Social-Awareness in Opportunistic Networking

Radu-Ioan Ciobanu and Ciprian Dobre

Faculty of Automatic Control and Computers,
University Politehnica of Bucharest,
313 Splaiul Independentei, Bucharest, Romania, 060042
E-mail: radu.ciobanu@cti.pub.ro
E-mail: ciprian.dobre@cs.pub.ro (corresponding author)

Abstract: Since uninterrupted connectivity has become such an important part of everyday life, the amount of energy consumed by the various devices used has increased considerably. We propose a way to limit this consumption by employing opportunistic networks, which are mainly composed of mobile devices that have no need for a static network infrastructure. They communicate when in range, using a store-carry-and-forward paradigm. We believe that opportunistic networks can be deployed in closed environments if there is a certainty that messages sent in the network eventually reach their destination. Therefore, we propose the addition of social data to existing opportunistic routing algorithms. We investigate our approach using two traces collected in different environments and we present an analysis of our findings. Most importantly, we show that by adding knowledge such as social links between participants, the performance of the opportunistic network can be improved.

Keywords: Opportunistic networking, online social data, tracing, power reduction, energy efficiency

Reference to this paper should be made as follows: Radu-Ioan Ciobanu, Ciprian Dobre. (2013) 'Social-Awareness in Opportunistic Networking', *Int. J.* , Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Radu-Ioan Ciobanu is currently a teaching assistant and PhD student within the Computer Science Department, in Politehnica University of Bucharest, Romania. He received his BSc in 2010 and MSc in 2012 at the same university. His general interests regard mobile networks and algorithms. His current research is focused on opportunistic networking, social interaction and context-awareness.

Ciprian Dobre received his PhD in Computer Science at the University Politehnica of Bucharest in 2008. He has scientific and scholarly contributions in the field of large scale distributed systems concerning monitoring (MonALISA), data services (PRO, DataCloud@Work), high-speed networking (VINCI, FDT), large scale application development (EGEE III, SEE-GRID-SCI), evaluation using modeling and simulation (MONARC 2, VNSim). His research activities were awarded with the Innovations in Networking Award for Experimental Applications in 2008 by the Corporation for Education Network Initiatives (CENIC). He supervised the PUB Team in projects such as "CAPIM - Context-Aware Platform using Integrated Mobile Services", and "TRANSYS – Models and Techniques for Traffic Optimizing in Urban

Environments”.

This paper is a revised and expanded version of a paper entitled “Social Aspects to Support Opportunistic Networks in an Academic Environment” presented at ADHOC-NOW, 9-11 July 2012, Belgrade, Serbia.

1 Introduction

Nowadays, connectivity has become an important part of everyday life. Whether it is through a personal computer, a tablet, a smartphone, or even a TV, communication is all around us. Many people need to be connected at all times, which has led to a great increase in energy consumption in the past years. Researchers have been trying to reduce this consumption by means of green communication and computing, and in this paper we propose a way to achieve this through opportunistic networking.

An opportunistic network is a particular type of delay-tolerant network (DTN) that is generally composed of mobile devices, where users participate at propagating the information. Human-carried mobile devices communicate with each other in a store-carry-and-forward fashion, without any infrastructure. The lack of an infrastructure means that the need for static access points that can facilitate the communication between two nodes is gone, the mobile devices themselves being the facilitators. This leads to a reduction in energy consumption because routers and switches are no longer necessary, since the communication is performed point-to-point when two devices are in wireless or Bluetooth range. The store-carry-and-forward paradigm entails that a mobile node stores data for a while (usually in the form of messages), carries it around until it reaches the intended destination (or a node that has a higher chance of reaching the destination than the carrier), and finally forwards it when an encounter occurs. In opportunistic networks, disconnections and highly variable delays caused by human mobility are the norm. The solution consists of dynamically building routes, since contacts between nodes are viewed as opportunities 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. Traditionally, delays are seen as networking problems caused by connectivity interruptions. However, in DTNs they are the rule: messages are deliberately delayed and offloaded to alternative communication routes in order to relieve wireless and mobile networks of data traffic. Moreover, using opportunistic networking can also lead to an increase in battery life for mobile devices, since the point-to-point communication between two nodes is done through Bluetooth or WiFi, both of which consume less power than 3G.

In an opportunistic network, members are people that carry mobile devices. In real life, people are organized into social 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, and have various social preferences (Yogo et al., 2011). This is why we propose and investigate the use of community organization and social preferences 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 networking (Boldrini et al., 2008; Boldrini and Passarella, 2010; Toral et al., 2011). Here we show how social information can improve the hit rate of an opportunistic network. We believe that social networks are more accurate than existing community detection algorithms in terms of human relationships. Achieving a high hit rate through our simulations signifies that opportunistic networking might be applicable in real life scenarios, which can lead to the relinquishing of wired and wireless network interfaces altogether in certain situations.

The properties of dissemination algorithms for opportunistic networks can best be investigated using real life traces. Such traces capture accurate information about the mobility patterns of people carrying mobile devices. We present the use of such traces, and investigate our proposed socio-aware networking algorithms using the data collected in a tracing experiment performed between November and December 2011 at our faculty's campus. The validity of our assumptions is further analyzed 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 social information that may be relevant in designing a routing or dissemination algorithm for opportunistic networks.

In an earlier version of this paper, we presented a mobility trace taken at our faculty, and attempted to improve the existing BUBBLE Rap opportunistic routing algorithm by adding social information. In this article, we expand on the proposed solution by adding another two BUBBLE Rap improved versions that use social information. Moreover, we analyze our proposed trace in more detail and test our algorithm for different scenarios, as well as on a different type of mobility trace.

The rest of the paper is structured as follows. Section 2 presents an overview of related work in the field of tracing, routing and dissemination in opportunistic networking. In Sect. 3 we present a socio-aware modified BUBBLE Rap algorithm that takes advantage of social network knowledge. Section 4 describes our social tracing experiment and the results obtained in analyzing the output. Section 5 presents a different tracing scenario used to further evaluate the efficiency of our approach. Section 6 describes the experiments performed and shows the results obtained. Finally, Sect. 7 concludes and presents future work.

2 Related Work

The main goal of routing and dissemination algorithms for opportunistic networks is deciding whether an encountered node is the most suitable next hop for a certain message. If the data is delivered to a node that will not encounter the desired destination in the near future, the delivery latency can increase drastically. Moreover, a message relayed to an unsuitable node may end up never reaching the intended destination at all. Furthermore, sending data to the wrong nodes can lead to node or network congestion, which in turns leads to an increase in battery consumption. If a mobile device has to perform too many send/receive operations, it may end up consuming its battery very quickly.

