JDER: A history-based forwarding scheme for Delay Tolerant Networks using Jaccard Distance and Encountered Ration

Authors: Ciobanu, R.I.[1], Reina, D. G[2], Dobre, C.[1], Toral, S.L.[2], Johnson, P.[3]

[1]*University Politehnica of Bucharest (Romania)*

[2]*University of Seville (Spain)*

[3]*Liverpool John Moores University (UK)*

[1] Faculty of Automatic Controls and Computers, University Politehnica of Bucharest

Spl. Independentei 313, Bucharest (Romania)

email: radu.ciobanu@cti.pub.ro, ciprian.dobre@cs.pub.ro

phone: +40 745 174 359; fax: +40 318 163 028

[2] Escuela Superior de Ingenieros. Avda. Camino de los Descubrimientos, s/n

41092 Sevilla (Spain)

email: toral@esi.us.es, dgutierrezreina@us.es

phone: +34 954 48 12 93; fax: +34 954 48 73 73

[3] School of Engineering, Technology and Maritime Operations

Liverpool L3 3AF (UK)

email: p.johnson@ljmu.ac.uk

phone: +44 151 231 2588; fax: +44 151 207 3460

Corresponding author: S. L. Toral

JDER: **A history-based Forwarding Scheme for Delay Tolerant Networks using Jaccard Distance and Encountered Ration**

Abstract: Delay Tolerant Networks have arisen as a new paradigm of wireless communications in which nodes follow a store-carry-and-forward operation. Unlike other ad hoc networks, mobility of nodes is seen as an interesting feature to deliver information from a source node to a destination node. New forwarding schemes have been proposed to deal with the intermittent communications carried out by nodes in delay tolerant networks. Most forwarding schemes assume that nodes are divided into social communities and the communications are likely to be established between two nodes belonging to the same community. However, the social information is not always available, especially in large environments like cities so it has to be inferred from the history of encounters among nodes. Furthermore, there are cases in which the information has to be widely disseminated throughout the network such as alarm and emergency messages so it has to pass through different communities. In this paper, we propose JDER, a new probabilistic forwarding scheme which guarantees high reachability throughout the network by selecting cut-nodes. JDER is based on two metrics: the history encountered ration and the Jaccard distance, and it has been extensively validated through simulations using 8 different mobility models obtained from real life traces.

Keywords Delay Tolerant Networks, Jaccard Distance, Forwarding Scheme, Opportunistic Networks, History Encountered Ration

1. Introduction

Delay Tolerant Networks (DTNs) are decentralized networks in which nodes cooperatively communicate to transmit application data from a source node to a destination node. Unlike other ad-hoc networks such as Mobile Ad Hoc Networks (MANETs) (Hoebeke et al, 2004) and Wireless Sensor Networks (WSNs) (Akyldiz et al, 2002), network partitioning, disconnections, and high end-to-end delay are dominant factors in DTNs due to dynamic topologies, mobility of nodes and low density conditions. While in MANETs and WSNs mobility of nodes causes undesirable effects such as broken links (Gutiérrez-Reina et al, 2011, 2012a) and low delivery rates deteriorating the performance of such networks, in DTNs mobility is used as an opportunity to deliver information. DTNs are suitable for a large range of applications including disaster scenarios (Martín-Campillo et al, 2012), intelligent transportation systems (Gutiérrez-Reina et al, 2012b; Shin et al., 2012), and pervasive healthcare, among others (Conti and Kumar, 2010b). In DTNs, nodes follow a store-carry and-forward operation (Pelusi et al, 2006) (Conti et al, 2010a). Whenever a given node (source node) has certain information to be transmitted to a destination node, it will opportunistically transmit a copy of such information to one or several intermediate nodes. These intermediate nodes will carry and forward the information again to new intermediate nodes until a copy of the message reaches the destination node. Therefore, the forwarding schemes are responsible for deciding which nodes have to retransmit the information. Consequently, the main goal of the forwarding schemes is to select suitable intermediate nodes that are likely to communicate with the target destination node. Many aspects can be considered to efficiently select a forwarding criterion such as geographical information (Leontiadis and Mascolo, 2007), social information (Boldrini et al, 2010a; Gao et al., 2012), among others. If geographical information is used, nodes need to be equipped with a positioning system like a Global Positioning Systems (GPS) incurring additional cost. However due to the significant variation in the location of both the source and the destination nodes over time, the positioning information becomes inefficient in DTNs. On the other hand, if social information is used, nodes have to be aware of the social relationships among nodes in the network. However, the availability of social information may not always be possible due to the size of the network and/or the lack of access to a real life social network like Facebook or Twitter. In order to overcome the above mentioned drawbacks, forwarding schemes based on the history of encounters among nodes in the networks have been proposed (Boldrini et al 2007; Lindgren et al, 2003). These are aimed to use the past history of encounters among nodes and to select the ones that are likely to communicate with the destination nodes thus restricting the number of retransmissions to low values. The main idea is to predict future encounters among nodes using the past history of encounters, preferences and similarities among nodes (Boldrini et al, 2007; Ciobanu et al, 2013). However, prediction mechanisms have the drawback of being hard to tune successfully depending on the specific behavior of the network, especially when dealing with highly dynamic DTNs. In contrast to the previously proposed algorithms for DTNs, we specifically focus on using the properties of cut-nodes (cut-vertex in social networks theory) in order to make retransmission decisions. Cut-nodes are defined as those nodes whose deletion increases the number of components in the network (De Nooy et al, 2005), and they occupy critical positions for the flow of information within the network as they control the flow from one part to another part of the network. The main advantage of using cut nodes as forwarding nodes in DTNs is that they belong to more than one community so they are vital to message retransmission whether the destination node belongs to the same community or different community to the source node. Therefore, considering the cut-nodes as forwarders will guarantee a high delivery ratio in DTNs. Though identifying cut-nodes may be a trivial task when the whole network is known, this is not the case in DTNs due to their dynamic topology and low connectivity of nodes. In this paper, we propose a method for identifying such nodes by combining two different measures: 1) the history of encountered ration, which finds possible forwarding nodes belonging to the same social network, and 2) the Jaccard distance (explained in more detail in section 3) between two nodes, which measures the dissimilarity between two nodes. By combining these two measures, nodes can identify possible forwarding nodes that belong to the same social network and at the same time, are connected to nodes belonging to other communities.

The main contributions of this paper are:

- To propose a new probabilistic forwarding scheme based on cut nodes for DTNs.
- To evaluate the proposed forwarding scheme using real-life mobility traces.
- To compare the performance of the proposed scheme with that of other forwarding schemes found in the literature.

This paper continues as follows. A brief review of existing forwarding schemes for DTNs is presented in section 2. Section 3 discusses the characteristics of cut-nodes in detail. The proposed forwarding scheme is described in section 4 along with the procedure to calculate the history of encountered ration using the history of encounters among nodes and the Jaccard distance using neighborhood information. Section 5 presents information on how the simulations were conducted and a discussion of the simulation results. Finally, section 6 includes the main conclusions of this paper.

2. Forwarding Schemes in Delay Tolerant Networks

Since mobile devices have become very popular in recent years, DTNs have been the focus of increasing research. The major issues in DTNs concern routing (Tahsin et al, 2011), forwarding and dissemination (Conti et al, 2010a) of information.

A taxonomy for opportunistic data dissemination algorithms has been proposed in (Ciobanu and Dobre, 2011) and describes such techniques using four main categories: network infrastructure (how the network is organized using overlays for nodes), node characteristics (neighbor discovery, content identification, and data exchange), content characteristics and social awareness. Several well-known dissemination algorithms were then analyzed using the proposed taxonomy. The Socio-Aware Overlay (Yoneki et al, 2007) is a routing technique that creates an overlay for a DTN and uses a publish/subscribe communication mechanism. The components of the overlay are broker nodes that have high values of centrality, which means that they can maintain a higher message delivery rate than the regular nodes. In Lenders et al. (2008), the authors propose an opportunistic dissemination algorithm that distributes content through podcasting when mobile devices are within the wireless range of one another. Another forwarding technique for DTNs is ContentPlace (Boldrini et al, 2010a), which uses social information about the participants in the network to select the next hop whenever a contact occurs. The authors assume that the network participants can be grouped together based on the type of content they are interested in, and that their movements and interactions are governed by the strength of their social relationships. This means that two nodes with a strong social connection meet more often and for longer periods of time than two nodes that do not have many things in common. On the other hand, ContentPlace uses a utility function that is applied to every data object (belonging to both the observer and to the observed nodes) when a contact occurs. The objects with the highest utility values are then selected, in order for each node's data memory to be maximized.

One of the most cited and well-known routing algorithms for DTNs is BUBBLE Rap (Hui et al, 2008, 2011), which uses social knowledge about the nodes in the network to deliver messages. There are two important assumptions made by the authors: similar to ContentPlace, they consider that a mobile device carrier's role in society is also true in the network, and that the communities that people form in real life can also be observed in the DTN. Therefore, BUBBLE Rap begins by forwarding data to nodes that are more popular than the current node. The second part of the algorithm identifies members of the destination community and passes the data item to them. Thus, the data moves up on a symbolic hierarchical ranking tree based on a global popularity level, until it reaches a node that is in the same community as the data item's destination. Then, a local ranking is used inside the community up to the point when the destination is reached. A node's popularity is the value of its betweenness centrality, i.e. the number of times a node is on the shortest path between any two nodes in the network. Communities in BUBBLE Rap are dynamically discovered using k-CLIQUE algorithm (Hui et al, 2007). Because the default implementation of BUBBLE Rap was not feasible in real life, the authors also proposed a distributed version entitled DiBuBB, which uses distributed k-CLIQUE for community detection, and a single- or cumulative-window algorithm for centrality computation. The single window (S-window) algorithm computes the 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 cumulative values.

