Social Aspects to Support Opportunistic Networks in an Academic Environment

Radu Ioan Ciobanu, Ciprian Dobre, and Valentin Cristea

University Politehnica of Bucharest, Romania
Faculty of Automatic Control and Computers
radu.ciobanu@cti.pub.ro, {ciprian.dobre,valentin.cristea}@cs.pub.ro

Abstract. As wireless and 3G networks become more crowded, users with mobile devices have 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 neighboring smartphones. Recently various opportunistic routing or dissemination algorithms were proposed and evaluated in different scenarios emulating real-world phenomena as close as possible. In this paper we present an experiment performed at the Politehnica University of Bucharest in which we collected social and mobiltity data to evaluate opportunistic routing and dissemination algorithms. We present an analysis of our findings, highlighting key social and mobility behavior 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.

1 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 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

X.-Y. Li, S. Papavassiliou, S. Ruehrup (Eds.): ADHOC-NOW 2012, LNCS 7363, pp. 69–82, 2012.

different scenarios. This paper describes a social tracing experiment that took place between November and December 2011 at the Politehnica University of Bucharest and the way it was implemented. Furthermore, we analyze the results obtained and try to gather information that may be relevant in designing a good dissemination algorithm for opportunistic networks.

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.

2 Related Work

There are two ways of testing the performance of a data dissemination algorithm in an opportunistic network. First of all, traces such as the ones presented in this paper can be taken from various situations. Such traces have been performed for WiFi [1,2] or Bluetooth [3]. The Bluetooth trace experiment is similar to the one presented in this paper in terms of number of participants, but the duration differs greatly, as the traces in [3] were performed for 3 and 5 days (compared to 35 in our case, as presented in Sect. 3). Another difference between our trace and the ones from [3] is that the iMote traces are performed by nodes that interact with each other for long times during the day, as the carriers work in the same enclosed place for large parts of the day. We show in Sect. 3 that our trace covers a larger array of node types because the participants are not grouped together. A good place for finding mobility traces for various situations is CRAWDAD [4], a community resource for archiving wireless data. The second way of testing a dissemination algorithm is to use mobility models. There have been several such models proposed in recent years. The research began with random models such as the waypoint model, and continued with mobility models that take into consideration the social aspect of human movement [5] as well as the attraction of physical locations [6]. Existing human mobility models for oppotunistic networks, including models that explore location preference or use the social graph, are reviewed in detail in [7], and a taxonomy for classifying such models is proposed.

A thorough review of opportunistic networking is presented in [8]. 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 [9], PROPICMAN [10] and HIBOp [11].

There are several papers that propose dissemination algorithms for opportunistic networking. Authors of [12] 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 [13]. Choosing the next-hop node is a scheduling hard problem, with fault-tolerant requirements [14]. 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. Authors of [15] 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. 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.

A taxonomy for data dissemination algorithms is proposed in [16]. 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 [17]. 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. Several other papers address the issue of using social information in opportunistic networks. In [7], an analytical model for the expected number of hops and delay of messages delivered in a social-based opportunistic routing algorithm is proposed, where the forwarding process is modeled as a semi-Markov process. Social information about the participants in an opportunistic network can be used not only for data forwarding, but also for content sharing. Thus, the authors of [18] propose a context- and social-aware middleware that learns context and social information about the nodes in the network, which is then used to predict their future movement. The middleware was integrated with the Haggle architecture and was used for content sharing, yielding up to 200% improvement in terms of hit rate and 99% reduction in resource consumption in terms of traffic generated in the network.

3 Social Tracing

In order to obtain trace information regarding the mobility of the members of a faculty, a real-world tracing experiment has been performed at the Politehnica University of Bucharest, in the autumn-winter season of 2011. This section presents the setup and additional details about this experiment.