A taxonomy for opportunistic data dissemination algorithms considers splitting such algorithms into four large categories (Ciobanu and Dobre, 2011). The first category deals with the infrastructure of the network, meaning the way the network is organized into an overlay for the nodes. Another category splits routing and dissemination algorithms according to the characteristics of the 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 nodes in such networks are mobile devices carried by humans, which interact with each other according to social relationships. They are split into communities, and users belonging to the same community have a higher chance of interacting.

There are several papers that propose dissemination algorithms for opportunistic networking. Yoneki et al. (2007) propose Socio-Aware Overlay, an algorithm that creates an overlay for an opportunistic network with pub/sub communication. The overlay is composed of nodes with high values of centrality (i.e. that are on the most shortest paths between any two other nodes), in order for the chosen broker node to maintain a higher message delivery rate. The algorithm is socially-aware, having its own community detection methods. The authors also propose two algorithms for distributed community detection, named Simple and k-CLIQUE. Another opportunistic dissemination algorithm that enables podcasting (distribution of content) using opportunistic contacts whenever mobile devices are in wireless communication range was proposed by Lenders et al. (2008). When two nodes are within range of each other, they associate together and start soliciting episodes from the channels they are subscribed to. Boldrini et al. (2008) propose a dissemination technique called ContentPlace, that 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 and selects those objects that maximize the local utility of its cache. This idea was also extended by Ciobanu et al. (2011), but the utilities are computed not only based on social information, but also on the history of encounters and context information (such as battery life).

A thorough review of opportunistic networking was performed by Conti et al. (2010). 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. Several well-known opportunistic forwarding algorithms are also presented, such as BUBBLE Rap (Hui et al., 2008), PROPICMAN (Nguyen et al., 2007) and HIBOp (Boldrini et al., 2007). Social networking in the context of opportunistic communications has been studied mainly to build efficient forwarding algorithms. Studies such as the ones performed by Hui et al. (2008) and Mtibaa et al. (2008) propose the use of various properties of the social graph such as node centrality and community structures in order to make efficient forwarding decisions. However, previous solutions use mainly machine learning techniques and past observations of the user mobility habits to predict such properties. Here we propose the addition of online social networking services to maximize message delivery. We propose an approach to combine this data with learned information about the user's social community structure and node centrality (i.e. how many shortest paths between two other nodes pass through the current node). We argue that opportunistic dissemination algorithms should greatly benefit from the introduction of social components into the dissemination process, and evaluate this idea in two different scenarios. Both experiments show improvements in performance.

Similar to the approach proposed in this paper, the addition of social network information to opportunistic routing has been studied by Bigwood et al. (2008). 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.

The evaluation of opportunistic dissemination algorithms can be done in two ways. One way of testing an opportunistic network is to use mobility models. Several such models were proposed in recent years. In the waypoint model (Broch et al., 1998) nodes move freely and without restrictions. However, such a model does not have a good similarity to real life patterns, where users are grouped into communities and social circles. One example of a mobility model that considers the social aspect of human movement is CMM (Musolesi and Mascolo, 2007), which models the degree of social interaction between two people using a value between 0 and 1, and isolates highly connected sets of nodes into social groups based on their centrality. HCMM, or the Home-Cell Mobility Model (Boldrini and Passarella, 2010), takes this approach one step further by assuming that nodes in an opportunistic network are not driven only by the social relationships between them, but also by the attraction of physical locations. Thus, each community has a home cell. This mobility model is based on the caveman model (Watts, 1999) 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. According to this model, the attraction of an external cell is computed based on the relationships with nodes that have their home in that cell. When a node reaches a cell that is not its own home community cell, it stays there with a probability p_e, and returns to its home cell with the probability $1 - p_e$.

More recently, many authors used mobility traces to evaluate the properties of opportunistic data dissemination algorithms. Traces such as the ones presented in this paper can accurately capture various real life situations. Experiments designed on top of emulation of such traces use real life mobility patterns of users, which sometimes show properties not captured by any mathematical mobility model. McNett and Voelker (2003) describe an experiment performed at UCSD designed to analyze the mobility patterns of users equipped with wireless PDAs. The experiment had 275 participants and ran for 11 weeks between September and December 2002. The devices used WiFi networking to search for access points located on college grounds. A trace based on WiFi was also recorded at Dartmouth (Henderson et al., 2004) between November 2003 and February 2004. The participating devices in this trace were laptop computers owned by students. Another set of traces, performed by Hui et al. (2005), used the iMote devices from Intel to perform a Bluetooth scan of five seconds at every two minutes. The authors conducted two experiments: the first one had 17 researchers and interns from Intel Research Cambridge participating, while the participants in the second experiment were 18 members of a research group at the University of Cambridge Computer Lab. While the traces collected from these experiments are similar to the one presented in this paper in terms of number of participants, the duration differs greatly, as the iMote traces were performed for 3 and 5 days (compared to 35 in our case, as presented in Sect. 4). Another difference between our trace and the ones taken by Hui et al. (2005) 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. 4 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[a], a community resource for archiving wireless data. In this paper we evaluate the proposed dissemination algorithms using two different traces.

3 Opportunistic Routing

The results presented in this paper have been obtained by improving the distributed BUBBLE Rap algorithm (DiBuBB) proposed by Hui et al. (2008) so that it takes into account social relationships between participants. Therefore, in this section we first present the DiBuBB algorithm, followed by the proposed improvements that incorporate social data into the opportunistic approach.

3.1 BUBBLE Rap and DiBuBB

BUBBLE Rap 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 similar to its role 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 it 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 (the global centrality), or the number of times a node can be found on the shortest path between two other community nodes (the local centrality).

In order for encountering nodes to be aware of the social relationship between them, each of them must have a unique ID and their own social network information, represented as a list of connected nodes. Thus, when two nodes meet, they advertise their own IDs. By receiving the encountered node's ID, a network participant can check its social network data and decide if the two nodes belong to the same community or not. The ID of a node can be any uniquely identifying number, such as the IMEI number for smartphones or tablets, or the MAC address for any type of network device. By using the MAC address, there wouldn't even be a need for the advertising message, since mobile devices are found by broadcast beacons with regard to their MAC. If safety and security are desired, we propose obtaining the ID by hashing the MAC address.