The addition of social network information in opportunistic routing in DTNs has also been studied in Bigwood et al. (2008), where the authors consider two types of networks: a detected social network (DSN) obtained after applying a community detection algorithm such as *k*-CLIQUE, and a self-reported social network (SRSN), created using information such as Facebook connections. When two nodes have a contact, they only exchange data if they are in the same network (either DSN or SRSN), and the authors show that using the SRSN instead of the DSN decreases the delivery cost and produces comparable delivery ratio. Another socially-aware middleware that learns information about the nodes in the network and then uses it to predict the node's future behavior is proposed in Boldrini et al. (2010b).

Aside from using social information when routing, some algorithms attempt to perform a prediction of nodes' behavior in the DTN, in order to better approximate each node's encounters in the near future. When successful, such an algorithm has the benefit of reducing congestion (both at node level, as well as at the network level), since messages are being delivered only to nodes that have high chances of reaching the intended destinations. Such a social prediction-based opportunistic routing algorithm is SPRINT (Ciobanu et al, 2013). At every encounter between two SPRINT nodes, utility values are computed for all the messages carried by the two nodes, and then each node requests the messages with the highest utilities from its standpoint. The utility function has two components, that focus both on the social aspect, as well as on the prediction aspect of data routing and dissemination. A node's future behavior is predicted by approximating its history of past encounters per time unit as a Poisson distribution, and combining it with social information. A Poisson distribution has been shown to function correctly, because of the highly regular behavior of DTN nodes. Since these nodes are mobile devices carried by humans, which have strong habits they follow every day, patterns can easily be seen in their mobility models. Generally, same sets of nodes are encountered at the same time intervals every day. SPRINT uses the Poisson distribution to compute a node's number of contacts per time unit for the following 24 hours. Information about a node's social connections is then used to predict which nodes are going to be encountered (not only how many will there be), and the result is used when computing the utility function. SPRINT is shown to outperform existing algorithms in terms of hit rate, as well as network and node congestion.

However, there are circumstances when information about social connections is not available, or the social network is much too large to be used efficiently. For example, if a DTN would span an entire town, it would be very hard to keep track of all connections between the citizens, especially since social relationships are ever-changing. In such cases, the socially-aware algorithms presented above do not behave efficiently. These algorithms are mostly focused on forwarding messages to popular nodes that are likely to communicate with many other nodes in the network. However, in DTNs as in real social networks, there are critical nodes with less apparent popularity that play an important role in the dissemination process of messages in the network so they have to be found and selected as forwarders in order to guarantee a high delivery ratio. We propose a novel forwarding scheme based on a heuristic mechanism to find such critical nodes (cut nodes) in terms of dissemination reachability. This mechanism uses the history of encounters among nodes to calculate the history of encountered ration and the Jaccard distance.

3. Finding cut-nodes in social networks

Cut-nodes play an important role in the connectivity of different communities in social networks. If cut-nodes are not taken into account by the forwarding schemes, some nodes in the network will never be reached. Let us consider the following example, Figure 1, to illustrate the importance of cut-nodes in the dissemination process.

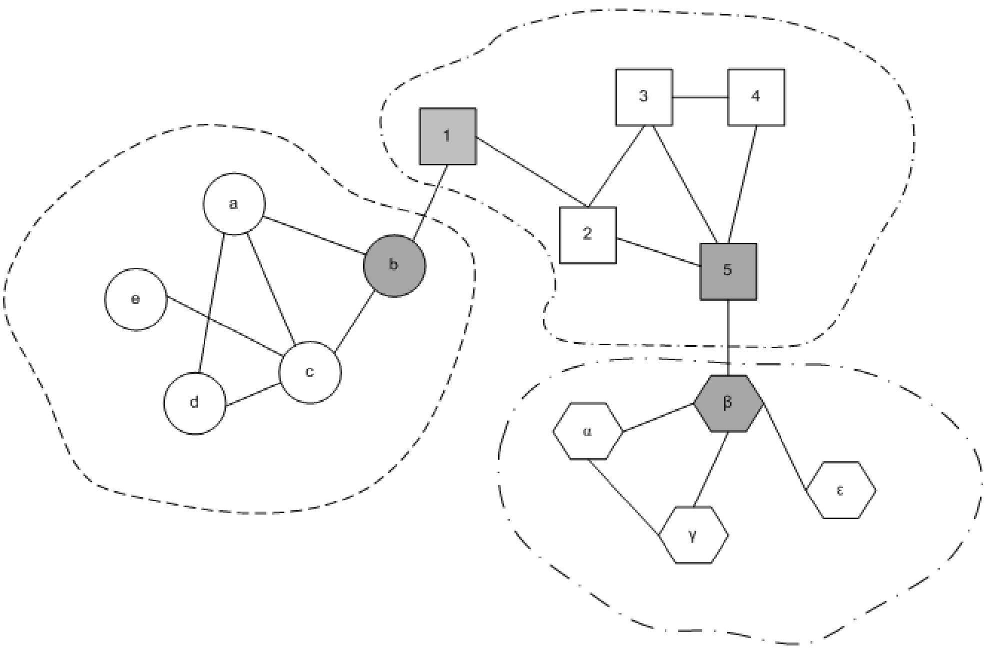

Figure 1. Cut-nodes in social networks

In the above Figure 1 we can distinguish three different communities (circles, squares, and hexagons). The grey nodes represent the cut-nodes and they link different communities. Therefore, if high reachability is desired for reaching nodes outside the source node’s community, the cut-nodes have to be selected as forwarders by the forwarding scheme. Let us consider the scenario where the node ‘a’ in Figure 1 wants to send a message to node 4. It is obvious that the node ‘b’ is a vital connection in order for the message to reach the community of node 4. The question now is which parameters can be used to select node b as a forwarder. The answer is easy if the global picture (complete information) of the network can be had by the selection mechanism. This is not feasible in DTNs because nodes only have partial information on the network (information stored in the memory of nodes). However, certain other characteristics can be used to predict the cut-nodes. First, the node b is in the same community as node a. Thus, nodes included in the same community have to be considered. Secondly, node b is likely to encounter nodes from the community of squares since it is linked to that community through node 1. As a result, node b will be dissimilar to node a in terms of node encounters.

As a summary, cut-nodes are nodes belonging to the same community as the source node but dissimilar to the rest of the nodes in terms of encounters (history of encounters). These features will be used by the proposed forwarding scheme JDER to predict cut-nodes. Notice that if there are two cut-nodes between two communities those two nodes will exhibit the same features. Consequently, the two nodes are suitable to be selected as forwarders. The proposed JDER scheme will choose cut-nodes even when several cut-nodes exist.

4. JDER (Jaccard Distance-Encountered Ration)

To achieve a high delivery ratio in DTNs with a low delivery cost, we propose JDER, a novel probabilistic forwarding scheme. The main objectives of JDER are:

- To select nodes inside the same social network by using the history encountered ration.
- To select dissimilar nodes inside the same community by using the Jaccard distance. It will enable the exploration of new regions of the network through cut-nodes as well as the reduction of the delivery cost for transmitting information inside the same community.

4.1.History encountered ration

In opportunistic networks, nodes are more likely to communicate with nodes inside the same community (Boldrini and Passarella, 2010c). To consider this feature, we proposed the encountered ration E_r as a metric to select nodes inside the same community. Considering the above example illustrated using Figure 1, when the node ‘a’ wants to transmit a message to an intermediate node ‘c’, the encountered ration is calculated as the number of times that the nodes ‘a’ and ‘c’ have encountered each other divided by the total number of encounters that the node ‘a’ has had. So in order to compute the encountered ration, the nodes have to store all the details of their encounters with other nodes in a ‘neighbor table’.

4.2. Jaccard distance

Similarity/dissimilarity coefficients have been used in other scientific areas such as biology, ecology, marketing and psychology among others (Tullos 1997; Härdle and Simar, 2003), as a classification method to analyze similar elements between two lists. Similarity coefficients are aimed to find coincidences between two groups for any or some specific characteristics. The number of shared neighbors between two nodes will be selected as the target characteristic to determine the similarity or dissimilarity between two nodes in an opportunistic network. In general, the proximity or similarity among objects can be described using a matrix *D(n,n)* (Härdle and Simar, 2003). The matrix *D* contains measures of similarity or dissimilarity among the *n* objects. If d_{ij} represents distances between the elements *i* and *j*, and if they measure dissimilarity, then the greater the distance between the objects the more dissimilar they are from each other. On the other hand, if d_{ij} represents similarity between two objects, the opposite is true. By definition, distance and similarity are dual, so if d_{ij} is a measure of distance, then $\mathbf{s_{ij} = max\{d_{ij}\} - d_{ij}}$ is a proximity measure. Depending on the nature of the observations, the similarity or distance measures can be calculated for binary variables or continuous variables. The Euclidean distance is an example of a dissimilarity measure of two objects calculated using two continuous variables such as the coordinates *x* and *y*. In order to measure the similarity or dissimilarity between two lists, the observations are compared in pairs $(\mathbf{x_i}, \mathbf{x_j})$ where $\mathbf{x_i} = (\mathbf{x_{i1}}, \dots, \mathbf{x_{ik}})$, $\mathbf{x_j} = (\mathbf{x_{j1}}, \dots, \mathbf{x_{jk}})$, with $\mathbf{k = 0, \dots, p}$, where *p* is the number of observations, that is nodes in a DTN, and $\mathbf{x_{ik}}$, $\mathbf{x_{jk}} \in \{\mathbf{0}, \mathbf{1}\}$. Four cases are possible:

$$\boldsymbol{x_{ik} = x_{jk} = 1,}$$

$$\boldsymbol{x_{ik} = 0, x_{jk} = 1,}$$

$$\boldsymbol{x_{ik} = 1, x_{jk} = 0,}$$

$$\boldsymbol{x_{ik} = 0, x_{jk} = 0,}$$

In opportunistic networks $\mathbf{x_i}$ and $\mathbf{x_j}$ represent the neighbors of two mobile nodes *I* and *J*. They can easily be identified by the node's ID. The notation detailed below is normally used to represent the level of similarity between two lists. This notation has been extended to two mobile nodes *I* and *J* deployed in an ad hoc manner, see Figure 2 which represents the entry details for two sets *I* and *J*.