3.1 Social Tracer

Tracing was performed using an Android application we developed entitled Social Tracer, which is presented in more detail in [19]. The participants have been asked to run the application whenever they are in the faculty grounds, as we were interested in collecting data about the mobility and social traces in an academic environment. 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 type of mobile devices in range. The reason Bluetooth was preferred to WiFi is mainly the battery use [20]. For example, in 4 hours of running the application on a Samsung I9000 Galaxy S with discovery messages sent at every 5 minutes, the application used approximately 10% of the battery's energy. The period between two successive Bluetooth discovery invocations can be set from the application, ranging from 1 to 30 minutes (the participants have been asked to keep it as low as possible, in order to have a more fine-grained view of the encounters).

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 so far 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 [4]. 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. Data gathered this way is presented and analyzed in Sect. 4.

3.2 Experimental Setup

The experiment was performed for a period of 35 days at the Politehnica University of Bucharest between November 18 and December 22 2011. There were a total of 22 participants, chosen to be as varied as possible in terms of year, in order to obtain a better approximation of mobility in 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, because we were only interested in the mobility patterns and social interaction in the academic environment. However, this did not always happen, but the outcome of the experiment was not affected because the only devices seen after leaving the faculty were external devices.

4 Trace Analysis

This section presents a detailed analysis of the logs obtained at the end of the experiment described in Sect. 3.

4.1 Details

We define internal devices as the ones carried by the participants in the experiment, while external devices are represented by other nodes encountered during the course of the experiment. There were 22 internal devices numbered from 0 to 21. The total number of contacts between two internal devices (i.e. internal contacts) was 341, while the number of external contacts was 1127. A contact is considered to start at the first time a certain device was seen and to end at the last time it was seen in a given time interval. There were 655 different external devices sighted during the course of the experiment. This means that in average each different external device has been seen about 2 times. External devices may be mobile phones carried by other students or laptops and notebooks found in the laboratories at the faculty. Some of these external devices have high contact times because they may belong to the owner of the internal device that does the discovery, therefore being in its proximity for large periods of time. However, external contacts are in general relatively short.

4.2 Contact and Inter-contact Times

Encounters in an opportunistic network are characterized by two important notions: contact time and inter-contact time. The contact time represents the duration of a contact between two devices from the moment they discover they are in range until the moment the link between them is gone. This represents the time window in which the two participating nodes can send data to each other. Inter-contact times are intervals between two successive encounters of the same two devices. They are relevant in deciding whether data should be sent directly between two nodes when they are in range or whether it should be relayed to a third node for forwarding.

Figure 1 shows the distribution of contact and inter-contact times for the entire duration of the experiment (ranging from 2 minutes to 35 days) for all internal devices. Axis Y presents the percentage of time values that are greater than the time on axis X. As shown in [3], the distribution of contact times follows an approximate power law for both internal and external devices, as well as contact time and inter-contact time. The contact time data series are relevant when discussing the bandwidth required to send data packets between the nodes in an opportunistic network, because they show the time in which a device can communicate with other devices. As stated before, the number of internal contacts is 341, with the average contact duration being 30 minutes, which means that internal contacts have generally been recorded between devices belonging to students attending the same courses or lecturers and research assistants teaching those courses. External contacts also follow an approximate power law, with an average duration of 27 minutes. However, in this case there are certain external contacts that have a duration of several hours. This situation is similar to the one previously described, where these devices belong to the same person carrying the internal device. The inter-contact time distribution shows a heavy tail property, meaning that the tail distribution function decreases slowly. The impact of such a function in opportunistic networking has been studied in more detail in [21] for four different traces. The authors conclude that the probability of a packet being blocked in an inter-contact period grows with time and that there is no stateless opportunistic algorithm that can guarantee a transmission delay with a finite expectation.

Figure 2 shows contact and inter-contact times for encounters with any nodes. Thus, contact time in this case (called any-contact time in [3]) represents the time in which any internal or external node is in range with the current observer, while the inter-any-contact time is the time when the current device does not see anyone in range. These any-contact times are greater than regular contact times, but the shape of the distribution is also a power law function. A conclusion that can be drawn from these charts is, as observed in [3], that durations of contact times are bigger and intervals between contacts are smaller, so if a node wants to perform a multicast or to publish an object in a publish/subscribe environment it has a great chance of being able to do so.

