring smartphones. Recently various opportunistic routing and dissemination algorithms were proposed and evaluated in various scenarios emulating real-world phenomena as close as possible. Such algorithms generally rely on mobility patterns of users and the context of communication. In this we investigate the addition of social data to improve the performance of communication algorithms and data transmission schema. When the routing decision is influenced by the chance of a particular user being able to successfully carry the data to the next hop, we believe that opportunistic communication algorithms could greatly benefit not only from learning the behaviour of users, but also their history of contacts coupled with the online social familiarity patterns between them. We believe users tend to be in contact more with familiar sets of users, with whom they share common interests. We investigate our approach using two real-world traces collected in two different environments. We first investigate our hypothesis using mobility data collected in an indoor academic environment. We then evaluate our assumptions in an outdoor urban scenario. We present an analysis of our findings, highlighting key social and mobility behaviour factors that can influence such opportunistic solutions. Most importantly, we show that by adding knowledge such as social links between participants in an opportunistic network routing and dissemination algorithms can be greatly improved.**

***Keywords*-opportunistic networking; online social data;**

I. Introduction

Opportunistic mobile networks consist of human-carried mobile devices that communicate with each other in a store-carry-and-forward fashion, without any infrastructure. Compared to classical networks, they present distinct challenges. In opportunistic networks, disconnections and highly variable delays caused by human mobility are the norm. The solution consists of dynamically building routes, as each node acts according to the store-carry-and-forward paradigm. Thus, contacts between nodes are viewed as an opportunity to move data closer to the destination. Such networks are therefore formed between nodes spread across the environment, without any knowledge of a network topology. The routes between nodes are dynamically created, and nodes can be opportunistically used as a next hop for bringing each message closer to the destination. Nodes may store a message, carry it around, and forward it when they encounter the destination or a node that is more likely to reach the destination.

In an opportunistic network, the members are people that carry mobile devices. These people are organized into communities, according to common professions, workplaces, interests, etc. Generally, members of the same community interact with each other more often than with members of outside communities, so the community organization should be taken into consideration when designing algorithms for opportunistic networks. In recent years, due to the advent of social networks and applications, researchers have started showing interest in the use of such elements in opportunistic algorithms. We show here that adding knowledge about social links between opportunistic network nodes to routing and dissemination algorithms greatly improves their effect.

In order for researchers to be able to implement dissemination algorithms for opportunistic networks, real-life traces can be used to offer information about the patterns that people carrying mobile devices follow. Traces are taken using different types of mobile devices for various kinds of communication as well as for different scenarios. We also used such traces and investigate our social approach using the data collected in a tracing experiment that took place between November and December 2011 at the University Politehnica of Bucharest. The validity of our finding is further analysed using a different situation, the case of a trace collected in an urban tracing experiment. We present our findings as well as an analysis of information that may be relevant in designing a good dissemination algorithm for opportunistic networks.

The rest of the paper is structured as follows. Section II presents an overview of related work in the field of routing and dissemination in opportunistic networking. In Section III we present the BUBBLE Rap algorithm and the modified decision approach we have brought to take advantage of social network knowledge. Section IV describes the two traces used in analyzing the proposed approach. Section V describes the experiments performed and shows the results obtained. Finally, Section VI concludes the paper and presents future work.

II. Related Work

A thorough review of opportunistic networking is presented in [1]. The analysis, developed in the context of the EU Haggle project, highlights the properties of main networking functions, including message forwarding, security, data dissemination and mobility models. The authors also propose various solutions for communication in opportunistic networks, and introduce HCMM, a mobility model that merges the spatial and social dimensions. Several well-known opportunistic forwarding algorithms are also presented, such as BUBBLE Rap [2], PROPICMAN [3] and HIBOp [4]. Finally, data dissemination in opportunistic networks is described, along with the presentation of the ContentPlace algorithm.