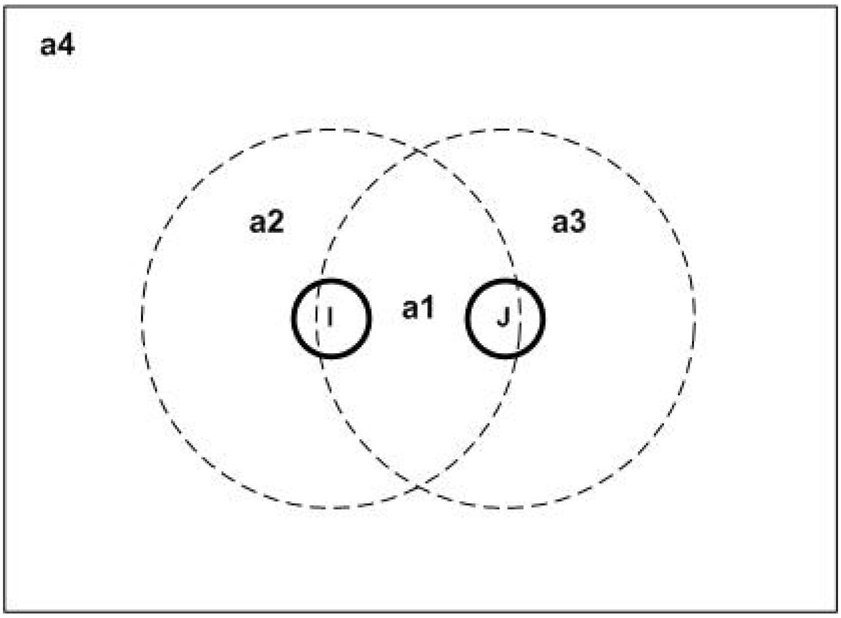

Figure 2. Extension of notation for opportunistic networks

- a1: The number of entries that are common to both lists. In opportunistic networks a1 represents the number of common neighbors for two different nodes so it can be seen as a similarity parameter between the two nodes.
- a2: The number of entries in the first list that are not in the second. Given two nodes *I* and *J*, a2 represents the number of nodes which are neighbors of *I* but are not neighbors of *J*.
- a3: The number of entries in the second list that are not in the first. Unlike a2, a3 represents the number of neighbors which are neighbors of *J* but are not neighbors of *I*.

- a4: The number of entries that do not appear in either list. In an opportunistic network, a4 represents the number of nodes which are not neighbors of either *I* or of *J*.

The above notations for the pairs $(\mathbf{x_i}, \mathbf{x_j})$ can be represented using the following mathematical expressions:

$$\mathbf{a_1} = \sum_{\mathbf{k=1}}^{\mathbf{p}} \mathbf{I}(\mathbf{x_{ik}} = \mathbf{x_{ik}} = \mathbf{1})$$

$$\mathbf{a_2} = \sum_{\mathbf{k=1}}^{\mathbf{p}} \mathbf{I}(\mathbf{x_{ik}} = \mathbf{0}, \mathbf{x_{jk}} = \mathbf{1})$$

$$\mathbf{a_3} = \sum_{\mathbf{k=1}}^{\mathbf{p}} \mathbf{I}(\mathbf{x_{ik}} = \mathbf{1}, \mathbf{x_{jk}} = \mathbf{0})$$

$$\mathbf{a_4} = \sum_{\mathbf{k=1}}^{\mathbf{p}} \mathbf{I}(\mathbf{x_{ik}} = \mathbf{x_{ik}} = \mathbf{0})$$

Depending on the relationship between the values of variables a1 through to a4 the following similarity coefficients can be defined

$$\mathbf{C_{ij}} = \frac{\mathbf{a_1} + \boldsymbol{\delta}\mathbf{a_4}}{\mathbf{a_1} + \boldsymbol{\delta}\mathbf{a_4} + \boldsymbol{\lambda}(\mathbf{a_2} + \mathbf{a_3})}$$

Where $\boldsymbol{\delta}$ and $\boldsymbol{\lambda}$ are weighting factors. Note that, a4 is impossible to obtain from a local point of view of a node in a DTN, and so, $\boldsymbol{\delta} = \mathbf{0}$. For Jaccard coefficient, $\boldsymbol{\delta} = \mathbf{0}$ and $\boldsymbol{\lambda} = \mathbf{1}$ results in the following expression (for further information on other similarity coefficients the readers are referred to Härdle and Simar, (2003))

$$\mathbf{J_{ij}} = \frac{\mathbf{a_1}}{\mathbf{a_1+a_2+a_3}}$$

Where, $\mathbf{J_{ij}} \in [\mathbf{0}, \mathbf{1}]$. Since $\mathbf{J_{ij}}$ is a similarity coefficient, the Jaccard distance can be calculated as $\mathbf{Jd_{ij}} = \mathbf{1} - \mathbf{J_{ij}}$ and $\mathbf{d_{ij}} \in [\mathbf{0}, \mathbf{1}]$. Notice that the use of similarity coefficients to adapt forwarding decisions is not new (Boldrini et al, 2007). However, they have not been exploited in the context of opportunistic networks. In contrast to Boldrini et al. (2007), in JDER the dissimilarity among nodes is used to explore new regions by forwarding information to cut-nodes which are critical in order to guarantee high reachability throughout the network. The benefits of using a similarity-based algorithm, as opposed to a socially-aware or prediction-based algorithm, is that the assumptions it is based on hold regardless of the external conditions. For socially-aware techniques, information about social connections between nodes may not be available, or may prove to have too much overhead if there are many nodes in the DTN. Prediction-based solutions are especially tricky, since they only fit certain types of DTN, where the nodes act in patterns with a certain regularity. Nonetheless, the Jaccard distance should also be adapted to the features of DTNs. The low connectivity of opportunistic networks makes it impossible for a given node to have simultaneously a high number of neighbors. It means that given two nodes in a DTN, the Jaccard distance between them is likely to be close to 1. In the next section we present the History-based Jaccard distance which solves this shortcoming.

4.3. History-based Jaccard distance

In order to address the aforementioned shortcoming of the Jaccard distance in opportunistic networks, we propose a history-based version of the Jaccard distance, which takes into account the previous encounters established by the nodes to compute the history-based Jaccard distance. In this approach, nodes store the ID of the encountered nodes in a neighbor table. Whenever a new encounter occurs, nodes will use such neighbor table to calculate the Jaccard distance. Let us consider the following example shown in Figure 3 to illustrate the idea behind the history-based version of the Jaccard distance.

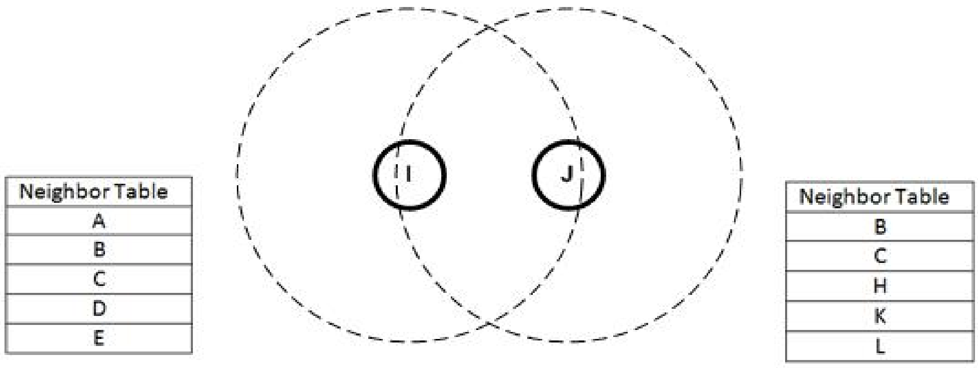

Figure 3. Example of history-based Jaccard

According to neighbor tables, the values of the coefficients a1, a2, and a3 are:

$\boldsymbol{a1 = \{B, C\} = 2}$

$\boldsymbol{a2 = \{A, D, E\} = 3}$

$\boldsymbol{a3 = \{H, K, L\} = 3}$

As a result, the history-based Jaccard distance HJ_d can be calculated as follows

$$HJ_d = 1 - \frac{a1}{a1 + a2 + a3} = 1 - \frac{2}{2 + 3 + 3} = 0.75$$

Notice that the main difference from the previous Jaccard distance version is that in the history-based Jaccard version the coefficients a1, a2, and a3 are not calculated using the current number of neighbors, but using the neighbor tables instead.

4.4. Forwarding probability in JDER

The forwarding probability of nodes in JDER will be adjusted using both the encountered ration and the history-based Jaccard distance. However, there is still a situation that has to be addressed, that is when two nodes have never encountered each other or they are not included in the neighboring table (they have encountered each other previously, but long time ago). We assign a forwarding probability of $\boldsymbol{P_m = 0.3}$ in order to assure certain reachability outside the community of the sender. As a result, the forwarding probability is adjusted as follows

$$if\ E_r = 0, p = P_m$$
$$else\ p = JH_d \cdot\ E_r$$

5. Experimental Setup and Simulation Results

This section presents an experimental analysis of the JDER data dissemination algorithm for DTNs, in terms of five metrics chosen to highlight various capabilities of such algorithms. We compare the performance of the algorithm presented above to that of distributed BUBBLE Rap (Hui et al, 2008, 2011), which is one of the most well-known and efficient data dissemination algorithms in terms of hit rate and delivery latency, and SPRINT (Ciobanu et al, 2013), a socially-aware prediction-based routing and dissemination technique. We also compare JDER to the Epidemic algorithm (Vahdat et al, 2000). Epidemic is a simple but unpractical technique which assumes that nodes have unlimited data memories, making them able to download all messages from an encountered node when a contact occurs. We use Epidemic in order to have an ideal case to compare to, since maximum hit rate is always achieved. Furthermore, we also implemented a version of JDER that has a limited encounter history cache (from now on, we will refer to this version as "limited-cache JDER").