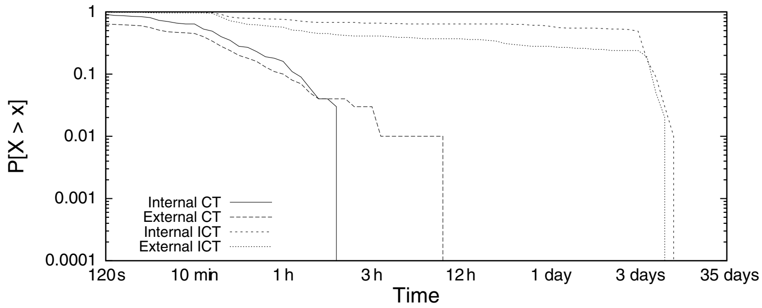

Fig. 1. Probability distribution of contact and inter-contact times (CT = contact time, ICT = inter-contact time)

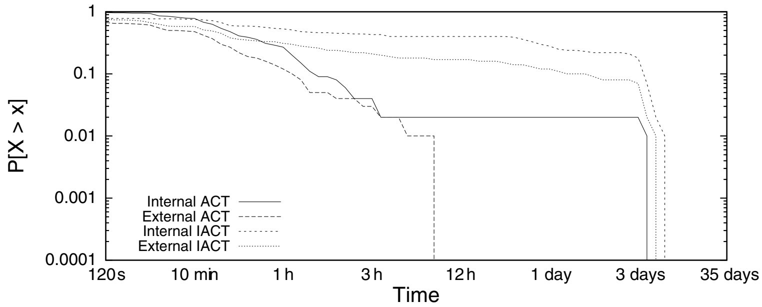

Fig. 2. Probability distribution of any-contact and inter-any-contact times (ACT = any-contact time, IACT = inter-any-contact time)

4.3 Contact Distribution

Figure 3 shows the distribution of the number of times a node (internal or external) was sighted by a device participating in the experiment. It can be seen that the maximum number of encounters of an internal device is 55 during the course of the 35 days of the experiment, whereas some internal nodes have never been seen. Most internal devices have been seen from 16 to 20 times. As for external devices, the majority of them have been encountered less than 5 times, with 534 of them having been sighted only once. There are few exceptions, as three external devices have been encountered more than 16 times. The conclusion is that there is a large number of nodes available in such an environment that can be used to relay a message, meaning that there is a lower chance of traffic congestion. Figure 4 presents the number of times specific pairs of devices saw each other. It shows that the maximum number of contacts between two internal nodes or an internal and an external node is 17. Generally the number of contacts with external devices is larger than the number of contacts with internal devices. This shows that the participants in the experiment have been chosen well so that

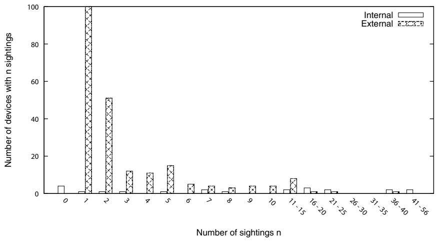

Fig. 3. Distribution of the number of sightings of a device

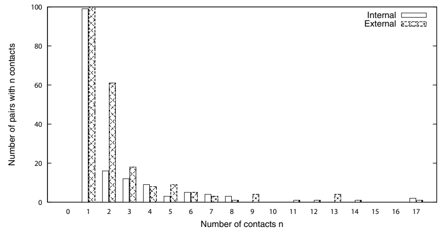

Fig. 4. Distribution of the number of contacts between pairs of devices

they represent various groups from the social and logical grouping of nodes in a network based on mobile device carriers in an academic environment.

4.4 Communities and Social Structures

As stated in Sect. 2, the social aspect has become very important in the world of opportunistic networking, because mobile devices are carried by people that are organized into communities and social circles. Users from the same community or social circle tend to interact more with each other, so relaying a packet by taking into consideration the community an encountered node belongs to could lead to a lower latency and a better hit rate. Human mobility models such as CMM [5] and HCMM [6] have been proposed and implemented, but an experimental approach could show the interaction patterns better.