There are several papers that propose dissemination algorithms for opportunistic networking. Authors of [5] propose Socio-Aware Overlay, an algorithm that creates an overlay for an opportunistic network with publish/subscribe communication. The overlay is composed of nodes with high values of centrality, so that the chosen broker node maintains a higher message delivery rate. The Socio-Aware Overlay algorithm is socially-aware, having its own community detection methods. Thus, the authors of the article propose two algorithms for distributed community detection, named Simple and k-CLIQUE.

Another dissemination algorithm is proposed in [6]. Wireless Ad Hoc Podcasting has the purpose of wireless ad hoc delivery of content among mobile nodes. The technique enables the distribution of content using opportunistic contacts whenever podcasting devices are in wireless communication range. When two nodes are within range of each other, they associate together and start soliciting episodes from the channels they are subscribed to. In order to advertise containing data, the nodes first exchange a Bloom filter hash index that contains all channel IDs that each node offers. Then, each node checks the peer's hash index for channels it is subscribed to. The data exchange phase begins if one of the nodes has found a matching channel, in which case it starts querying for episodes. When two nodes meet and neither has content from a channel the other is subscribed to, one of five solicitation strategies is employed, with the goal of increasing the probability of a node having content to share with other nodes in future encounters.

Authors of [7] propose a dissemination technique called ContentPlace, that attempts to deal with data dissemination in resource-constrained opportunistic networks by making content available in regions where interested users are present, without overusing available resources. In order to optimize content availability, ContentPlace exploits learned information about users' social relationships, to decide where to place user data. ContentPlace's design is based on two assumptions: that the users can be grouped together logically, according to the type of content they are interested in, and that their movement is driven by social relationships. When two nodes meet, they advertise the set of channels the local user is subscribed to. In order to be able to select data from an encountered node, nodes from ContentPlace use a utility function by means of which each node can associate a utility value to any data object. When a node encounters a peer, it computes the utility values of all the data objects stored in the local and in the peer's cache. Then, it selects the set of data objects that maximizes the local utility of its cache, without violating the considered resource constraints. Being a socially-aware algorithm, ContentPlace uses a community representation similar to the caveman model [8], which assumes that users are grouped into home communities, while at the same time having relationships in acquainted communities. By using weights based on the social aspect of opportunistic networking, ContentPlace offers the possibility of defining five different policies, called Most Frequently Visited (MFV), Most Likely Next (MLN), Future (F), Present (P) and Uniform Social (US).

A taxonomy for data dissemination algorithms is proposed in [9]. The authors propose splitting such algorithms in four large categories. The first category deals with the infrastructure of the network, meaning the way the network is organized into an overlay for the nodes. Then, the dissemination techniques are also split according to the characteristics of their nodes, such as node state and node interaction (which includes node discovery, content identification and data exchange). The third category of the taxonomy is represented by content characteristics, meaning the way content is organized and analyzed, and finally the last category (and the most important one) is social awareness. Social awareness is considered to be the future of opportunistic networks, because the nodes in such a network are mobile devices carried by humans, which interact with each other according to social relationships.

Similar to the approach proposed in this paper, the addition of social network information to opportunistic routing has been studied in [10]. The authors consider two types of networks: a detected social network (DSN) as given by a community detection algorithm such as k-CLIQUE and a self-reported social network (SRSN) as given by Facebook relationships. When two nodes meet in their simulation, they exchange data only if they are in the same network (either DSN or SRSN). The authors show that using SRSN information instead of DSN decreases the delivery cost and produces comparable delivery ratio. In this paper we extend the idea and demonstrate an approach where we combine the two sets of data to increase the delivery ratio, without heavily affecting the delivery latency.

III. Opportunistic Routing

The results presented in this paper have been obtained by improving the distributed BUBBLE Rap algorithm (DiBuBB [2]) so that it takes into account social relationships between participants. Therefore, in this section we present first the DiBuBB algorithm, followed by the proposed improvements that incorporate social data into the opportunistic approach.