5.1.Performance metrics

The first metric is hit rate, which is computed as the ratio between successfully delivered messages and the total number of generated messages. It shows the fraction of requests that can be served by a routing algorithm, highlighting its efficiency. The delivery latency represents the time passed between the generation of a message and its eventual delivery to the destination. In DTNs, high delivery latency values are acceptable, nonetheless, latency is a metric which should be improved in order to have a successful deployment of opportunistic networks in real life. Another metric we used in our experiments is the delivery cost, which is the ratio between the total number of messages exchanged during the course of the experiment and the total number of generated messages. Ideally, it should be as low as possible, since it shows the congestion of the network. The hop count is the number of nodes that retransmitted a message until it reached the destination on the shortest path, and this metric should also be as low as possible in order to avoid node congestion. Node congestion is also highlighted by the amount of buffer overflow events that occur at every node in the network. Buffer overflows take place when nodes are receiving messages at a higher rate than the rate at which they are able to forward, which causes their data memories to fill up. This situation should be avoided, since overflowing nodes may have to remove useful older messages to make room for new ones.

5.2.Experimentation

In order to test the capabilities of JDER, we used a trace emulator that replays existing mobility traces (Ciobanu et al 2012a). We used eight such mobility traces, most of which can be found in the CRAWDAD archives (CRAWDAD): UPB 2011, UPB 2012, St. Andrews, Intel, Cambridge, Content, Infocom and Infocom 2006. UPB 2011 (Ciobanu, 2012b) and UPB 2012 (Marin, 2012) are two traces registered in an academic environment at the University Politehnica of Bucharest using an Android application, where the participants were students and teachers at the faculty. UPB 2011 lasted for 25 days and had 22 participants, while UPB 2012 had a duration of 64 days, having 66 participants. St. Andrews (Bigwood et al, 2008) is a similar trace, but it is taken not only in the premises of the University of St. Andrews, but also in and around the surrounding town. It lasted for 79 days and had 27 participants using T-mote Invent devices. The other traces all belong to a common collection of traces of Bluetooth sightings by groups of users carrying iMote devices in various situations (Scott et al, 2009). The first of these traces, Intel, was recorded for six days in the Intel Research Cambridge Laboratory, having nine participants: a stationary node and eight iMotes. The Cambridge trace was also taken for six days, at the Computer Lab of the University of Cambridge, having as participants 19 graduate students from the System Research Group. The Content trace differs from the traces presented so far, since it was not recorded in an academic environment. Instead, it contains sightings recorded in various locations around the city of Cambridge that are likely to be visited by many people, such as grocery stores, pubs, market places and shopping centres. The participants in the experiment were students from Cambridge University, but also a series of stationary nodes placed in the key places described above. There were 18 such fixed nodes out of a total of 54. Finally, Infocom and Infocom 2006 were collected during academic conferences (the IEEE Infocom Conference in Grand Hyatt Miami and the IEEE Infocom Conference in Barcelona, Spain). Infocom had 50 participants, all students attending the student workshop at the conference, while the Infocom 2006 participants were 78 students and researchers attending the student workshop, as well as 20 stationary iMotes. The Table 1 contains the main features of the scenarios considered for analysis.

Trace	Devices		Duration (days)	Communication	Trace type
	Mobile	Fixed			
UPB 2011	22	0	25	Bluetooth	Academic
UPB 2012	66	0	64	Bluetooth and WiFi	Academic
St. Andrews	27	0	79	Bluetooth	Academic and urban
Intel	8	1	6	Bluetooth	Academic
Cambridge	19	0	6	Bluetooth	Academic
Content	36	18	25	Bluetooth	Urban
Infocom	50	0	4	Bluetooth	Conference
Infocom 2006	78	20	4	Bluetooth	Conference

Table 1. Summary of scenarios considered

We modelled the generation of messages to resemble as closely as possible a real life academic environment. Thus, every node generates 30 messages per weekday, and the destinations are chosen according to a Zipf distribution with an exponent of one, based on the nodes' social relationships. This means that nodes send more messages to members of their own communities (the nodes in these communities being chosen randomly). This is done for traces that offer such information (UPB 2011, UPB 2012 and St. Andrews), while for the other traces, the destinations are chosen randomly. The time of the day when the messages are sent is the two-hour interval when the most contacts occur for each particular trace.

For every trace, we vary the size of the data memory (the amount of messages each node can store) from 20 to 4500, since nowadays mobile devices are less and less limited by the data memory. For limited-cache JDER, we set the cache memory size empirically to 40 (i.e. a node stores information about its last 40 encounters). We ran each algorithm on each trace five times, with various values for the random number generator seed, in order to have an average value and confidence intervals. Generally, as can be seen in the next subsection, the results do not vary too much between runs.

5.3.Experimental results

5.3.1 UPB 2012 scenario

We first present the experimental results for the UPB 2012 trace. Regarding hit rate, it can be seen in Figure 4 that the two versions of JDER outperform both BUBBLE Rap, as well as SPRINT, for data memory sizes lower than 4500, with slightly better results for the unlimited-cache version. This shows that our proposed algorithm behaves well when we are dealing devices with limited storage space. For a data memory of 4500 messages, JDER achieves a hit rate close to the maximum value (which is obtained by Epidemic). In terms of delivery cost, which is shown in

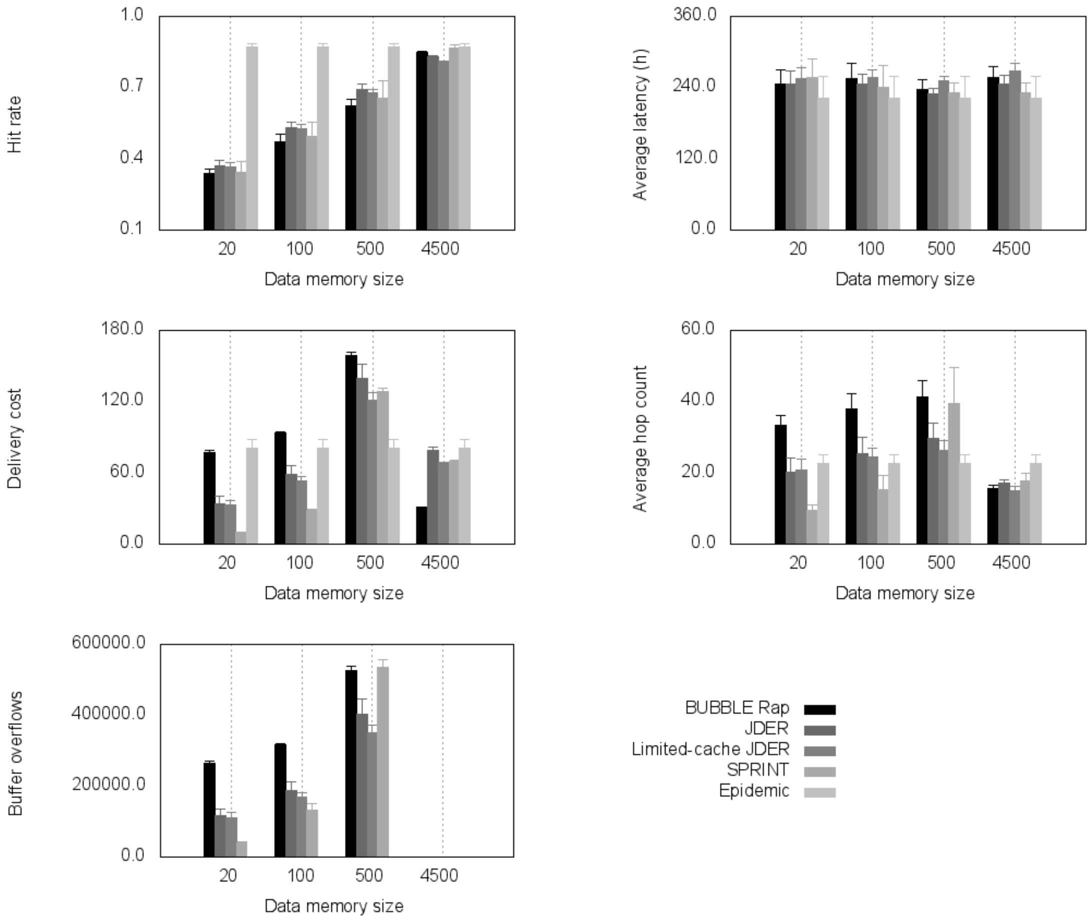

Figure 4, there is a great improvement for JDER, with less than half the number of messages from BUBBLE Rap being exchanged. Although JDER does not outperform

BUBBLE Rap for a memory size of 4500, it behaves very well for the other cases, and this happens because fewer messages are being sent due to the restriction given by the encountered ration. However, for higher data memory sizes, there is an improvement over the SPRINT algorithms obtained by the limited-cache JDER version. The JDER average latency is similar to the one obtained by SPRINT and improved by the unlimited-cache JDER version in regard to BUBBLE Rap, as shown in