Community detection is done using k-CLIQUE (Hui et al., 2007), an algorithm that dynamically detects the community of a node by analyzing its encounters with other

[a]http://www.crawdad.org/

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 traces, applying a flooding algorithm and then computing the number of times a node acts as a relay on a shortest path. Newman (2004) presents several other methods for detecting communities, ranging from classic methods (e.g. the spectral bisection method or Kernighan-Lin) to sociological methods (e.g. the simple or the complete linkage methods), and finally to recent approaches, like the Girvan-Newman algorithm.

However, this implementation of BUBBLE Rap proved to be unfeasible in real life, since the centrality values computation described above can only be performed retroactively. Therefore, a distributed version entitled DiBuBB was also proposed by Hui et al. (2008). 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 accumulated values. For both these methods, the global centrality takes into account all the nodes in the network, while the local centrality only counts nodes that are in the same community as the current node. For our implementation, we used the C-window algorithm since it led to better results in terms of hit rate.

3.2 Social Connections in DiBuBB

We believe that using information about the social relationships between members of an opportunistic network (such as Facebook data) can increase the effectiveness of routing in terms of hit rate. Therefore, we propose and present here several modified versions of the distributed BUBBLE Rap algorithm that take advantage of the social connections matrix. They are entitled Social, Max, Popularity and Popularity Squared and are described in more detail in the following subsections.

3.2.1 Social

When two nodes meet in DiBuBB, they first check if they belong to the same community according to k-CLIQUE, and if they do they compare their local centralities. 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. We believe that social relationships offer a closer approximation of human relationships than the results of existing community detection algorithms. Furthermore, they reduce the computation done at every contact, since there is no need to recompute the community every time a node is encountered.

3.2.2 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 by dividing them to the maximum number of nodes in the network, so they are between 0 and 1.

3.2.3 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 then multiplied by w_1 and w_2, which are weight values following the condition $w_1 + w_2 = 1$. The reason weights are used is to control the importance of each of the two components.

3.2.4 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.

3.3 Applicability

As a real-life scenario where applying an opportunistic network would be suitable, we chose an academic environment such as the campus of a faculty, where students and teachers can perform energy-aware communication by not using the static infrastructure of the faculty. Furthermore, the mobile devices used for communication can save battery life by using WiFi or Bluetooth for point-to-point communication, instead of 3G, which consumes more power.

We believe that a platform for supporting generic context-aware mobile applications, such as CAPIM (Dobre et al., 2011), can fully benefit from our solution. This would happen because CAPIM already uses various mobile phone sensors (such as GPS) for collecting context information, so an important part of the phone's battery life is used for gathering the context data. Thus, using opportunistic networking for the communication layer can help limit the battery consumption and extend the life of the CAPIM-capable devices.

However, when testing, we not only show the results obtained on a mobility trace taken in an academic environment, but also for a different scenario taken in a larger and more open space. The purpose is to prove that our solution performs well for other types of situations, and in future work we plan on going into even more detail regarding such scenarios.

4 UPB 2011 Trace

In order to obtain trace information regarding the mobility of the members of a faculty to be used for demonstrating the efficiency of the proposed DiBuBB changes, a tracing experiment has been performed at the University Politehnica of Bucharest in the autumn-winter season of 2011. This section presents the setup and additional details about this experiment and the resulting trace, entitled UPB 2011.

4.1 Social Tracer

Tracing in this experiment was performed using an Android application entitled Social Tracer[b]. The participants were asked to run the application whenever they were in the faculty grounds, as we were interested in collecting data about the mobility and social traces in an academic environment. Social Tracer sent regular Bluetooth discovery messages at certain intervals, looking for any type of device that had its Bluetooth on. These included the other participants in the experiment, as well as phones, laptops or other types of mobile devices in range. The reason Bluetooth was preferred to WiFi was mainly the battery use (Ferro and Potorti, 2005). For example, in 4 hours of running the Social Tracer application on a Samsung I9000 Galaxy S with discovery messages sent every 5 minutes, it used approximately 10% of the battery's energy. The period between two successive Bluetooth discovery invocations could be set from the application, ranging from 1 to 30 minutes.

When encountering another Bluetooth device, the Social Tracer application logged data containing its address, name and timestamp. The address and name were used to uniquely identify devices and the timestamp was used for gathering contact data. Data logged was stored in the device's external 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 from CRAWDAD. Successive encounters between the same pair of devices within a certain time interval (chosen based on the discovery interval) were considered as continuous contacts, also taking into consideration loss of packets due to network congestion or low range of Bluetooth.

[a]http://code.google.com/p/social-tracer/

4.2 Experimental Setup

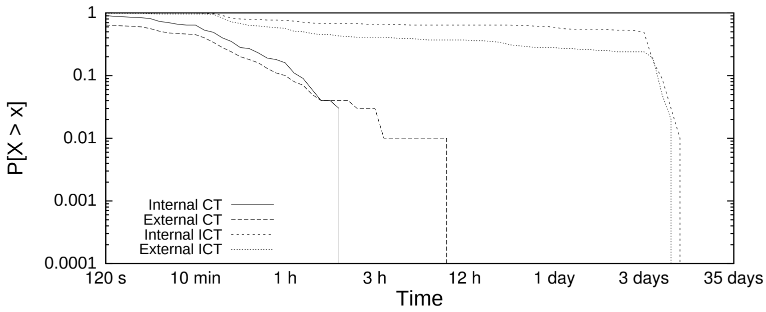
The experiment was performed for a period of 35 days at the campus of University Politehnica of Bucharest between November 18 and December 22, 2011. There were a total of 22 participants, chosen to be as varied as possible in order to obtain a better approximation of mobility in a real academic environment. 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. 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.

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. 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 also 655 different external devices sighted during the course of the experiment, for a total of 1127 different sightings. 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.