The environment represented by the faculty grounds already has a logical organization into communities, namely the groups of students and the teachers or lecturers. At the University Politehnica of Bucharest there are four years for Bachelor students, each split into several groups of about 30 students each. For Master's students, the two years are formed of seven directions with about 20 students each. We tried to choose the participants in the experiment so that the distribution would be as good as possible, as shown in Sect. 3. However, because there were only 22 participants in our experiment, we decided that the logical grouping should be done by year instead of group. We applied the k-CLIQUE algorithm for community detection [22] on the traces collected by Social Tracer. k-CLIQUE 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 community graph obtained after applying k-CLIQUE is presented in Fig. 5, along with the logical organization of participants into year groups. In the figure, Bachelor students are represented as triangles (first year), circles (third year) and double circles (fourth year), Master's students are shown as pentagons (first year) and diamonds (second year), while assistants and lecturers are squares. An arrow from node A to node B means that A sees B as part of its community. The algorithm was applied only for internal nodes, using a contact threshold of fifteen minutes and a community threshold of five nodes, values that were chosen after analyzing the traces. Because social networks and groups of communities are represented as matrices, we define the similarity value between two matrices as the percentage of values that are equal in both of them. Thus, the similarity value between the k-CLIQUE graph and the logical distribution of participants into year groups is 79.95%. This shows that k-CLIQUE functions correctly in the case of our trace.

Because a logical grouping into communities may not always be as straightforward as in this case, the social relationships between device owners can be taken into account. The social graph of the participants in our experiment is shown in Fig. 6, where the year group representation is the same as in Fig. 5 and the edges symbolize a social link.. Some nodes (such as 1, 4, 8 and 20) are represented by students that participated in the experiment but did not have or did not provide a Facebook account. It can be observed from Fig. 6 that the node with the most social links (12) is 11, which is followed by nodes 3 and 7 with 11 links each. It can also easily be seen that most nodes that are in the same community share a social link between them, as well as the fact that in most cases the number of social links of a node is close to the number of communities it belongs to according to k-CLIQUE. This means that a more popular node in terms of social relationships will belong to more communities, which makes it a better candidate for relaying data for other nodes in the opportunistic network. The similarity value for the social network organization and k-CLIQUE is 83.06%, showing that k-CLIQUE is even better at detecting social communities than is it for logical grouping.

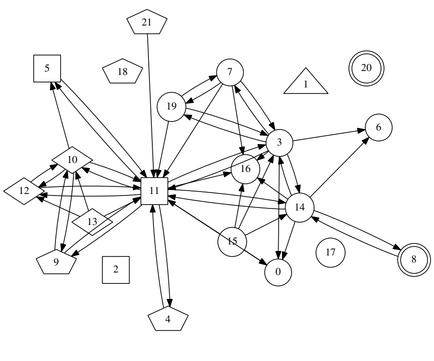

Fig. 5. The graph of participants in the experiment as computed by k-CLIQUE

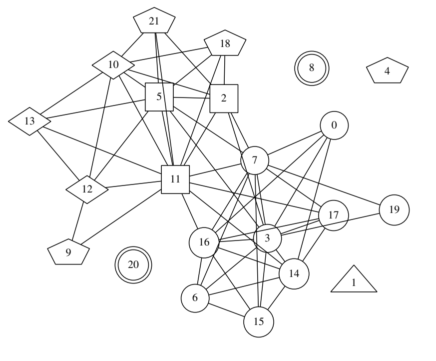

Fig. 6. The social graph of participants in the experiment (via Facebook)

5 Opportunistic Networking Using Social Organization

The social organization of members in an opportunistic network can be used to improve the effectiveness of routing algorithms. In order to prove this, we have implemented some modified versions of the distributed BUBBLE Rap algorithm (DiBuBB [9]) that take social links into consideration when routing. This section describes our modified versions of DiBuBB along with the simulation setup, the metrics used for analyzing the performance of our improvements and the results obtained through the simulation.