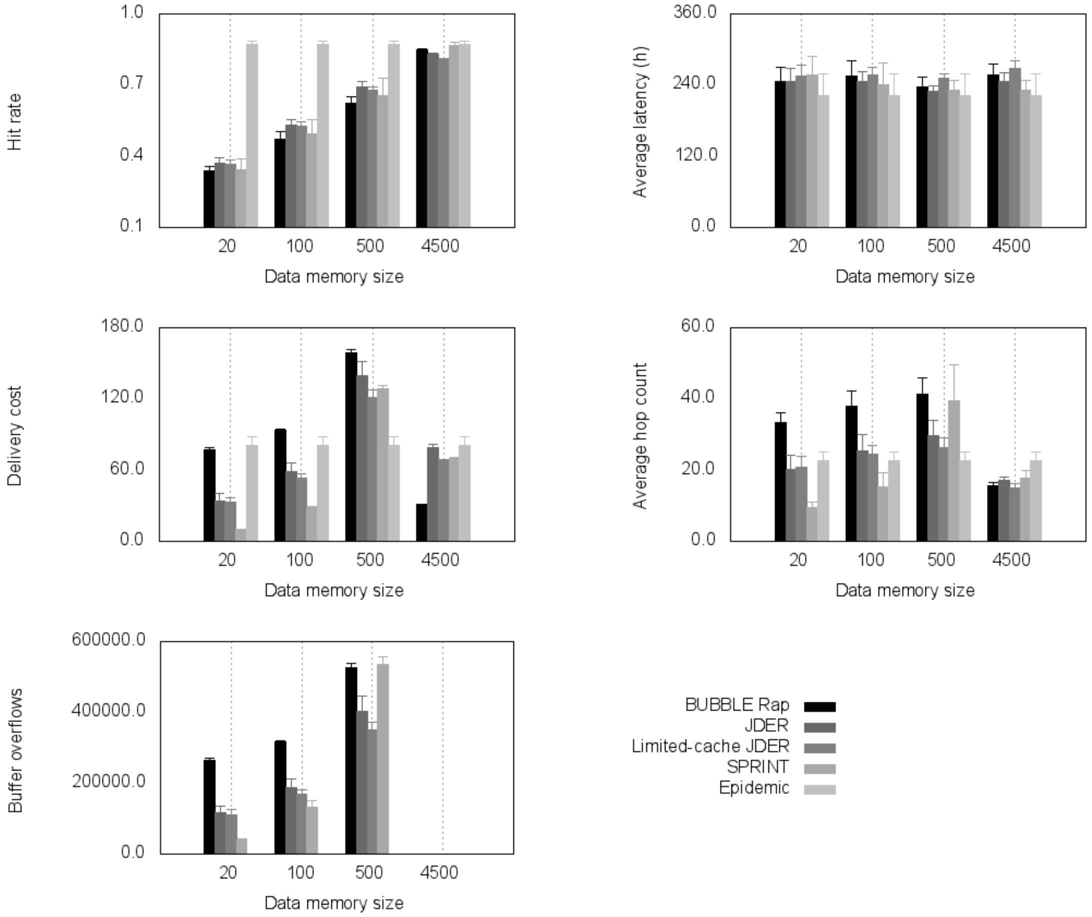

Figure 4, with an improvement of about 40 hours. This happens because, using history-based information, a node can decide more correctly what should the next hop be. Similar to the results seen for the delivery cost, the hop count is improved by both JDER versions, for the same reasons. Finally, buffer overflows are also reduced by our solution, especially for higher data memory sizes. The conclusion here is that using an encountered ration when performing opportunistic routing leads to a drop in congestion, since the exchange of messages is limited where possible, based on the history of encounters.

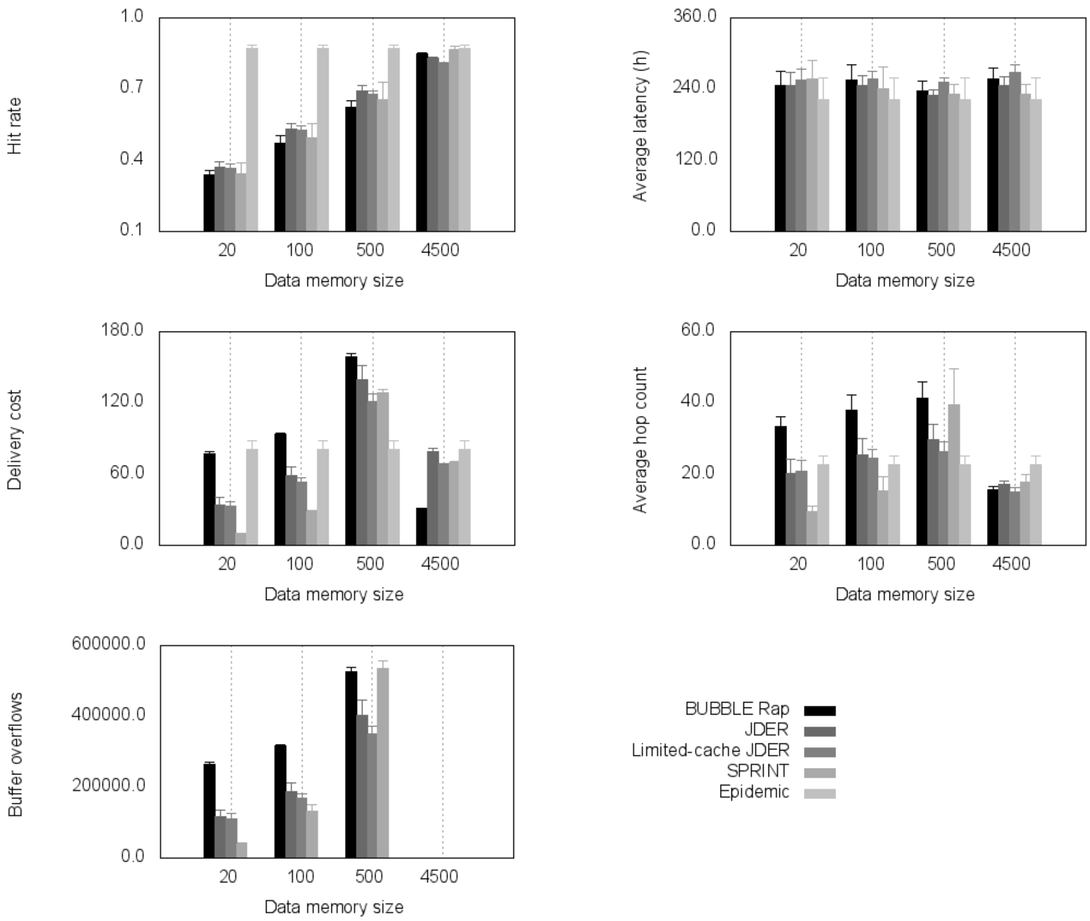

Figure 4. UPB 2012 results

5.3.2 UPB 2011 scenario

Since UPB 2011 is a trace taken in similar conditions to UPB 2012 (differing only in the duration and number of participants), the results are naturally similar, as it can be seen in Figure 5. Therefore, the hit rate is close to the value obtained when running BUBBLE Rap, and slightly lower than SPRINT's for larger data memories. However, the maximum hit rate is only achieved by SPRINT for a data memory of 4500 messages. Similarly, we achieve good results in terms of congestion (delivery cost, hop count and buffer overflows) for the same reasons as above. Both JDER versions greatly outperform distributed BUBBLE Rap (delivery cost is reduced by 37%, hop count by 19% and buffer overflow count by 94%) and perform similarly to SPRINT (delivery cost, hop count and buffer overflows are all improved for higher data memory sizes). The latency is kept around the same values as BUBBLE Rap's, being slightly higher than SPRINT's. It can be seen in Figure 5 that unlimited-cache JDER fares better than the limited-cache version, because it has a more general view of the history of encounters, and thus more information based on which it can make a decision.

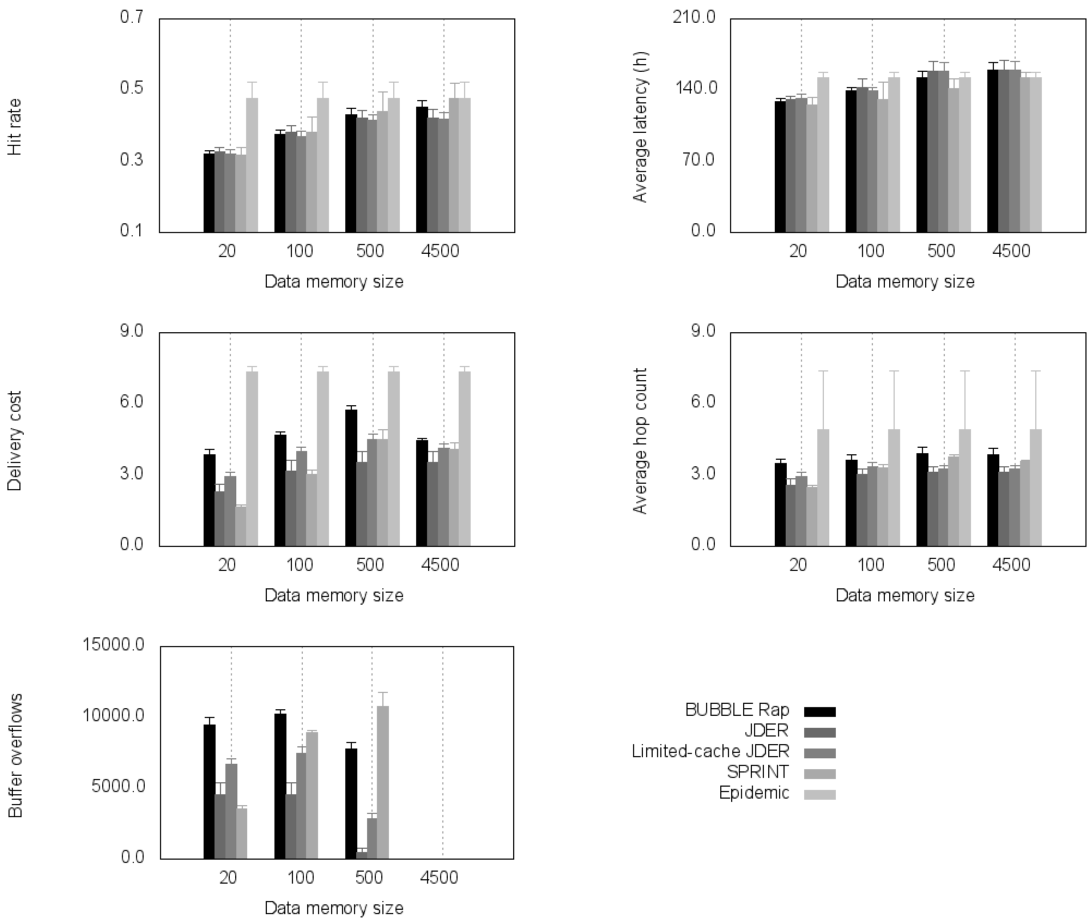

Figure 5. UPB 2011 results

5.3.3 St Andrews scenario

Since the St. Andrews trace was also taken in an academic environment, the results are somewhat similar to the ones shown so far, because the contacts between participants occur relatively regularly, see Figure 6. This happens even though the area of the experiment was a lot larger than for UPB 2011 and UPB 2012, the only difference being the time between successive contacts (the inter-contact time). Members act as carriers for one another, and since they probably belong to the same social community, they exchange a lot of directed messages between each other. Therefore, both JDER versions yield better hit rate values than both BUBBLE Rap, as well as SPRINT (with an increase of as much as 7%), although the maximum value (achieved by Epidemic) is not obtained by any other solution (meaning that this trace would require higher data memories for optimum performance). The congestion problem is drastically reduced by using the encountered ration on a history-based Jaccard algorithm. The latency obtained when running JDER is lower than for BUBBLE Rap for both JDER versions. The only other noticeable aspect of this series of tests is that the latency values fluctuate heavily between successive tests. This is most likely caused by the fact that, when changing the seed of the random number generator, we may end up having as destinations nodes that are not located in the same place as the others (e.g. the same computer lab), which leads to a drastic increase in latency, since the destination nodes are seen very rarely.