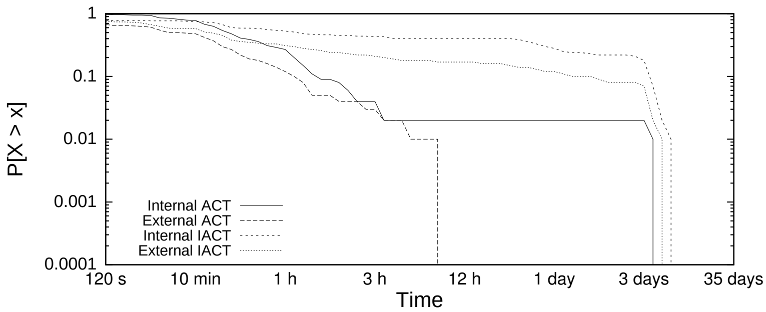
4.3 Contact and Inter-Contact Times

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(a) 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 by Hui et al. (2005), 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 is relevant when discussing the bandwidth required to send data packets between the nodes in an opportunistic network, because it shows 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



(a) CT and ICT

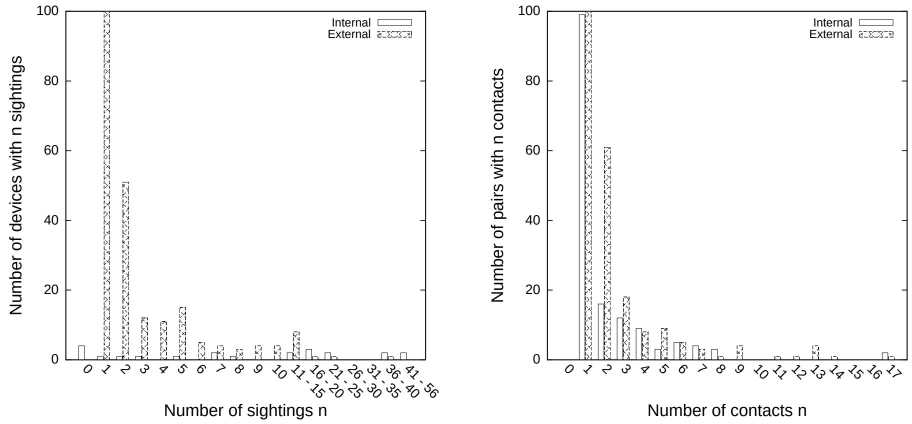


(b) ACT and IACT

Figure 1: Probability distribution of contact (CT), inter-contact (ICT), any-contact (ACT) and inter-any-contact (IACT) times.

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 described previously, 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 by Chaintreau et al. (2005) 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 1(b) shows contact and inter-contact times for encounters with any nodes. Thus, contact time in this case, called any-contact time by Hui et al. (2005), represents the time in which any internal or external node is in range with the current observer (instead of a single node), while the inter-any-contact time is the time when the current device does not see any kind of Bluetooth device in range. These any-contact times are greater than regular contact times, but the shape of the distribution continues to be a power law function. A conclusion that can be drawn from these charts is that, as also observed by Hui et al. (2005), durations of any-contact times are higher and intervals between contacts are lower than for regular contact times, so if a node wants to perform a multicast or to publish an object in a publish/subscribe environment, it has a better chance of being able to do so than it has if it performs a unicast transmission. This is



(a) Sightings of a device. (b) Contacts between pairs of devices.

Figure 2: Distribution of sightings of a device and contacts between pairs of devices.

one of the main reasons opportunistic networking is veering towards information-centric organization (namely publish/subscribe), where the network flow is based on channels. Nodes subscribe to channels, and when data is published at those channels, it is disseminated in the network until it reaches all subscribed nodes.

4.4 Contact Distribution

Figure 2(a) shows the distribution of the number of times a node (internal or external) was sighted by a device participating in the experiment. Axis X shows a number of sightings n, while axis Y shows how many devices have had n sightings. 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 2(b) 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.

Figure 3 shows the distribution of the number of nodes encountered by the internal devices participating in the experiment. Axis X shows a number of devices n, while axis Y shows how many devices have encountered n nodes during the experiment. The maximum number of internal devices spotted by a participant is 17, whereas some participants have only encountered external nodes. It is also clear from the figure that most internal devices have been in contact with between 10 and 15 other internal devices. As shown previously, the total number of external devices encountered during the 35 days of the experiment is far greater than that of the internal devices, and this can

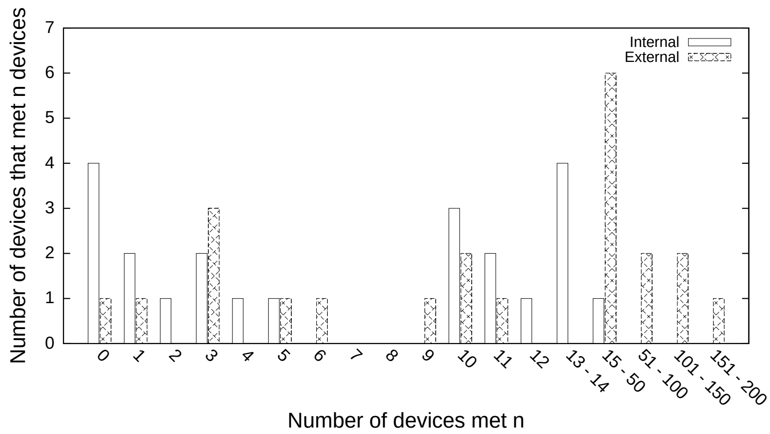


Figure 3: Distribution of the number of nodes encountered by experiment participants.

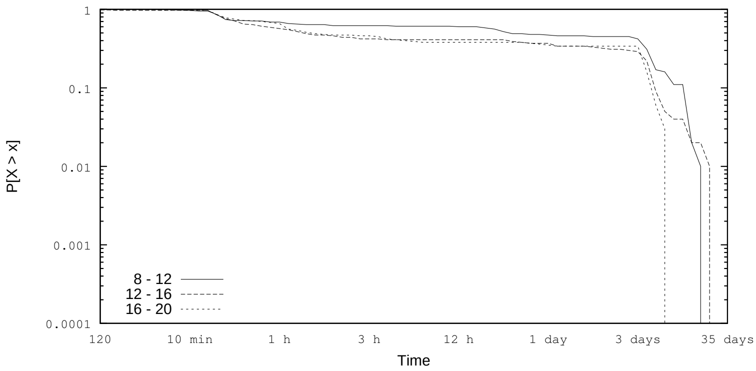
easily be observed by looking at Fig. 3. There are 6 participants that have encountered between 15 and 50 external devices and 5 that have been in contact with more than 50 external nodes. The maximum number of different external devices spotted by a single participant is 197.