5.1 Simulation Setup

BUBBLE Rap [9] is a routing algorithm for opportunistic networks that uses knowledge about nodes' communities and centralities to deliver messages. It uses a node's betweenness centrality, which represents the number of times a node is on the shortest path between two other nodes in the network. Each node in BUBBLE Rap has two centrality values, a local one (for its own community) and a global one. A node sends a message to nodes that have a higher global centrality than it until the message arrives at the destination community. Then, the local ranking is used to send the message up the community hierarchy until it reaches the destination. Community detection is done using k-CLIQUE, while the centralities are computed by carrying out an emulation that replays collected mobility traces, applies a flooding algorithm, and then computes the number of times a node acts as a relay on a shortest path. However, such a method is not feasible in real life, so a distributed version of BUBBLE Rap entitled DiBuBB was also proposed by the authors. It uses distributed k-CLIQUE [22] for community detection and a cumulative or single window algorithm for distributed centrality computation.

We have also implemented a version of DiBuBB to test the trace data presented in this paper. As stated in Sect. 3, we use k-CLIQUE with a contact threshold of fifteen minutes and a community threshold of five nodes to detect the communities. For computing the centrality values for each node in a distributed fashion we implemented the cumulative window (C-window) method, which counts the number of individual nodes encountered for each six-hour time window and then performs an exponential smoothing on the cumulated values. We will refer to this version from now on as "base".

We consider that knowledge about the social relationships between members of an opportunistic network can increase the effectiveness of routing. Therefore, we modified the base version of DiBuBB to use the social network matrix instead of k-CLIQUE. Thus, when two nodes meet, instead of checking if they belong in the same community according to k-CLIQUE, they look for a social link between them. If that social link exists, then the nodes will compare their community centralities, and the one with the lower value will send its messages to the other one. If there is no link between the nodes, the global centralities will be verified. This will be refered to as the "social" version.

We then tried to take this approach one step further, by using the social network in the computation of centrality values as well. When there is an encounter between two nodes, they 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 (i.e. they are directly connected or they share a common friend). The centrality value is computed according to the following formula: $centrality = w_1 * S_{window} + w_2 * popularity$, where S_{window} is the original value of the centrality as computed by DiBuBB, *popularity* is the number of social links a node has, and w_1 and w_2 are weight values that follow the conditions $w_1 + w_2 = 1$ and $w_1 > w_2$. For nodes that are not part of the social network (e.g. participants in the tracing experiment that do no own or have not provided a Facebook account link), the centrality will be computed the same as in the base version of DiBuBB. Having weights for the two components of the centrality value allows us to fine-tune the algorithm according to the trace it is applied on. We will refer to this version as the "popularity" version.

In the simulation scenario, each node sends 11 messages to other nodes in the opportunistic network. We managed uncertainty while testing by eliminating from the list of internal devices the ones that have few or no encounters with other internal devices. Such nodes are participants in the experiment that either have not turned off the Social Tracer application when arriving at the faculty, or they have not been attending classes. There were four test cases in which nodes sent messages to other randomly chosen nodes (the destinations were kept the same between the three versions of DiBuBB for each test case).

5.2 Results

We applied the base, social and popularity versions of DiBuBB on the trace collected in our experiment. 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 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), 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.

The results of our testing scenario are shown in Table 1. It can be seen that the hit rate increases from base to social to popularity in each of the four test cases. The improvement caused by the social version is very significat, going up to 16% in some cases. The popularity version offers a smaller but very important

Table 1. Results of applying the three versions of DiBuBB (base - B, social - S, popularity - P) on four test cases. Latencies are in the DD:HH:MM:SS format