A. BUBBLE Rap and DiBuBB

BUBBLE Rap [2] is a routing algorithm for opportunistic networks that uses knowledge about nodes' communities to deliver messages. It assumes that a mobile device carrier's role in the society is also true in the network, thus the first part of the algorithm is to forward data to nodes that are more popular than the current node. The second assumption made in BUBBLE Rap is that the communities people form in their social lives are also observed in the network layer, therefore the second part of the algorithm is to identify the members of the destination community and pass them the message. Thus, a message is bubbled up the hierarchical ranking tree using a global popularity level, until in reaches a node that is in the same community as the destination. Then, the message is bubbled up using a local ranking until it reaches the destination. The popularity of a node is given by its betweenness centrality, which is the number of times a node is on the shortest path between two other nodes in the network.

Community detection is done using k-CLIQUE [11], an algorithm that dynamically detects the community of a node by analyzing its encounters with other devices. There are two important parameters to the k-CLIQUE algorithm: the contact threshold and the community threshold. The contact threshold specifies the amount of time that two nodes have to be in contact before being considered as part of the same community, while the community threshold is used to specify the number of community nodes two encountering devices must have in common in order for them to belong to the same community. The centralities are computed by replaying the last collected mobility trace, applying a flooding algorithm and then computing the number of times a node acts as a relay on a shortest path.

However, this implementation of BUBBLE Rap proves to be unfeasible in real life. Therefore, a distributed version entitled DiBuBB was also proposed by the authors [2]. It uses distributed k-CLIQUE for community detection and a cumulative or single window algorithm for distributed centrality computation. The single window (S-window) algorithm computes centrality as the number of encounters the current node has had in the last time window (chosen usually to be six hours), while the cumulative window (C-window) algorithm counts the number of individual nodes encountered for each time window and then performs an exponential smoothing on the cumulated values.

B. Social Connections in DiBuBB

We believe that using information about the social relationships between members of an opportunistic network can increase the effectiveness of routing. Therefore, we propose and present here several modified versions of the DiBuBB algorithm that take advantage of the social connections matrix, entitled *Social*, *Max*, *Popularity* and *Popularity Squared*.

1) Social: When two nodes meet in DiBuBB, they first check if they belong in the same community according to k-CLIQUE, and if they do they compare their local centralities. Then the algorithm continues as described before. However, in the first modified version of the algorithm, instead of using k-CLIQUE communities, we use information from the social network. Thus, the nodes will use the local centrality value if they share a social link (i.e. they are friends on Facebook). If there is no social link between the nodes, the global centrality is used. The centrality values are computed in the same way as in DiBuBB.

2) Max: By taking this approach one step further, we used the social network in the computation of centrality values as well. Therefore, the Max version of DiBuBB computes a node's centrality according to the following formula:

$$centrality = max(C_{window}, popularity)$$

C_{window} is the original value of the centrality as computed by DiBuBB using the C-window algorithm and $popularity$ is the number of social links a node has. Both of these values are normalized, so they are between 0 and 1.

3) Popularity: In the Popularity version of our proposed algorithm, two encountering nodes are considered to be in the same community if either they are seen as such by k-CLIQUE or if they have a social distance of less than 3 (meaning that they are directly connected or they share a common Facebook friend). The centrality value in this case is computed using the following formula:

$$centrality = w_1 * C_{window} + w_2 * popularity$$

The C_{window} and $popularity$ values are also normalized and are then multiplied by w_1 and w_2, which are weight values that follow the condition $w_1 + w_2 = 1$.

4) Popularity Squared: This version is similar to the Popularity version, but uses a slightly different formula:

$$centrality = w_1 * C_{window}^2 + w_2 * popularity^2$$

These four DiBuBB versions have the role of pointing out the advantage of using information about social relationships in routing in an opportunistic network.

IV. TRACES

To analyse the efficiency of our proposed algorithm (as well as to evaluate the different formulas used by the routing decision process), we implemented a simulator that takes as input a real-life mobility trace and re-executes on top of it the opportunistic dissemination algorithm. In our case we investigated our approach starting from two different mobility traces. The two traces are *St. Andrews* [10] and *UPB* [12], presented in detail in this section.