4.5 Time of Day Dependence

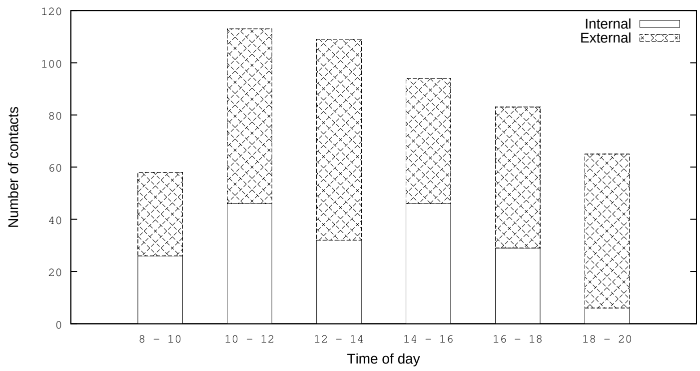
This subsection analyzes the UPB 2011 trace in terms of time of day dependence. This is important in deciding various parameters when emulating the trace, such as times when messages are sent or when nodes may become inactive.

Figure 4(a) shows the distribution of inter-contact times for both types of devices for three time intervals. They are chosen between 8 AM and 8 PM because that is when students or teachers may be at the faculty, and this experiment in not concerned with what happens in the rest of the day. The 8 AM - 8 PM interval has been split into three parts, corresponding to three main periods of the day: morning (8 AM - 12 PM), noon (12 - 4 PM) and afternoon (4 - 8 PM). As we can see from the figure, the three plots are very similar, following the same approximate power law function. This is different from the conclusions of Hui et al. (2005), where daytime periods had a greater power law coefficient than night periods. This happens because we are only interested in periods when there are classes. If we would split the entire day into three periods, the charts would look very much the same.

Figure 4(b) shows the number of contacts that take place in six two-hour intervals between 8 AM and 8 PM. It can be seen that most contacts (113) happen between 10 AM and 12 PM and the smallest number of contacts in a two-hour interval (58) is recorded between 8 AM and 10 AM. External contacts have a distribution similar to the one for all contacts, which shows that the faculty is populated the most between 10 AM and 2 PM. This can also be explained by the fact that at noon students usually have lunch at the cafeteria, so they meet in a common place. Internal contacts are less evenly distributed, but the intervals with the most encounters are approximately the same.



(a) Probability distribution of inter-contact times for three four-hour time intervals.



(b) Distribution of contacts for six two-hour time intervals.

Figure 4: Time of day dependence.

4.6 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 and HCMM have been proposed and implemented, but a trace-based experimental approach can show the interaction patterns better.

The environment represented by the faculty grounds is already logically organized into communities, namely the groups of students and the teachers or lecturers. At our faculty there are four years for Bachelor students, each split into several groups of about 30 students each. For Master 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. However, because there were only 22 participants in our experiment, we decided that the logical grouping should

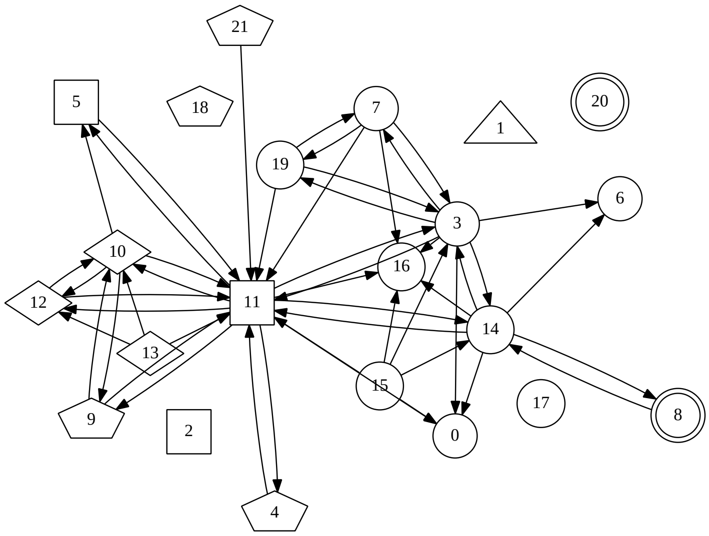


Figure 5: The graph of participants in the experiment as computed by k-CLIQUE. Bachelor students are represented as triangles (first year), circles (third year) and double circles (fourth year), Master students are shown as pentagons (first year) and diamonds (second year), while assistants and lecturers are squares. A directed edge from node A to node B means that A sees B as part of its community.

be done by year instead of group. We applied k-CLIQUE for community detection on the traces collected by Social Tracer. The community graph obtained after applying k-CLIQUE is presented in Fig. 5, along with the logical organization of participants into year groups (which are shown in the figure as having different shapes). 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 in order for the community resulted from k-CLIQUE to be as close as possible to the real life organization given by social relationships and the logical grouping. 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 for our trace.

Because a logical grouping into communities may not always be as straight-forward as in this case, the social relationships between device owners can be taken into account. The social graph of the participants in our experiment, obtained by analyzing Facebook contact information, is shown in Fig. 6. 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 in the

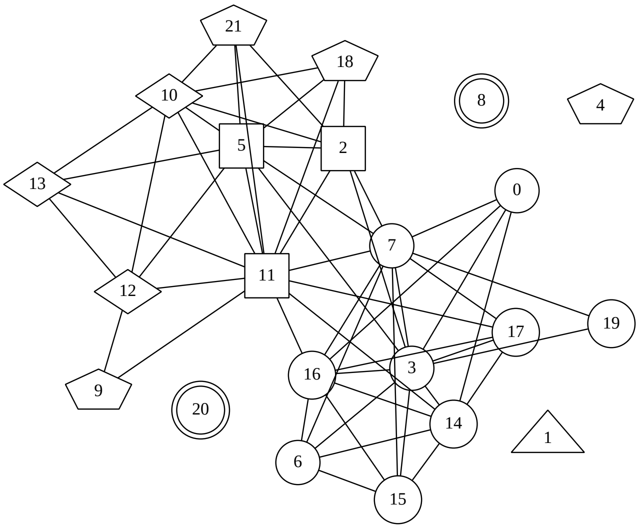


Figure 6: The social graph of participants in the experiment. Year group representation is the same as in Fig. 5. Edges symbolize a social link.

opportunistic network. The similarity value for the social network organization and the logical distribution of participants into years is 83.06%, showing that the social network approximates the real-life relationships between the participants better than k-CLIQUE.