Test Case	**Run 1**			**Run 2**		
Metric	**B**	**S**	**P**	**B**	**S**	**P**
Hit Rate	81.81%	97.72%	98.48%	82.57%	98.48%	99.24%
Delivery Cost	5.89	8.76	15.71	5.87	9.06	15.43
Avg Latency	01:08:35:48	01:21:37:42	01:21:41:48	01:03:36:10	01:18:17:12	01:21:14:01
Min Latency	00:11:04:12	00:11:04:12	00:11:04:12	00:11:04:12	00:11:04:12	00:11:04:12
Max Latency	31:10:53:00	31:10:53:00	32:10:14:29	20:10:32:01	20:10:32:01	32:10:14:29
Hop Count	1.22	3.13	6.33	1.18	3.23	6.01
Test Case	**Run 3**			**Run 4**		
Metric	**B**	**S**	**P**	**B**	**S**	**P**
Hit Rate	88.63%	96.21%	99.24%	90.15%	93.93%	97.72%
Delivery Cost	4.87	7.62	14.57	4.23	7.53	14.586
Avg Latency	01:23:43:41	02:20:04:19	02:17:33:00	02:03:05:56	02:23:47:10	02:20:50:52
Min Latency	00:11:04:11	00:11:04:11	00:11:04:11	00:11:04:11	00:11:04:11	00:11:04:11
Max Latency	33:15:22:43	33:15:22:43	33:15:22:43	33:15:22:43	33:15:22:43	33:15:22:43
Hop Count	2.57	3.87	6.16	2.73	4.04	6.17

increase in hit rate, which is brought close to 100%, the ideal value for this metric. However, having such a good hit rate comes with certain costs, which is clear when looking at the other metrics in the table. The increase in delivery cost and average hop count are correlated, because if a message passes through more nodes it is transferred more times. These values are highest for the popularity version, but given the size of a message in an environment such as the one from our trace experiment, they are not so significant. On the other hand, the average latency grows by a maximum of about 14 hours, which may prove to be too big of a difference for such an algorithm to be feasible. Nevertheless, by analyzing the maximum latency from the table, we observe that it increases by an order of up to approximately 12 days. The explanation is that there are some nodes that very rarely interact with other internal nodes from the network, so the opportunity of delivering data to them comes at very large time intervals, thus increasing the average latency. By eliminating such nodes from the simulation, the latency values fall into acceptable ranges. Another way of solving this issue is to use the external nodes as message carriers when running the three 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 to even these remote nodes in a timely manner greatly increases. It should also be noted that the scenario presented here assumes that nodes send messages to random destinations. However, in reality there is a greater chance that a node will send data to someone from its social circle or community.

6 Conclusions and Future Work

We have presented a social tracing experiment that took place at the Politehnica University of Bucharest in the winter of 2011, with the purpose of gathering contact information between mobile devices participating in an opportunistic network. We analyzed the traces and showed that the contact and inter-contact times follow approximate power law functions. We applied a community detection algorithm on the traces and compared the results obtained with the social network. The conclusion was that nodes with more social links belong to more communities from the perspective of k-CLIQUE and that the social and logical grouping of nodes are in direct correlation with their interactions.

We have also shown that, by including knowledge about the social relationships between nodes in an opportunistic routing algorithm, the hit rate can be significantly increased at low latency and hop count costs under certain conditions. This has been proven on the trace presented in the paper, by comparing it to the distributed implementation of BUBBLE Rap, DiBuBB. Further tests to cement these affirmations will be performed, where nodes will send data to nodes in their own community or social circle with a higher probability. As future work, we plan to implement an opportunistic data dissemination algorithm of our own, specialized for scenarios like the one presented in this paper (i.e. an academic environment). 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.

Acknowledgement. This work was partially supported by project "ERRIC - Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1", ID: 264207, and 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 cofunded 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. McNett, M., Voelker, G.M.: Access and mobility of wireless PDA users. SIGMOBILE Mob. Comput. Commun. Rev. 7, 55–57 (2003)
2. Henderson, T., Kotz, D., Abyzov, I.: The changing usage of a mature campus-wide wireless network. In: Proc. of the 10th Annual Int. Conf. on Mobile Computing and Networking, MobiCom 2004, pp. 187–201. ACM, New York (2004)
3. Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J., Diot, C.: Pocket switched networks and human mobility in conference environments. In: Proc. of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking, WDTN 2005, pp. 244–251. ACM, New York (2005)