A. St. Andrews

The St. Andrews trace [10] was collected using a mobile sensor network with Tmote invent devices carried by 27 participants from the University of St. Andrews. Out of these participants, 22 were undergraduate students, 3 were postgraduate students and 2 were members of the staff. The experiment was performed for a period of 79 days, in which the participants were asked to carry their devices and to keep them on at all times, whether in or out of the town of St. Andrews.

The invent devices are able to detect and store information about encounters between each other within a radius of 10 metres, and were programmed to send discovery beacons at every 6.67 seconds. The encounter information, comprised of timestamp, the sending and the detected device's IDs were occasionally uploaded to one of three basestations across the two Computer Science buildings located in the campus of the university. This information was used to create a trace of encounters between Tmotes during the duration of the experiment (the DSN, as presented in Section II). In addition, a topology (the SRSN) was generated using the participants' Facebook information. The nodes were logically split into three large roles according to the SRSN and four weakly-defined roles according to the DSN.

B. UPB

The UPB trace is the result of a social tracing experiment that took place between November and December 2011 at the University Politehnica of Bucharest, which shows not only the interactions of the experiment participants, but also the social relationships they have with each other according to their Facebook profiles. The experiment collected mobility data using an Android application called Social Tracer [13]. The participants were asked to run the application whenever they were in the faculty grounds. Social Tracer sends regular Bluetooth discovery messages at certain intervals, looking for any type of device that has its Bluetooth on. These include the other participants in the experiment, as well as phones, laptops or other types of mobile devices in range.

When encountering another Bluetooth device, the Social Tracer application logs data containing its address, name and timestamp. The address and name are used to uniquely identify devices, and the timestamp is used for gathering contact data. Data logged is stored in the device's memory, therefore every once in a while participants were asked to upload the data collected to a central server located within the faculty premises. All gathered traces were then parsed and merged to obtain a log file with a format similar to the ones in [14]. Successive encounters between the same pair of devices within a certain time interval were considered as continuous contacts, also taking into consideration possible loss of packets due to network congestion or low range of Bluetooth. The experiment lasted for a period of 35 days, and involved a total of 22 participants, chosen as statistically varied as possible in order to obtain a good approximation of the mobility aspects of a real academic environment. Thus, there were twelve Bachelor students (one in the first year, nine in the third and two in the fourth), seven Master students (four in the first year and three in the second) and three research assistants. The participating members were asked to start the application whenever they arrived at the faculty and to turn it off when they left.

It is shown in [12] that the UPB trace follows the observations in [15], namely that the contact and intercontact times correspond to a power law distribution. It is also shown that the participants have been chosen well so that they represent different groups from the social and logical grouping of nodes in a network based on mobile device carriers in an academic environment. Finally, the k-CLIQUE algorithm has been applied on the trace to show that the local grouping of participants into study years yields similar results to a dynamic grouping such as k-CLIQUE, as well as to the social network organization.

V. EXPERIMENTAL SETUP AND RESULTS

This section describes the experiments performed after modifying the base DiBuBB algorithm to include the information collected for the social connections. The analysis of the obtained results shows an improvement in the hit rate of delivering the messages.

A. Experimental Setup

We performed two series of experiments for the two traces presented above. The UPB trace corresponds to an enclosed academic environment. The St. Andrews trace, on the other hand, was chosen to validate our algorithm in a different, more open and largely distributed type of environment. An advantage of both these traces is that they include information about the social network formed by the participants (they were asked to also provide publically-available Facebook information).

For our experiments we implemented an emulator entitled MobEmu that parses the two traces and then applies an opportunistic routing algorithm at every encounter. Because in both of the traces there are a large number of external nodes that are only seen once, we only took into consideration for our experiments the internal nodes (i.e. nodes that are actual participants in the experiment). We believe that an approach such as the one proposed in this paper would best be suited in a college or faculty (an enclosed space), where social information about each node is known and can be used efficiently in data routing.