5 St. Andrews Trace

To analyze the efficiency of our proposed algorithm in a situation other than an academic indoor environment, we also tested on a trace collected in an urban tracing experiment by Bigwood et al. (2008).

The St. Andrews trace 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 were able to detect and store information about encounters between each other within a radius of 10 meters, and were programmed to send discovery beacons at every 6.67 seconds. The encounter information, comprised of timestamp and 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 (this was the DSN, as presented in Sect. 2). 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.

6 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 and latency in delivering the messages.

6.1 Experimental Setup

We performed two series of experiments for the two traces presented above. The UPB 2011 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 publicly-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 between two nodes. Because in both of the traces (but especially in ours) there are a large number of external nodes that are only seen once, we only took into consideration for our experiments the internal nodes. These are nodes that are actual participants in the experiment. Moreover, we had no social information about external nodes so the centrality would not have been computed correctly. We believe that this approach would best be suited in a college or faculty (an enclosed space), where social information about each node is known, and achieving good performance results would lead to power reduction since the static network infrastructure would no longer be necessary.

In each set of experiments we applied the base DiBuBB algorithm, along with our four versions presented in the Sect. 3. 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 Sect. 3, 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 our trace 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.

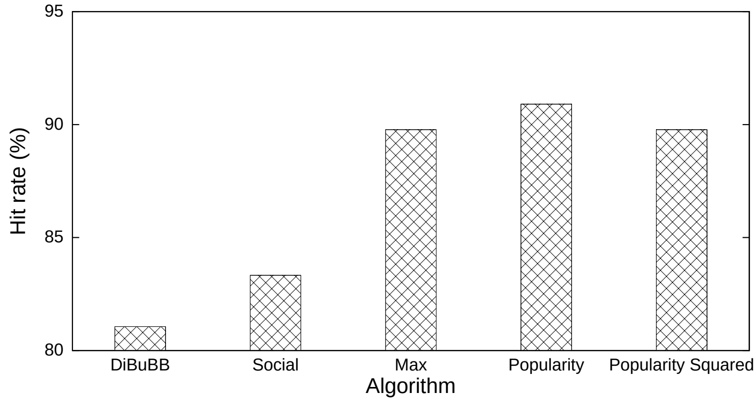
Each node in the experiments sends a number of messages equal to the total number of nodes in the network. All messages are sent at the beginning of a run, following a Zipf distribution with an exponent of 1. Thus, a node sends the largest amount of messages to a node in its own social network (i.e. the value of the distribution for rank 1), then to nodes belonging to the same community according to k-CLIQUE (rank 2) and finally to nodes that are not in relation to it (according to both k-CLIQUE and the social network, with the rank 3 in the Zipf distribution). For example, if there are 22 nodes in the opportunistic network, each node sends 12 messages to nodes in the same social network, 6 messages to nodes in the same k-CLIQUE community, and 4 messages to any other nodes. We chose a Zipf distribution of messages because Adamic and Huberman (2000) have shown that data requests and sends follow power law distributions. 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. We did this because we were interested to highlight the way that messages circulate in the network in tough conditions and how often nodes become congested. When a node sends many messages, the battery drains faster, since it must connect to its neighbors through WiFi and Bluetooth and perform the data transfer. For future work, we plan on varying the size of a node's data memory, and introduce a capability for MobEmu that limits the network bandwidth available when two nodes exchange data.

Unfortunately, there were certain nodes in the UPB 2011 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.

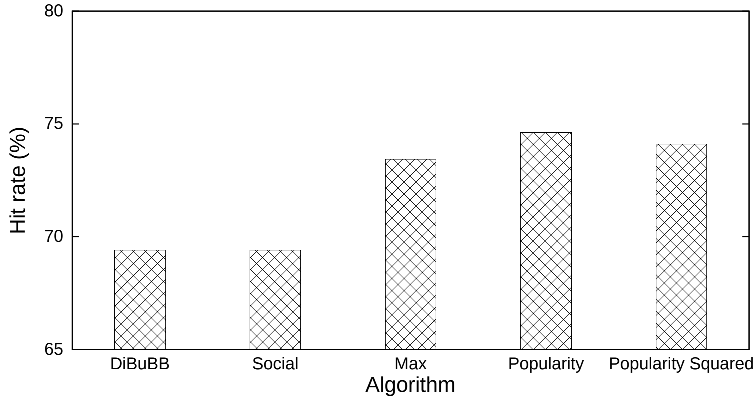
6.2 Results

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%. Another metric we used is the *delivery cost*, represented by the ratio between the total number of messages exchanged 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, or DTN (Warthman, 2003), 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.

Figure 7 shows the hit rate values for UPB 2011 and for St. Andrews. 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 our trace shown in Fig. 7(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%. The weight values used in these tests are $w_1 = 0.9$ and $w_2 = 0.1$ for the Popularity version of DiBuBB and $w_1 = 0.6$ and $w_2 = 0.4$ for Popularity Squared. Figure 7(b) presents the hit rate values for the St. 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 our trace, Max and Popularity Squared bring similar improvements (the weights in this case are $w_1 = 0.3$ and $w_2 = 0.7$ for Popularity and $w_1 = 0.5$ and $w_2 = 0.5$ for Popularity Squared). An interesting conclusion that can be drawn from Fig. 7 is that because the two traces differ in terms of scope, their hit rates also differ significantly. Our 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