4. CRAWDAD, http://crawdad.cs.dartmouth.edu/
5. Musolesi, M., Mascolo, C.: Designing mobility models based on social network theory. SIGMOBILE Mob. Comput. Commun. Rev. 11, 59–70 (2007)
6. Boldrini, C., Passarella, A.: HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships. Comput. Commun. 33, 1056–1074 (2010)
7. Karamshuk, D., Boldrini, C., Conti, M., Passarella, A.: Human mobility models for opportunistic networks. IEEE Comm. Magazine 49(12), 157–165 (2011)
8. Conti, M., Giordano, S., May, M., Passarella, A.: From opportunistic networks to opportunistic computing. Comm. Mag. 48, 126–139 (2010)
9. Hui, P., Crowcroft, J., Yoneki, E.: BUBBLE Rap: social-based forwarding in delay tolerant networks. In: Proc. of the 9th ACM Int. Symp. on Mobile ad Hoc Networking and Computing, MobiHoc 2008, pp. 241–250. ACM, New York (2008)
10. Nguyen, H.A., Giordano, S., Puiatti, A.: Probabilistic routing protocol for intermittently connected mobile ad hoc network (propicman). In: 2007 IEEE Int. Symp. on a World of Wireless Mobile and Multimedia Networks, pp. 1–6 (2007)
11. Boldrini, C., Conti, M., Jacopini, J., Passarella, A.: HiBOp: a History Based Routing Protocol for Opportunistic Networks. In: IEEE Int. Symp. on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2007, pp. 1–12 (2007)
12. Yoneki, E., Hui, P., Chan, S., Crowcroft, J.: A socio-aware overlay for publish/subscribe communication in delay tolerant networks. In: Proc. of the 10th ACM Symp. on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, MSWiM 2007, pp. 225–234. ACM, New York (2007)
13. Lenders, V., May, M., Karlsson, G., Wacha, C.: Wireless ad hoc podcasting. SIGMOBILE Mob. Comput. Commun. Rev. 12, 65–67 (2008)
14. Pop, F.: A fault tolerant decentralized scheduling in large scale distributed systems. In: Antonopoulos, N., Exarchakos, G., Li, M., Liotta, A. (eds.) Handbook of Research on P2P and Grid Systems for Service-Oriented Computing: Models, Methodologies and Applications. Info. Science Ref., pp. 566–589 (2010)
15. Boldrini, C., Conti, M., Passarella, A.: Exploiting users' social relations to forward data in opportunistic networks: The HiBOp solution. Pervasive Mob. Comput. 4, 633–657 (2008)
16. Ciobanu, R., Dobre, C.: Data dissemination in opportunistic networks. In: 18th Int. Conf. on Control Systems and Computer Science, CSCS-18, pp. 529–536 (2011)
17. Bigwood, G., Rehunathan, D., Bateman, M., Henderson, T., Bhatti, S.: Exploiting self-reported social networks for routing in ubiquitous computing environments. In: Proc. of the 2008 IEEE Int. Conf. on Wireless & Mobile Computing, Networking & Comm., pp. 484–489. IEEE Computer Society, Washington, USA (2008)
18. Boldrini, C., Conti, M., Delmastro, F., Passarella, A.: Context- and social-aware middleware for opportunistic networks. J. Netw. Comput. Appl. 33(5), 525–541 (2010)
19. Social Tracer, http://code.google.com/p/social-tracer/
20. Ferro, E., Potorti, F.: Bluetooth and Wi-Fi wireless protocols: a survey and a comparison. IEEE Wireless Comm. 12(1), 12–26 (2005)
21. Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding. Technical report, University of Cambridge, Computer Lab (2005)
22. Hui, P., Yoneki, E., Chan, S.Y., Crowcroft, J.: Distributed community detection in delay tolerant networks. In: Proc. of 2nd ACM/IEEE Inter. Workshop on Mobility in the Evolving Internet Architecture, MobiArch 2007, pp. 7:1–7:8. ACM, New York (2007)