In each set of experiments we applied the base DiBuBB algorithm, along with our four versions presented in the previous section. For Popularity and Popularity Squared, we tested with all possible weight combinations in increments of 0.1, but we only show here the results of the best values for each, as these weights can easily be adapted on-the-fly based on the conditions of the network.

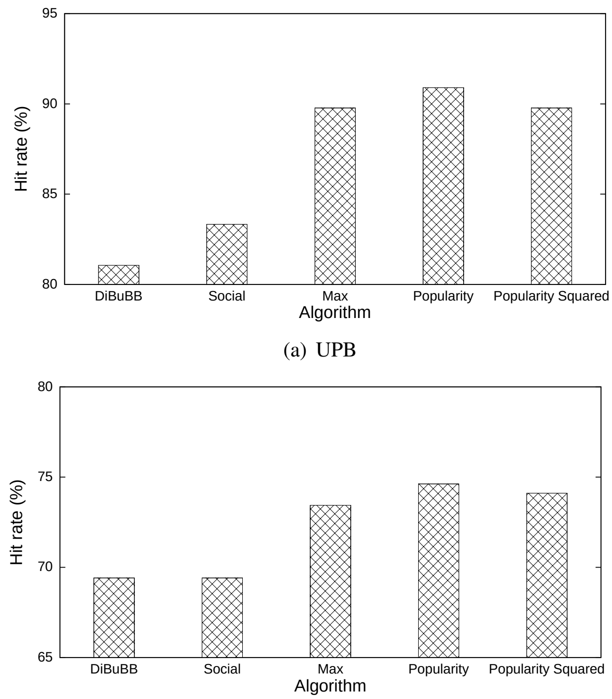
As stated in Section III, k-CLIQUE has a contact threshold and a community threshold. These have been chosen differently for the two traces, after analyzing their respective information. Therefore, the contact threshold for UPB is 5 and the community threshold is 15 minutes, while for St. Andrews we ran the experiments using a contact threshold of 4 and a community threshold of 10 minutes. By using these values, we obtained communities with similar sizes in relation to the total number of nodes in the experiments. Choosing the two thresholds is a relatively simple process: it is done by looking at the average total contact duration and the average number of encounters per node of the trace, and according to how many nodes a community should have, the two thresholds are computed.

Each node in the experiments sends a number of messages equal to the total number of nodes in the network. All messages are sent at the beginning of a run, following a Zipf distribution with an exponent of 1. Thus, a node sends the largest amount of messages to a node in its own social network, then to nodes belonging to the same community according to k-CLIQUE and finally to nodes that are not in relation to it (according to both k-CLIQUE and the social network). We chose a Zipf distribution of messages because it has been shown that data requests and sends follow power law distributions [16]. The choice of a node inside a community is done randomly, but the same seed is used when repeating an experiment in order to get the same values. We considered that each node in the network has a limited storage amount of 20 messages (assuming all messages have the same size) and that the network bandwidth is unlimited.

Unfortunately, there were three nodes in the UPB trace that belonged to participants that either have not uploaded their data to the collecting server or have not attended courses during the experiment. Therefore, in order to manage uncertainty, we have eliminated these nodes from the experiment, as results including them would not have been relevant.

B. Results

We applied the base, Social, Max, Popularity and Popularity Squared versions of DiBuBB on the two traces, looking at certain metrics. The first and most important metric that we chose is *hit rate*, which is computed as the ratio between successfully delivered and total messages. It suggests the efficiency of a routing algorithm and ideally it would be 100%. It shows the fraction of requests that can be served by a routing or dissemination algorithm. Another used metric is the *delivery cost*, represented by the ratio between the total number of exchanged messages during the course of the experiment and the number of generated messages. It should be as low as possible and it shows the congestion of the network. The *latency* values show the time (in seconds) passed between generating a message and delivering it to the destination. In an opportunistic network, which is a type of delay tolerant network (DTN [17]), delivery latency is not as important, but nonetheless it should be improved when possible. Finally, the *hop count* is the number of nodes that carried a message until it reached the destination on the shortest path.