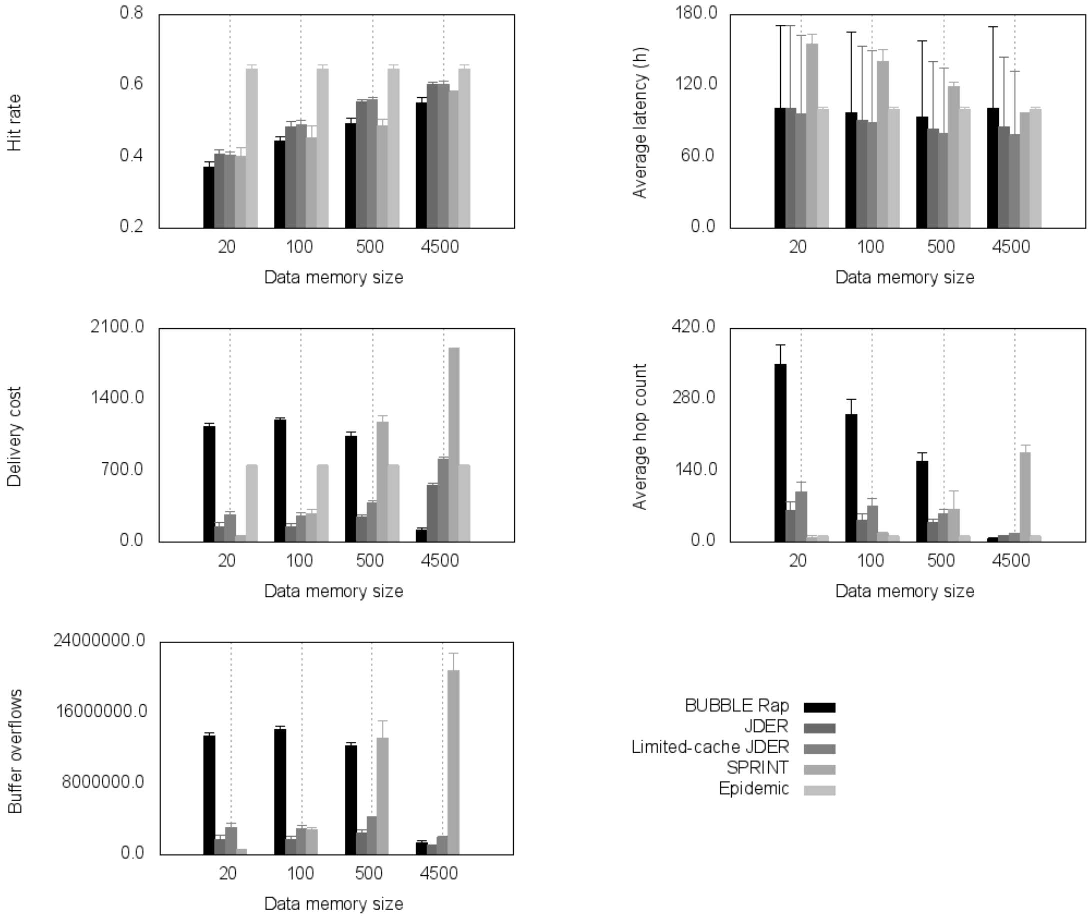

Figure 6. St Andrews results

5.3.4 Intel and Cambridge scenarios

After analyzing our proposed algorithm on the Intel and Cambridge traces, Figure 7 and Figure 8 respectively, it can be seen that the results stay the same as for the other traces presented so far in terms of hit rate: both JDER versions outperform BUBBLE Rap, as well as SPRINT, but the maximum hit rate is not achieved. However, when analyzing delivery cost, hop count and buffer overflow count, differences appear. For the previous traces, we obtained good results with both the limited and the unlimited-cache versions of JDER. In Intel's case however, limited-cache JDER performs worse than BUBBLE Rap and SPRINT, while the non-limited version retains its efficiency and still yields better results than BUBBLE Rap and SPRINT for lower data memory sizes.

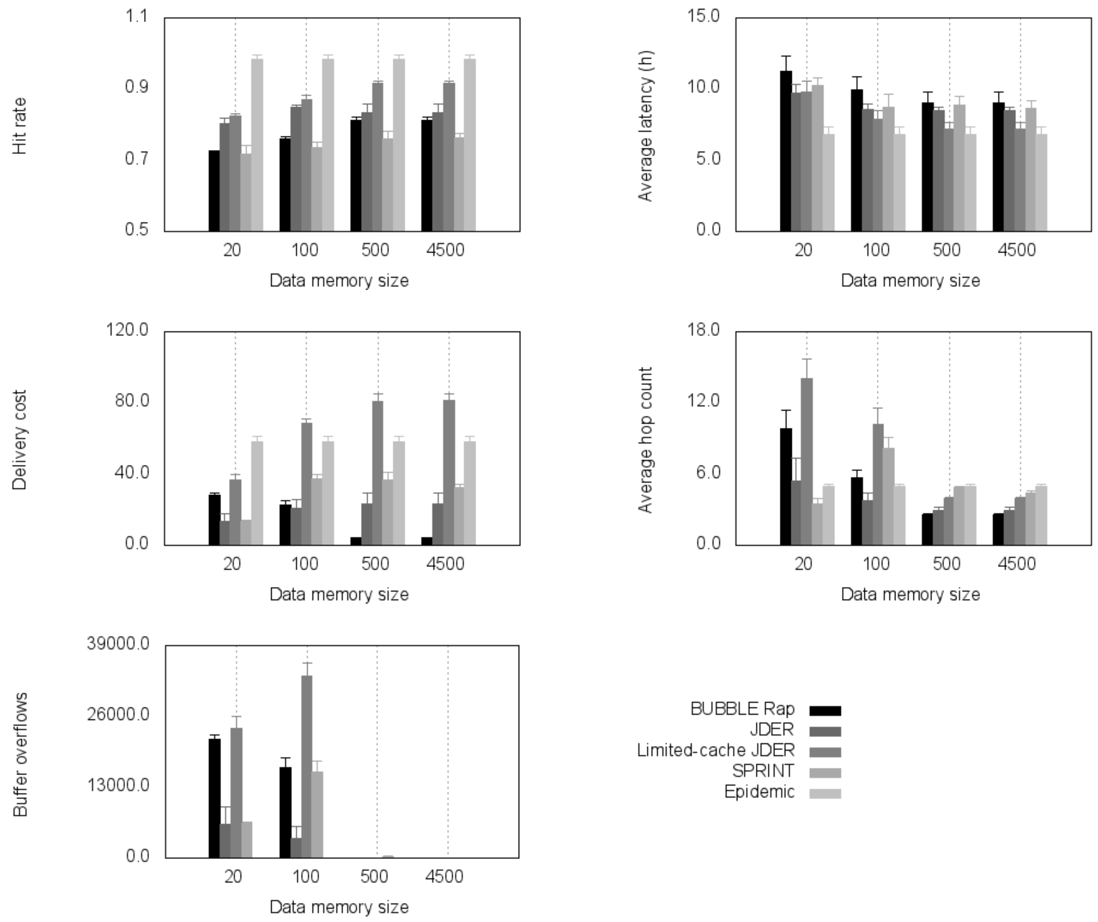

Figure 7. Intel results

The reason that limited-cache JDER behaves so badly is probably due to the size of the cache not being suitable for this situation. Increasing or decreasing it might improve upon the results (given that using an unlimited cache makes the algorithm behave fairly well, the size of the cache should probably be increased as well for the limited version of the algorithm). Regarding delivery latency, JDER still yields better results than BUBBLE Rap and SPRINT, with limited-cache JDER being the better variant.