(a) UPB 2011

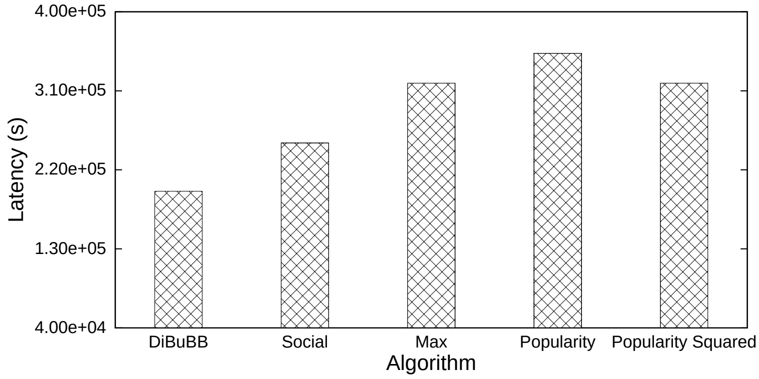


(b) St. Andrews

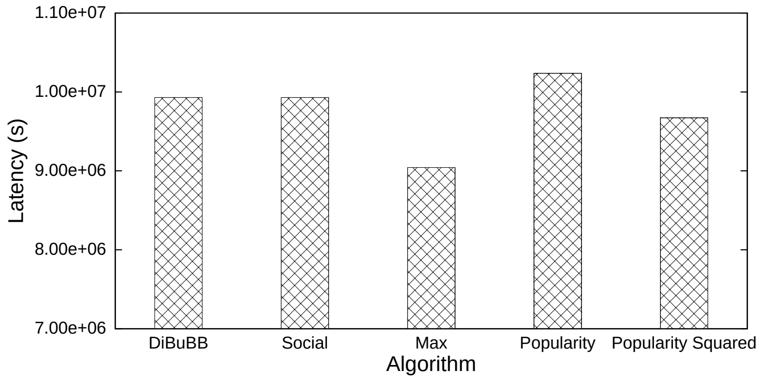
Figure 7: Hit rate for the base, Social, Max, Popularity and Popularity Squared versions of DiBuBB.

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 our 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) than 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 8 presents the latency values obtained by applying the social versions of DiBuBB on the traces. For the UPB 2011 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



(a) UPB 2011

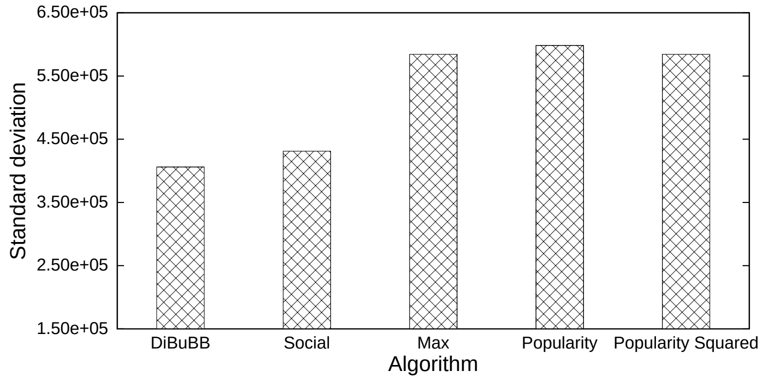


(b) St. Andrews

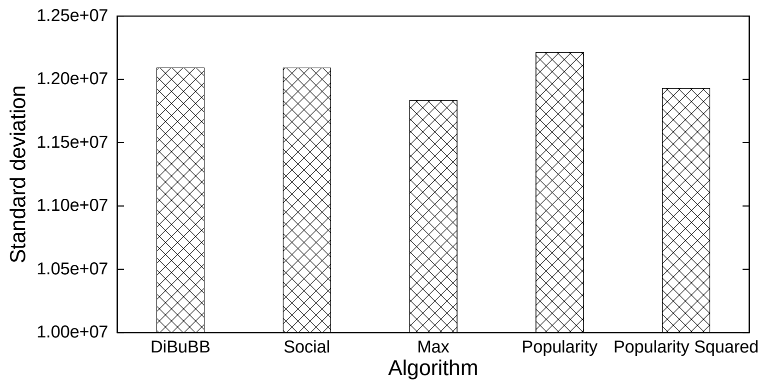
Figure 8: Average latency for the base, Social, Max, Popularity and Popularity Squared versions of DiBuBB.

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 Fig. 9(a) which shows the standard deviation for our 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.

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 our trace. Figure 9(b) shows that standard deviation values are very close to each other as well. An interesting observation regarding Fig. 9 is that the difference in order between latency values for the two traces is about one, which further highlights that an enclosed



(a) UPB 2011



(b) St. Andrews

Figure 9: Standard deviation of latency for the base, Social, Max, Popularity and Popularity Squared versions of DiBuBB.

environment opportunistically disseminates data faster. This leads us to believe that an implementation of an opportunistic network can be performed in such an environment.

Finally, Table 1 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 arise 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 and than the hop count (because the messages reach the destination quickly, but they keep on being sent in the network). The same conclusions as before regarding the differences between the two traces can be drawn, namely that our trace allows for a better and more efficient circulation of messages.

Table 1 Delivery cost and average hop count for the base, Social, Max, Popularity and Popularity Squared versions of DiBuBB.

	Delivery cost		Average hop count	
	UPB 2011	**St. Andrews**	**UPB 2011**	**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

When choosing one of the DiBuBB versions presented in Sect. 3, it is important to know how they can affect the four metrics we used, and this is what we have shown in this section. Popularity is the best version in terms of hit rate, but it has worse delivery cost and hop count than Social, Max, or the original DiBuBB version. In a crowded network, this can lead to congestion, because messages can be sent to nodes at higher rates than they can process. Moreover, sending more messages leads to consuming more battery power. In such a situation, it would be a better idea to use a method that reduces the delivery cost and hop count, even with the drawback of successfully delivering less messages. However, in a sparse network where the contacts are rare, they should be made the most of, so using Popularity or Popularity Squared would be more useful. Given that there are few contacts in such a situation, the battery is not affected as much, since data transmissions are done rarely. The two weight values used by the Popularity and Popularity Squared methods affect the outcome of the algorithm only with regard to the degree in which the social network correctly approximates the interactions between nodes. Thus, 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 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.

7 Conclusions

Recently, different opportunistic routing and dissemination algorithms were proposed and evaluated in various scenarios emulating real life 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 behavior of users, but also their history of contacts coupled with the online social familiarity patterns between them.

We analyzed our approach using two real life traces collected in two different environments. Firstly, we presented a social tracing experiment that took place at the campus of our faculty between November and December of 2011. The purpose of the experiment was to gather contact information between mobile devices that could be used as input for a dissemination algorithm 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 the interactions between nodes. We then evaluated our assumptions in an outdoor urban scenario. We presented an analysis of our findings, highlighting key social and mobility behavior factors that can influence 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. Having hit rates close to 100% and low latencies show that an opportunistic network can be successfully implemented in real life, making the static network infrastructure no longer needed for internal communication. This can in turn lead to a reduction in power consumption, thus increasing the energy efficiency.