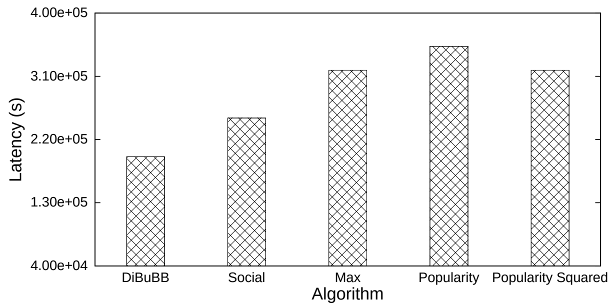


(a) UPB

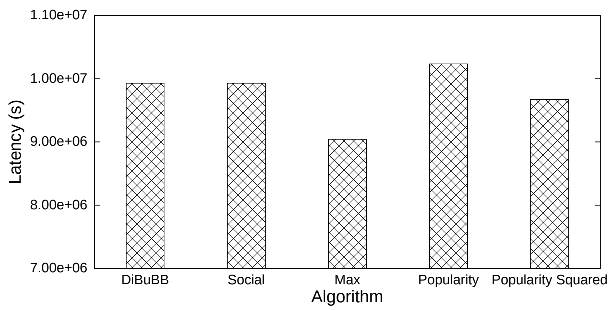
(b) St. Andrews

Figure 1. Hit rate for the base, Social, Max, Popularity and Popularity Squared versions of DiBuBB.

Figure 1 shows the hit rate values for the UPB and St. Andrews traces. It can be seen that we obtain a hit rate greater than or equal to the hit rate of the base DiBuBB algorithm for all four of our versions and both traces. For the UPB trace shown in Figure 1(a) the hit rate for the base version is 81.06%, while the maximum hit rate is achieved using Popularity and has a value of 90.90%, therefore bringing an improvement of 9.84%. Popularity Squared and Max also yield close results, each of them having a maximum hit rate of 89.77%. Figure 1(b) presents the hit rate values for the St.



(a) UPB



(b) St. Andrews

Figure 2. Average latency for the base, Social, Max, Popularity and Popularity Squared versions of DiBuBB.

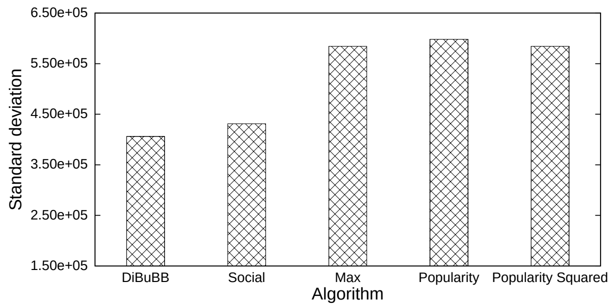
Andrews trace, and it can be observed that the maximum hit rate improvement (of 5.21%) is obtained using the Popularity version as well. Again, the same as for the UPB trace, Max and Popularity Squared bring similar improvements. An interesting conclusion that can be drawn from Figure 1 is that because the two traces differ in terms of scope, their hit rates also differ significantly. The UPB trace was performed in an enclosed indoor environment, where nodes interact often and for long intervals and are grouped close together, therefore the hit rate is higher because messages circulate better. On the other hand, the St. Andrews trace, being performed outdoors, is based on a network where the nodes are spread wide and do not interact so often. The hit rate for this trace is lower because nodes do not get the chance to forward messages so frequently. The social-based improvements brought to DiBuBB work better for the UPB trace because there is a higher chance of a social relationship existing between two individuals in the same small space (such as a faculty or an office building) then between individuals from the same city. As previously stated, we believe that the social network approach is more effective in such an enclosed space.