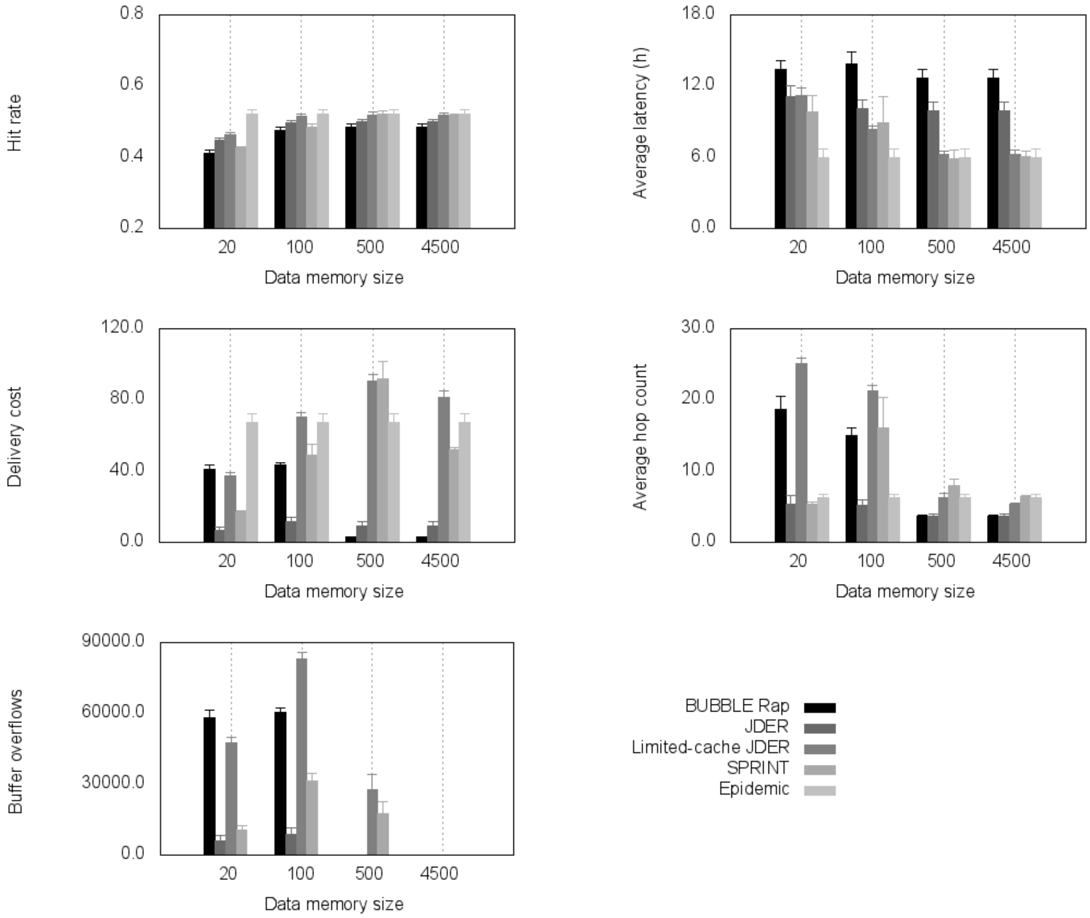

Figure 8. Cambridge results

5.3.5 Content scenario

The Content trace, as stated before, is different from the other traces because it was not taken exclusively in an academic environment, but also in the surrounding areas. The results in Figure 9, however, show the conclusions drawn so far to still hold. The hit rate is generally improved by our algorithm and so are the congestion metrics, with unlimited-cache JDER having the better results. However, for this particular scenario, SPRINT tends to handle congestion better than JDER for lower data memory sizes, because the trace exhibits a stronger Poisson distribution shape. The latency metric is not affected by our changes.

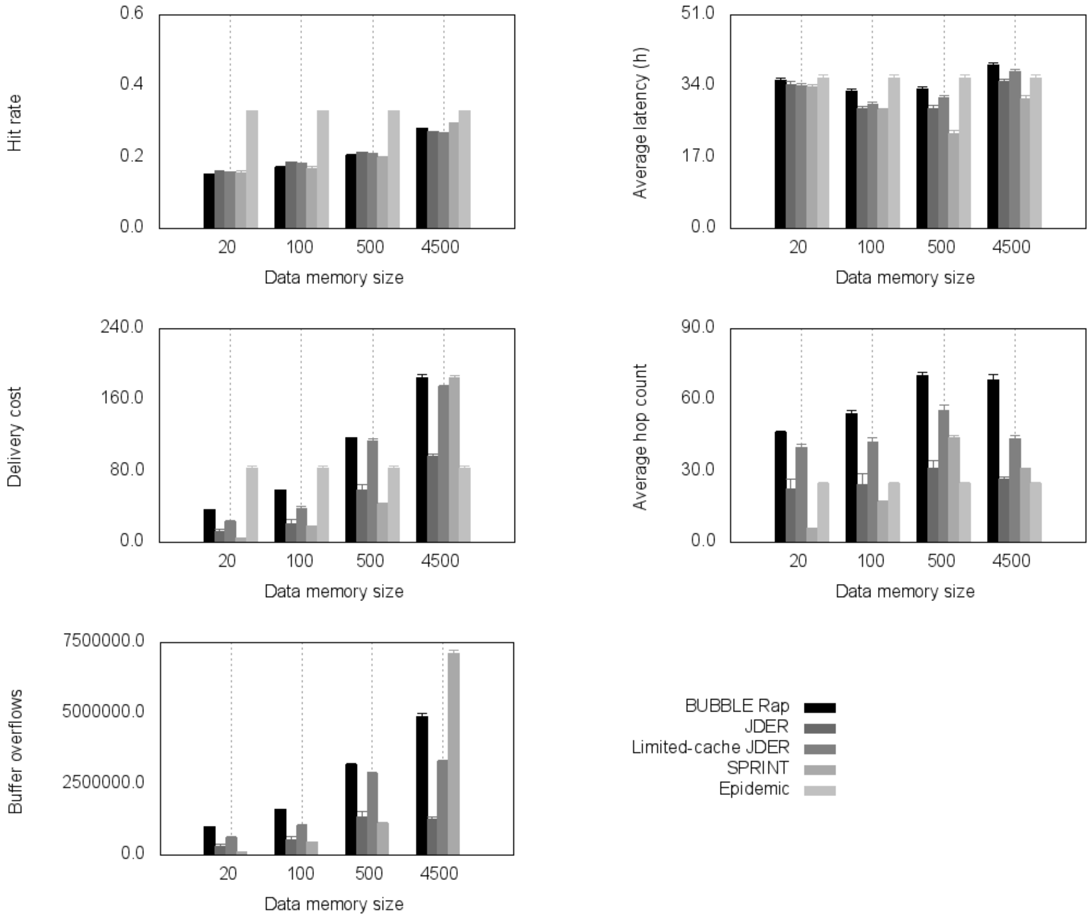

Figure 9. Content results

5.3.6 Infocom and Infocom 2006 scenarios

The two Infocom traces also represent a different situation, since they are not taken in a faculty environment, but at academic conferences, so the distribution of contacts is somewhat different. However, the results are very similar to the ones shown above. For both Infocom traces, Figure 10 and Figure 11, the highest hit rate was obtained when running the limited-cache JDER version and SPRINT, but both JDER versions outperform BUBBLE Rap. Regarding network congestion, the encountered ration helps bring the best results. For the Infocom trace, the size of the cache may be modified in order to bring better results for the limited-cache version (similar to the situation we had at Intel and Cambridge), but for Infocom 2006, the cache size seems to be correctly tuned, since the algorithm performs almost as well as the non-limited version. The latencies obtained are similar between the two versions, and both of them deliver messages faster than distributed BUBBLE Rap (and SPRINT for lower data memory sizes).