Acknowledgements

This work was supported by 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

Adamic, L. A. and Huberman, B. A. (2000), "Power-law distribution of the world wide web", *Science* , Vol. 287, p. 2115.
URL: *http://dx.doi.org/10.1126/science.287.5461.2115a*

Bigwood, G., Rehunathan, D., Bateman, M., Henderson, T. and Bhatti, S. (2008), 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', IEEE Computer Society, Washington, DC, USA, pp. 484–489.
URL: *http://dl.acm.org/citation.cfm?id=1475703.1476536*

Boldrini, C., Conti, M., Jacopini, J. and Passarella, A. (2007), HiBOp: a History Based Routing Protocol for Opportunistic Networks, *in* 'IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007', Helsinki, Finland, pp. 1–12.
URL: *http://dx.doi.org/10.1109/WOWMOM.2007.4351716*

Boldrini, C., Conti, M. and Passarella, A. (2008), 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', MSWiM '08, ACM, New York, NY, USA, pp. 203–210.
URL: *http://doi.acm.org/10.1145/1454503.1454541*

Boldrini, C. and Passarella, A. (2010), "HCMM: Modelling spatial and temporal properties of human mobility driven by users' social relationships", *Comput. Commun.* , Vol. 33, Butterworth-Heinemann, Newton, MA, USA, pp. 1056–1074.
URL: *http://dx.doi.org/10.1016/j.comcom.2010.01.013*

Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C. and Jetcheva, J. (1998), A performance comparison of multi-hop wireless ad hoc network routing protocols, *in* ‘Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking’, MobiCom ’98, ACM, New York, NY, USA, pp. 85–97.
URL: *http://doi.acm.org/10.1145/288235.288256*

Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R. and Scott, J. (2005), Pocket Switched Networks: Real-world mobility and its consequences for opportunistic forwarding, Technical report, University of Cambridge, Computer Lab.

Ciobanu, R. and Dobre, C. (2011), Data dissemination in opportunistic networks, *in* ‘18th Int. Conf. on Control Systems and Computer Science’, CSCS-18, Bucharest, Romania, pp. 529–536.

Ciobanu, R., Dobre, C. and Cristea, V. (2011), A data dissemination algorithm for opportunistic networks, *in* ‘13th Int. Symposium on Symbolic and Numeric Algorithms for Scientific Computing’, SYNASC-2011, Timisoara, Romania, pp. 299–305.

Conti, M., Giordano, S., May, M. and Passarella, A. (2010), “From opportunistic networks to opportunistic computing”, *Comm. Mag.* , Vol. 48, IEEE Press, Piscataway, NJ, USA, pp. 126–139.
URL: *http://dl.acm.org/citation.cfm?id=1866991.1867009*

Dobre, C., Manea, F. and Cristea, V. (2011), Capim: A context-aware platform using integrated mobile services, *in* ‘Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on’, Cluj-Napoca, Romania, pp. 533 –540.

Ferro, E. and Potorti, F. (2005), “Bluetooth and Wi-Fi wireless protocols: a survey and a comparison”, *IEEE Wireless Communications* , Vol. 12, pp. 12–26.
URL: *http://dx.doi.org/10.1109/MWC.2005.1404569*

Henderson, T., Kotz, D. and Abyzov, I. (2004), The changing usage of a mature campus-wide wireless network, *in* ‘Proceedings of the 10th annual international conference on Mobile computing and networking’, MobiCom ’04, ACM, New York, NY, USA, pp. 187–201.
URL: *http://doi.acm.org/10.1145/1023720.1023739*

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

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

Hui, P., Yoneki, E., Chan, S. Y. and Crowcroft, J. (2007), Distributed community detection in delay tolerant networks, *in* ‘Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture’, MobiArch ’07, ACM, New York, NY, USA, pp. 7:1–7:8.
URL: *http://doi.acm.org/10.1145/1366919.1366929*

Lenders, V., May, M., Karlsson, G. and Wacha, C. (2008), “Wireless ad hoc podcasting”, *SIGMOBILE Mob. Comput. Commun. Rev.* , Vol. 12, ACM, New York, NY, USA, pp. 65–67.
URL: *http://doi.acm.org/10.1145/1374512.1374535*

McNett, M. and Voelker, G. M. (2003), “Access and mobility of wireless PDA users”, *SIGMOBILE Mob. Comput. Commun. Rev.* , Vol. 7, ACM, New York, NY, USA, pp. 55–57.
URL: *http://doi.acm.org/10.1145/965732.965744*

Mtibaa, A., Chaintreau, A., LeBrun, J., Oliver, E., Pietilainen, A.-K. and Diot, C. (2008), Are you moved by your social network application?, *in* ‘Proceedings of the first workshop on Online social networks’, WOSN ’08, ACM, New York, NY, USA, pp. 67–72.
URL: *http://doi.acm.org/10.1145/1397735.1397751*

Musolesi, M. and Mascolo, C. (2007), “Designing mobility models based on social network theory”, *SIGMOBILE Mob. Comput. Commun. Rev.* , Vol. 11, ACM, New York, NY, USA, pp. 59–70.
URL: *http://doi.acm.org/10.1145/1317425.1317433*

Newman, M. E. J. (2004), “Detecting community structure in networks”, *The European Physical Journal B - Condensed Matter and Complex Systems* , Vol. 38, Springer Berlin / Heidelberg, pp. 321–330.
URL: *http://dx.doi.org/10.1140/epjb/e2004-00124-y*

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

Toral, S. L., Bessis, N., Martinez-Torres, M. R., Franc, F., Barrero, F. and Xhafa, F. (2011), An exploratory social network analysis of academic research networks, *in* ‘Proceedings of the 2011 Third International Conference on Intelligent Networking and Collaborative Systems’, INCOS ’11, IEEE Computer Society, Washington, DC, USA, pp. 21–26.
URL: *http://dx.doi.org/10.1109/INCoS.2011.49*

Warthman, F. (2003), ‘Delay-tolerant networks (DTNs) - a tutorial’.
URL: *http://www.dtnrg.org/docs/tutorials/warthman-1.1.pdf*

Watts, D. J. (1999), *Small worlds: the dynamics of networks between order and randomness*, Princeton University Press, Princeton, NJ, USA.

Yogo, K., Shinkuma, R., Konishi, T., Itaya, S., Doi, S., Yamada, K. and Takahashi, T. (2011), “Incentive-rewarding mechanism to stimulate activities in social networking services”, *International Journal of Network Management* , Vol. 22, pp. 1–11.
URL: *http://dx.doi.org/10.1002/nem.779*

Yoneki, E., Hui, P., Chan, S. and Crowcroft, J. (2007), 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’, MSWiM ’07, ACM, New York, NY, USA, pp. 225–234.
URL: *http://doi.acm.org/10.1145/1298126.1298166*