Figure 2 presents the latency values obtained by applying the social versions of DiBuBB on the traces. For the UPB trace, these values are significantly increased. A possible explanation is that these nodes belong to participants that do not have (or have not provided) a Facebook account, therefore the centrality values are computed only according to k-CLIQUE. If we eliminate these nodes (along with the ones belonging to participants that have not uploaded their information, as shown in the previous subsection) we obtain latencies closer to the original values. Another explanation may be that obtaining a higher hit rate means that some messages reach nodes that they weren't reaching in the original version, and if these nodes are rarely seen or very remote, their latencies automatically increase the average latency. This conclusion is further proven by Figure 3(a) which shows the standard deviation for the UPB trace. There is a big difference between base DiBuBB and Social on one hand, and Max, Popularity and Popularity Squared on the other hand, in terms of standard deviation, which is corroborated with the differences in latency and hit rate between them. A way of solving the high latency problem is to use the external nodes as message carriers when running the four DiBuBB versions. In a real-life situation, the number of participants in an opportunistic network represented by an academic environment is much larger, thus the chance of getting data even to these remote nodes in a timely manner greatly increases.

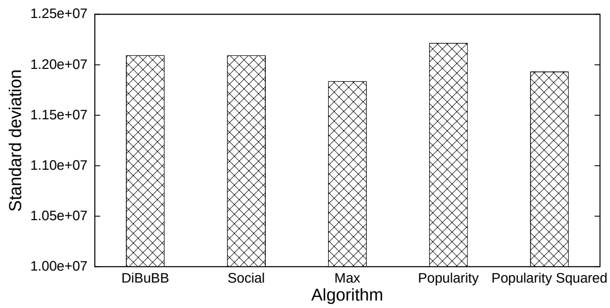
However, for the St. Andrews trace, using the social network in opportunistic routing actually helps improve the average latency values as well. This happens because the St. Andrews trace contains social network information about all the participating nodes, but also because the encounters between devices are spread more evenly than for the UPB trace. Figure 3(b) shows that standard deviation values are very close to each other as well. An interesting observation regarding Figures 2 and 3 is that the difference in order between latency values for the two traces is about one, which further highlights that an enclosed environment helps disseminate data faster using social opportunistic routing.

Finally, Table I shows the delivery cost and average hop count for our experiments. As it can be seen, generally the social-based DiBuBB versions increase the values of these metrics, but Max has better results than the original DiBuBB in terms of hop count in both traces. The reason is that Max is a somewhat greedy approach to opportunistic routing which basically attempts to minimize the number of nodes between the source and the destination. As in all greedy algorithms, problems arrise when the implementation becomes stuck in a local maximum and the number of messages sent between the same nodes increases. This is also the reason why the delivery cost is higher for Max than for the base version. The same conclusions as before regarding the differences between the two traces can be drawn, namely that UPB allows for a better and more efficient circulation of messages.

We have shown in this section that using social information about the nodes in an opportunistic network can lead to improved results in terms of hit rate, latency, delivery cost or hop count. Unfortunately, all these values can't be



(a) UPB



(b) St. Andrews

Figure 3. Standard deviation of latency for the base, Social, Max, Popularity and Popularity Squared versions of DiBuBB.

Table I
DELIVERY COST AND AVERAGE HOP COUNT FOR THE BASE, SOCIAL, MAX, POPULARITY AND POPULARITY SQUARED VERSIONS OF DIBUBB.

	Delivery cost		**Average hop count**	
	UPB	**St. Andrews**	**UPB**	**St. Andrews**
DiBuBB	3.93	958.14	1.39	515.17
Social	4.35	956.26	2.24	514.55
Max	10.4	1301.03	1.16	482.99
Popularity	7.71	1855.55	5.08	535.91
Popularity Squared	10.07	1906.05	5.89	595.01

improved at the same time in this implementation, so it is the network administrator's job to choose a suitable DiBuBB version according to the type of environment the algorithm is run in.

VI. CONCLUSIONS

Recently different opportunistic routing and dissemination algorithms were proposed and evaluated in various scenarios emulating real-world phenomena as close as possible. Such algorithms generally rely on mobility patterns of users and the context of communication. In this we proposed and investigated the addition of social data to improve the performance of communication algorithms and data transmission schema. When the routing decision is influenced by the chance of a particular user being able to successfully carry the data to the next hop, we believe that opportunistic communication algorithms could greatly benefit not only from learning the behaviour of users, but also their history of contacts coupled with the online social familiarity patterns between them. We believe users tend to be in contact more with familiar sets of users, with whom they share common interests.