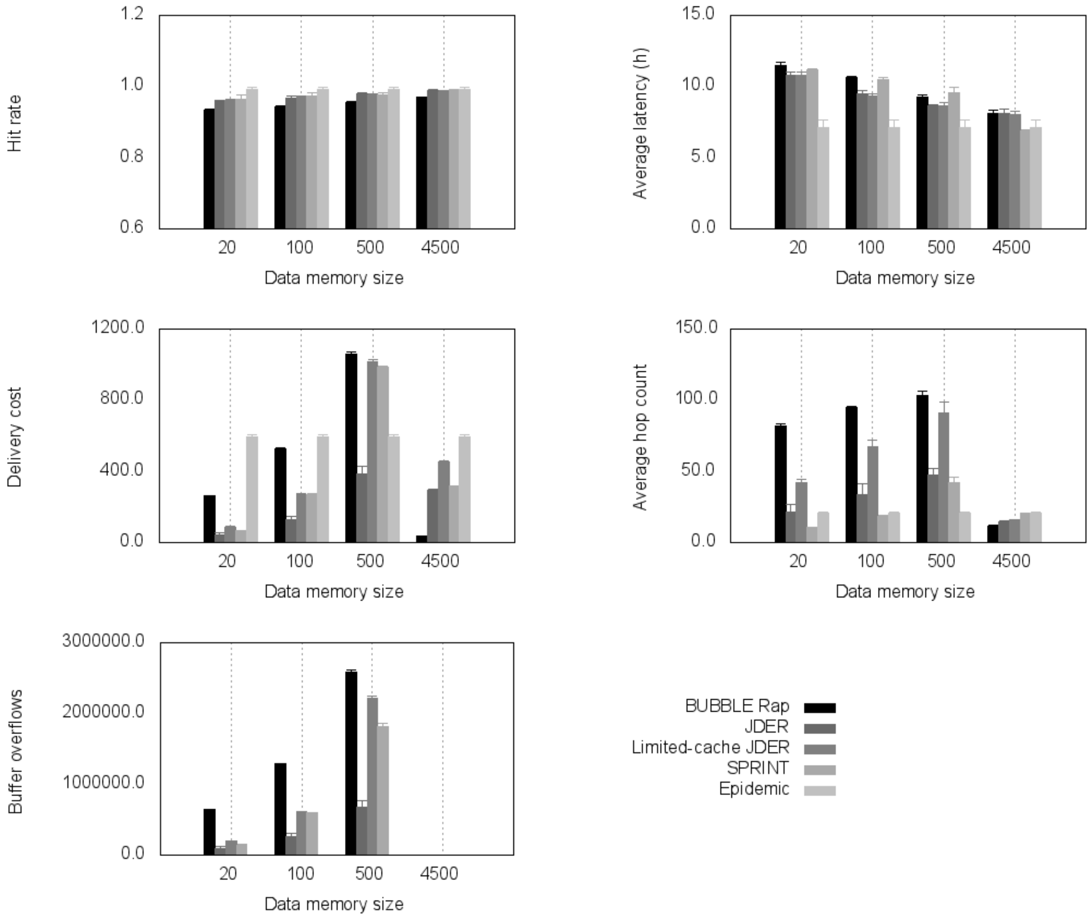

Figure 10. Infocom results

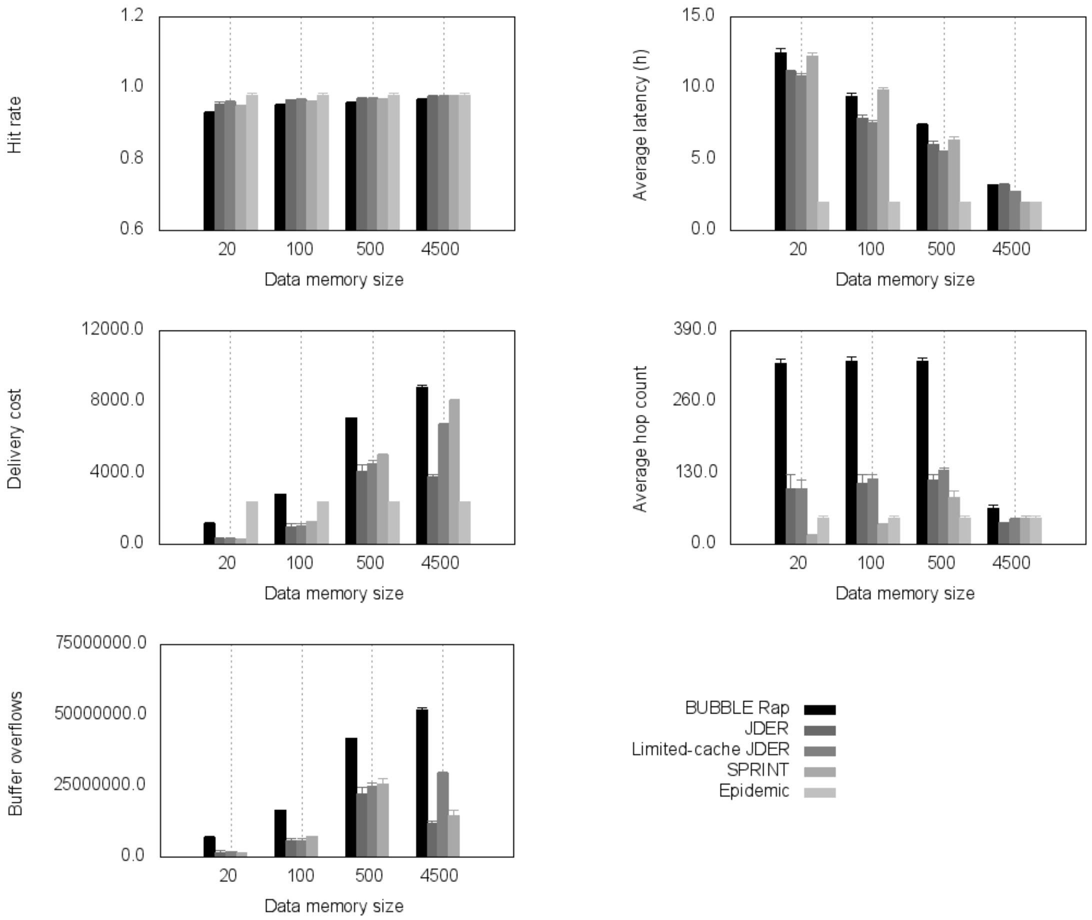

Figure 11. Infocom 2006 results

6. Conclusions

There are several conclusions that can be drawn from the results presented in the previous section. First of all, we have shown that an algorithm based on the Jaccard distance can be used for routing in opportunistic networks. Generally, for all cases we tested, the hit rate obtained is improved when using JDER. This happens because the decision of a message's next hop is more informed, and thus it has a greater chance of being delivered.

In terms of congestion, we obtain good results (which are clearly better than what distributed BUBBLE Rap yields and similar to SPRIN's) when using the encountered ration. This happens because the encountered ration has the exact purpose of limiting the total number of messages sent in the network, which leads to a decrease in congestion, both at node and at network level. Fortunately, applying this parameter only affects the hit rate very slightly, so it can be used if having a maximum hit rate is not imperative. Generally, the three congestion parameters (delivery cost, hop count and buffer overflow count) are directly proportional to each other.

Finally, although we are able to reduce the number of messages sent in the network and to improve upon the hit rate, we do not affect the delivery latency (and in some cases, we even decrease it). Although DTNs allow for high latency values, a low latency is paramount to a good opportunistic routing algorithm.

Acknowledgements

This work was supported in part by the University of Seville under the PhD grant PIF (Personal Investigador en Formación) of Daniel Gutiérrez Reina. This paper has also benefited from the collaborative research efforts of the EU Green-Net group.

References

Akyldiz I F., Su W, Sankarasubramaniam Y, and Cayirci E. A survey on sensor networks. IEEE Communications Magazine. 2002;40:102-114.

Bigwood G, Rehunathan D, Bateman M, Henderson T, and S. Bhatti S. 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, pp. 484–489. 2008.

Boldrini C, Conti M, Iacopini I, and Passarella A. HiBop: a History Based Routing Protocol for Opportunistic Networks, in IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007), pp. 1-12. 2007.

Boldrini C, Conti M, and Passarella A. Design and performance evaluation of ContentPlace, a social-aware data dissemination system for opportunistic networks. Computer Networks, 2010a;54:589–604.

Boldrini Chiara, Conti Marco, Delmastro Franca, and Passarella Andrea. Context- and social-aware middleware for opportunistic networks. Journal of Network and Computer Applications, 2010b;33:525–541.

Boldrini C, Passarella A. HCMM: Modeling spatial and temporal properties of human mobility driven by users' social relationships. Computer Communications. 2010c;33:1056-1074.

Ciobanu R. I., and Dobre C. Data dissemination in opportunistic networks, in 18th Int. Conf. on Control Systems and Computer Science, CSCS-18, pp. 529–536, 2011.

Ciobanu R I, Dobre C, Cristea V, and Al-Jumeily D. Social opportunistic networking, in Proceedings of 11th International Symposium on Parallel and Distributed Computing (ISPDC 2012). 2012a.

Ciobanu R I, Dobre C, and Cristea V. Social aspects to support opportunistic networks in an academic environment, in Proceedings of the 11th international conference on Ad-hoc, Mobile, and Wireless Networks (ADHOC-NOW'12), pp. 69-82. 2012b.

Ciobanu R I, Dobre C, and Cristea V. Reducing congestion for routing algorithms in opportunistic networks with socially-aware node behavior prediction, in Proceedings of IEEE 27th International Conference on Advanced Information Networking and Applications (AINA 2013), pp. 554-561. 2013.

Conti M, Giordano S, May M, and Passarella A. From Opportunistic Networks to Opportunistic Computing. IEEE Communications Magazine. 2010a;48:126-139.

Conti M and Kumar M. Opportunities in opportunistic computing. IEEE Computer Magazine. 2010b;43:42-50.

CRAWDAD, “http://crawdad.org/”

De Nooy W, Mrvar A, and Batagelj V. Exploratory network analysis with Pajek (Structural analysis in the social sciences). Cambridge University Press. 2005.

Hardle W and Simar L. Applied Multivariate Statistical Analysis, Springer, Ed. Method & Data Technologies. 2003.

Gao L, Li M, Bonti A, Zhou W, Yu S. M-Dimension: Multi-characteristics based routing protocol in human associated delay-tolerant networks with improved performance over one dimensional classic models. Journal of Network and Computer Applications, 2012;35:1285-1296.

Gutiérrez-Reina D, Toral S L, Johnson P, and Barrero F. A reliable route selection scheme based on caution zone and nodes’ arrival angle. IEEE Communications Letters, 2011;15:1252-1255.

Gutiérrez-Reina D, Toral S L, Johnson P, and Barrero F. Route duration improvement in wireless sensor and actuator networks based on mobility parameters and flooding control. EURASIP Journal on Wireless Communication and Networking. 2012a,1:147.

Gutiérrez-Reina D, Toral S L, Johnson P, and Barrero F. An evolutionary computation approach for designing mobile ad hoc networks. Expert System with Applications, 2012b, 39; 6838; Sy5.

Hoebeke J, Moerman I, Bart D B, and Demeester P. An overview of mobile ad hoc networks: Applications and challenges. Journal of Communications Networks. 2004;3:60-66.

Hui P, Yoneki E, Chan S Y, and Crowcroft J. Distributed community detection in delay tolerant networks, in 'Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture' (MobiArch '07), pp. 7:1–7:8. 2007.

Hui P, Crowcroft J, and Yoneki E. 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), pp. 241-250. 2008.

Hui P, Crowcroft J, and Yoneki E. Bubble rap: social-based forwarding in delay tolerant networks. IEEE Transaction on Mobile Computing, 2011;10:1576-1589.

Lenders V, May M, Karlsson G, and Wacha C. Wireless ad hoc podcasting. SIGMOBILE Mob. Comput. Commun. Rev. 2008;12:65–67.

Leontiadis I and Mascolo C. GeOpps: Geographical Opportunistic Routing for Vehicular Networks, in IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007), pp. 1-6. 2007.

Lindgren A, Doria A, and Schelèn O. Probabilistic routing in intermittently connected networks. ACM Mobile Computing and Communications Review. 2003;7:1-8

Marin R, Dobre C, and Xhafa F. Exploring predictability in mobile interaction, in Proceedings of the 3-rd International Conference on Emerging Intelligent Data and Web Technologies (EIDWT-2012). 2012.

Martín-Campillo A, Crowcroft J, Yoneki E, and Martí R. Evaluating opportunistic networks in disaster scenarios. Journal of Network and Computer Applications. DOI. 10.1016/j.jnca.2012.11.001, 2013.

Pelusi L, Passarella A, Conti M. Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks. IEEE Communications Magazine. 2006;44:134-141.

Shin K, Kim K, Kim S. Traffic management strategy for delay-tolerant networks. Journal of Network and Computer Applications, 2012;35:1762-1770.

Scott J, Gass R, Crowcroft J, Hui P, Diot C, and Chaintreau A. CRAWDAD data set cambridge/haggle, Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009.

Tahsin P A, Das R, Kabir H, and Shoja Gholamali C. TTL based routing in opportunistic networks. Journal of Network and Computer Applications, 2011;34: 1660-1670.

Tulloss R E. Assessment of Similarity Indices for Undesirable Properties and a new Tripartite Similarity Index Based on Cost Functions. M. E. Palm and I. H. Chapela, Eds. Parkway. 1997.

Vahdat A, Becker D. Epidemic routing for partially connected ad hoc networks. 2000.

Yoneki E, Hui P, Chan S Y, and Crowcroft J. 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, pages 225–234. 2007.