We analyzed our approach using two real-world traces collected in two different environments. We consider that opportunistic social networks are the future of mobile communication, especially in a world with more and more content available and with a higher degree of connectivity between individuals. Therefore, having real world traces of human movement and knowing that social relationships govern human interaction are paramount to creating suitable routing and dissemination algorithms.

We first investigated our hypothesis using mobility data collected in an indoor academic environment. We then evaluated our assumptions in an outdoor urban scenario. We presented an analysis of our findings, highlighting key social and mobility behaviour factors that can influence such opportunistic solutions. Most importantly, we showed that by adding knowledge such as social links between participants in an opportunistic network, routing and dissemination algorithms can be greatly improved.

ACKNOWLEDGMENT

The research presented in this paper is supported by national project: "TRANSYS Models and Techniques for Traffic Optimizing in Urban Environments", Contract No. 4/28.07.2010, Project CNCSIS-PN-II-RU-PD ID: 238. The work has been co-funded 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. All authors have equal contributions to the article. This paper has benefited from the collaborative research efforts of the EU Green-Net group.

REFERENCES

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

[2] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '08. New York, NY, USA: ACM, 2008, pp. 241–250. [Online]. Available: http://doi.acm.org/10.1145/1374618.1374652

[3] H. A. Nguyen, S. Giordano, and A. Puiatti, "Probabilistic routing protocol for intermittently connected mobile ad hoc network (propicman)," *2007 IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pp. 1–6, 2007.

[4] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella, "HiBOp: a History Based Routing Protocol for Opportunistic Networks," in *World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. IEEE International Symposium on a*, Jun. 2007, pp. 1–12. [Online]. Available: http://dx.doi.org/10.1109/WOWMOM.2007.4351716

[5] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks," in *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*, ser. MSWiM '07. New York, NY, USA: ACM, 2007, pp. 225–234. [Online]. Available: http://doi.acm.org/10.1145/1298126.1298166

[6] V. Lenders, M. May, G. Karlsson, and C. Wacha, "Wireless ad hoc podcasting," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 12, pp. 65–67, January 2008. [Online]. Available: http://doi.acm.org/10.1145/1374512.1374535

[7] C. Boldrini, M. Conti, and A. Passarella, "ContentPlace: social-aware data dissemination in opportunistic networks," in *Proceedings of the 11th international symposium on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '08. New York, NY, USA: ACM, 2008, pp. 203–210. [Online]. Available: http://doi.acm.org/10.1145/1454503.1454541

[8] J. Wu and D. J. Watts, "Small worlds: the dynamics of networks between order and randomness," *SIGMOD Rec.*, vol. 31, pp. 74–75, December 2002. [Online]. Available: http://doi.acm.org/10.1145/637411.637426

[9] R. Ciobanu and C. Dobre, "Data dissemination in opportunistic networks," in *18th International Conference on Control Systems and Computer Science*, ser. CSCS-18, 2011, pp. 529–536.

[10] 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

[11] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proceedings of 2nd ACM/IEEE international 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

[12] R. I. Ciobanu, C. Dobre, and V. Cristea, "Social aspects to support opportunistic networks in an academic environment," in *11th International Conference on Ad-Hoc Networks and Wireless*, ser. ADHOC-NOW, 2012.

[13] Social Tracer, "http://code.google.com/p/social-tracer/."

[14] CRAWDAD, "http://crawdad.cs.dartmouth.edu/."

[15] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN '05. New York, NY, USA: ACM, 2005, pp. 244–251. [Online]. Available: http://doi.acm.org/10.1145/1080139.1080142

[16] 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

[17] F. Warthman, "Delay-tolerant networks (DTNs) - a tutorial," Mar. 2003. [Online]. Available: http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf