A Data Dissemination Algorithm for Opportunistic Networks

Radu-Ioan Ciobanu*, Ciprian Dobre*, Valentin Cristea*
* University POLITEHNICA of Bucharest, Romania
* Bucharest, Romania
E-mails: radu.ciobanu@cti.pub.ro, {ciprian.dobre, valentin.cristea}@cs.pub.ro

Abstract

Opportunistic networks are an evolution of MANETs, where highly mobile nodes that have no physical route connecting them might need to communicate. In this case routes are built dynamically, as nodes act according to a store-carry-and-forward paradigm. A natural continuation of forwarding in opportunistic networks is data dissemination, which is usually performed using the publish/subscribe model. In this paper, we propose a socially-based data dissemination technique for opportunistic networks called Social Dissemination. The solution assumes that nodes are grouped in communities, where nodes in the same community meet often, while at the same time certain nodes may have relationships with nodes from other communities. When two nodes meet, they exchange information about the data they store, and select data objects from the other node according to a utility function. We define such a utility function and experimentally compare its results with state-of-the-art previously proposed dissemination techniques.

***Keywords-* opportunistic networking, data dissemination, mobile devices**

1. Introduction

Opportunistic mobile networks consist of human-carried mobile devices that communicate with each other in a "store-carry-forward" fashion, without any infrastructure. Compared to classical networks, they present distinct challenges [1]. In opportunistic networks, disconnections and highly variable delays caused by human mobility are the norm. The solution consists in dynamically building routes, as each node acts according to the store-carry-and-forward paradigm [2]. Thus, contacts between nodes are viewed as an opportunity to move data closer to the destination.

Such networks are, therefore, formed between nodes spread across the environment, without any knowledge of a network topology. The routes between nodes are dynamically created, and nodes can be opportunistically used as next hop, for bringing each message closer to the destination. Nodes may store a message, carry it around, and forward it when they encounter the destination or a node that is more likely to reach the destination.

An important topic in opportunistic networks is represented by data dissemination, which is a natural continuation of forwarding, and is usually based on a publish/subscribe model [3]. In this paper we present Social Dissemination, a data dissemination algorithm that considers human social behavior to form opportunistic networks. The algorithm uses the grouping of nodes (human-carried mobile devices) in communities, similar to the caveman model [4].

According to the proposed algorithm, when two nodes are in range of each other, they exchange information about the data they store, and they compute utility values for their own and the encountered node's data. The utility values are used by each node to request from its peers data that is needed, or that may be useful to other nodes it may encounter along the way. The utility function considers the communities that a node has encountered, as well as the communities it is likely to encounter in the future.

The rest of the paper is structured as follows. Section 2 presents an overview of related work. Section 3 describes the dissemination algorithm, including the protocol used when two nodes are in range, and the proposed utility function. In Section 4 we present experimental results obtained using the Social Dissemination algorithm, and compared them against results obtained in the same scenario for other existing algorithms. Finally, Section 5 concludes the paper and presents future work.

2. Related work

In the domain of opportunistic networking Conti, *et al* [5] presents a thorough review of state-of-the-art algorithms. The authors compare several well-known opportunistic forwarding algorithms, such as Bubble Rap, Propicman and HIBOp, as well as several data dissemination in opportunistic algorithms. We use similar experimental validations for drawing conclusions regarding the performance of our proposed algorithms, as compared to previously proposed solutions.

There are several papers that exclusively propose dissemination algorithms for opportunistic networking.

Yoneki, *et al* [6], proposed Socio-Aware Overlay, an algorithm that creates an overlay for an opportunistic network with publish/subscribe communication. Wireless Ad Hoc Podcasting [7] is another dissemination algorithm developed with the purpose of wireless ad-hoc delivery of content among mobile nodes. DTN Pub/Sub Protocol (DPSP) [8] is a publish/subscribe-based multicast distribution method for opportunistic networks. Finally, Boldrini, *et al*, [9] proposed a dissemination technique called ContentPlace that deal with data dissemination in resource-constrained opportunistic networks.

Based on previously proposed experimental results, ContentPlace will be used for comparing the results obtained using our proposed algorithm, as it has good performance in regard to hit rate and latency. Content Place tries to solve problems faced by previous solution to optimize content availability. For that it exploits learned information about users' social relationships to compute a utility function. The function uses weights based on the social aspect of opportunistic networking, and defines five different policies, called Most Frequently Visited (*MFV*), Most Likely Next (*MLN*), Future (*F*), Present (*P*) and Uniform Social (*US*). *MFV* favors communities a user is most likely to get in touch with, while *MLN* favors communities a user will visit next. *F* is a combination between *MLN* and *MFV*, as it considers all the communities the user is in touch with. In case of *P*, users do not favor other communities than the one they are in, while at *US* all the communities the users get in touch with have equal weights. Two other non-social policies are also analyzed, namely *Greedy*, in which all weights except the current node's are 0, and *Uniform*, where all weights are equal.

3. The Social Dissemination algorithm

As the name implies, Social Dissemination is a social algorithm that is also based on the publish/subscribe paradigm. Its purpose is to disseminate data published by a channel to all the subscribers of that channel, as fast and as efficiently as possible.

3.1. The Algorithm

Each node from the opportunistic network on which Social Dissemination is applied stores various types of data, as seen in Figure 1. First, it stores information about the channels it is subscribed to and the class of the data objects required from that channel, along with a timestamp of when the node has asked for a certain data object. Aside from this, each node has two types of memory: data and cache. The *data memory* stores data that the current node has received from other nodes. The data can either be of interest to the current node or it can be stored and carried for opportunistic use in the network. It is organized as a bundle containing the ID of the channel and the class that the data belongs to, together with a timestamp of when the data was received, and the actual data (see Figure 1).

The *cache memory*, on the other hand, stores information about the nodes from the opportunistic network that have been recently encountered. For each entry we store the ID of the node and the community it belongs to. Furthermore, each node in the network can act as a publisher, so the published data is also stored in this memory.

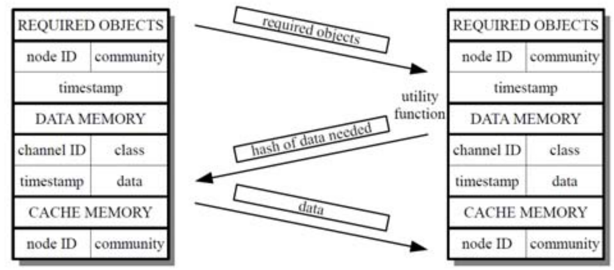


Fig.1. Social Dissemination algorithm protocol.

When two nodes running the Social Dissemination algorithm are within wireless range of each other, they begin their communication by advertising the data they store. This is sent as a bundle that contains the ID, the class and the timestamp of every data object contained (but not the actual data).

When a node receives this data bundle from another node, it analyzes it in order to find data that it needs (i.e. data from the channel it is subscribed to). If this kind of data is not found, then the node looks for data that it can disseminate along in the network. In order to select which data objects it will require from the neighboring node, the current node uses a function to compute a utility value, which is then used to rate the objects the encountering node contains, and to ask for the ones that are most useful. The most useful data is the one that maximize the value of the current node's data memory. Therefore, when an object is received from a neighboring node, the object having the lowest utility value from the current node's data memory is replaced.

After a node decides what objects to request from its neighbor, it sends a hash requesting those particular objects. The peer then sends the requested data as long as the two nodes are in range. The functioning of the Social Dissemination algorithm when two nodes are in range is presented in Figure 1.

3.2. The utility Function

Because the algorithm proposed in this paper is based on the social aspects of opportunistic networking, its utility function is computed according to the community a node belongs to, and to the communities it has had contacts with so far.

The social facet of the algorithm is based on the caveman model [4], which assumes that users belong to "home" communities, but they can also have relationships outside of their home communities (in "acquainted" communities). Recent studies revealed the qualities of the model to describing realistic mobility and social patterns [10].

According to this model, users spend their time in the locations of their home communities (which are bound to a particular place), but also visit areas where acquainted communities are located. Thus, the movement of nodes in the opportunistic network is governed by the social relationships between them.

Therefore, the utility function is computed according to the following formula:

$$u = \sum_{i=1}^{n} p_i^c \left(1 - p_i^e\right) + freshness + c_v \quad (1)$$

In this formula, n is the number of communities that exist in the network, p_i^c is the percentage of nodes from community i that the current node has encountered so far, and p_i^e is the percentage of nodes from community i that the encountered node has been in contact with so far. These percentages are computed over a range of values equal to the size of each node's cache, because the cache is where information about encountered nodes and their communities is stored. The reason why p_i^e is subtracted from *1* is that, if a node has not encountered many nodes from a specific community, the data object it stores might not have entered that community. Therefore, the value for that community should be higher. The *freshness* value ensures that the value of the utility is higher for newer objects. For recently generated data objects there is a high chance that the community has not seen them yet. Therefore, a high *freshness* value leads to a higher utility value, and the associated object will have a higher chance of reaching the current node's community faster. Finally, c_v is an additional element added to the utility value to reflect the case when two nodes belong to different communities.

4. Experimental results

In this section we present evaluation results for the proposed Social Dissemination algorithm. These experiments are based on the analysis and experiments first proposed in [9]. We extended the evaluation models with new capabilities that made possible the analysis of the Social Dissemination algorithm. Then, based on the survey presented in Section 2, we compared our solution with the one proposed by ContentPlace [9].

4.1. A mobility model for evaluating opportunistic data dissemination algorithms

The experimental evaluation is based on modeling and simulation. For that we developed a mobility model simulator, starting from the Home-Cell Mobility Model (HCMM) proposed in [11]. The mobility model assumes 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. The mobility model is also based on the caveman model, and assumes that each node is attracted to its home cell according to the social attraction exerted on that node by all nodes that are part of its community [11].

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 $1\text{-}p_e$. The authors in [11] present extensive results for the validation of the model.

We adapted the HCMM model to permit a graphical representation of the movement of nodes, as well as the division of the movement area into cells. This allows us to visually observe the positions of nodes and the contacts between them at all times. The mobility application also facilitates the setting of various parameters of the model through a specialized GUI. Among the parameters that the user can influence, there are: size of the movement area, number of existing cells, number of nodes in the network, number of communities, speed of the nodes, or simulation time. Moreover, statistical data is collected and written to a file at the end of the simulation. The mobility model simulator also allows the addition of a dissemination algorithm through the simple implementation of a Java interface.

4.2. Experimental scenario

The experimental scenario is similar to the one proposed in [11]. Thus, we assumed an opportunistic network with 45 nodes, uniformly grouped into three communities. Each of these three communities has a home cell chosen so that it is not adjacent to a home cell of another community (because that would lead to the existence of physical shortcuts between communities). The nodes move in a 1000x1000 grid, divided into 16 cells. There are three channels in this scenario, each of them generating 99 data objects, and each community is the source for one third of the objects generated by a channel. Therefore, each community generates 99 data objects, 33 for each channel.

According to this scenario a node can only be subscribed to a single channel. Also the interests of nodes are distributed according to a Zipfian distribution within each community, where one channel is the most popular in one community and the least popular in another community.

The nodes in the opportunistic network ask for data objects from a publisher according to a Poisson distribution with $\lambda = 200$, and the data object they request is selected according to a uniform distribution within the channel they are subscribed to.

Among the other parameters of the simulation, the speed of the nodes is between 1 m/s and 1.86 m/s, each node having a transmission range of 20 meters. A simulation runs for an average of 50,000 seconds, and the data memory size of each node is set to 33 objects. The simulation scenario assumes that each pair of communities is connected through a single node. Therefore, Community 1 (C1) has two nodes (entitled “travelers”) that have relationships in Community 2 (C2) and Community 3 (C3). These nodes leave C1, move for a while within their acquainted communities (C2 and C3), and then return to the home community. After moving around in C1, they repeat this sequence of actions. The results presented in the next subsection also assume that the two traveler nodes are subscribed to Channel 1.

We also performed a second series of experiments to analyze the scaling capabilities of the Social Dissemination algorithm, when the number of communities and nodes increases. In the first experiment we considered 120 nodes, with the number of communities varying from 2 to 6. In the second experiment, the number of communities is kept fix (to 4), while the number of nodes is 40, 60, 120 and 240. For both these experiments, the simulator’s parameters are the same as in the first scenario, except for the number of cells in the grid (which is 25) and the rewiring probability, which is 5%. This means that there will not be only a traveler node that moves between the first community and the others, but nodes from each community move between communities with a probability of 5%. There are also four channels in this particular scenario, each one producing 240 objects. The interests of nodes and the requests are distributed like in the first scenario.

4.3. Experimental results

Based on the described scenario we conducted a series of experiments. The analysis of an opportunistic networking algorithm is based on four metrics: hit rate, fairness, resource consumption and latency.

The *hit rate* is an important metric for an opportunistic network, as it represents the ratio between data objects that have successfully arrived at requesting nodes, and the total number of requests generated by all nodes. The hit rate suggests the efficiency of a dissemination algorithm, and in an ideal algorithm it would be 100%. It shows the fraction of requests that can't be served by a dissemination algorithm. Figure 2(a) shows the hit rates for the seven ContentPlace policies (presented in Section 2), compared with the results obtained for the Social Dissemination algorithm, for nodes belonging to community C1. The Social Dissemination algorithm obtains an average hit rate of about 91% for all three channels. If the travelers are subscribed to one of the other channels, the results are similar.

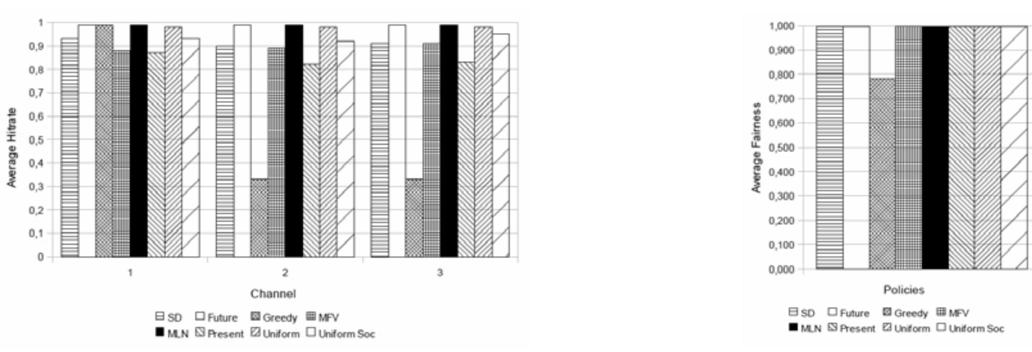
It can be observed that the Social Dissemination algorithm obtains an average hit rate that is better than three of the seven ContentPlace policies. Among the policies that are better than our algorithm, Future and Most Likely Next are based on approximating the next community that a node will visit. Still, for a larger memory size (both data and cache), the Social Dissemination algorithm manages to achieve a hit rate of 100%.

The *fairness* is computed according to Jain's fairness index, where the hit rate is a measure of the service level obtained by each channel. As seen in Figure 2(b), the Social Dissemination algorithm is fair in regard to the three channels used in our experiment, as we obtained a value very close to 1. The only ContentPlace policy that does not achieve such a fairness level is Greedy.

Another metric used for testing is the *resource consumption*, measured in terms of traffic generated in the network by the dissemination algorithm. According to [9], the best policy in terms of bandwidth overhead is by far Greedy. The next one, with an overhead of approximately 508 MB, is the Future Policy, followed closely by MSN with 509 MB.

For the Social Dissemination algorithm, the experiments indicate that the proposed technique performs slightly better than the Future policy in ContentPlace, for a bandwidth overhead value of about 480 MB.

The reason for an improvement over the Future policy is that a node in the Social Dissemination algorithm only fetches a node that has recently entered the community (according to the *freshness* value) and was delivered by a traveler node (by using the c_v value in the utility function).



(a) Hit rate.

(b) Fairness

Fig. 2. Hit rate and fairness for Social Dissemination and ContentPlace.

The final metric used for analyzing the performance of the Social Dissemination algorithm is the *latency* in satisfying node requests. This is computed as the difference between the time a data object request has been satisfied, and the time the request was made. Figure 3 shows the complementary cumulative distribution function for the delay of requests that have been satisfied for nodes in Channel 1. In [9], the results were presented only for the Greedy, Uniform and MFV policies, because Present and Uniform Social are equivalent to MFV, while Future and MLN are overlapped. As demonstrated in [9], Greedy clearly achieves the best performance, with the Future policy being the next one.

As presented in Figure 3 the Social Dissemination algorithm outperforms both Uniform and MFV. These experiments were performed for a network that accepts infinite delays for requests. For our algorithm, the average latency value is about 627 seconds.

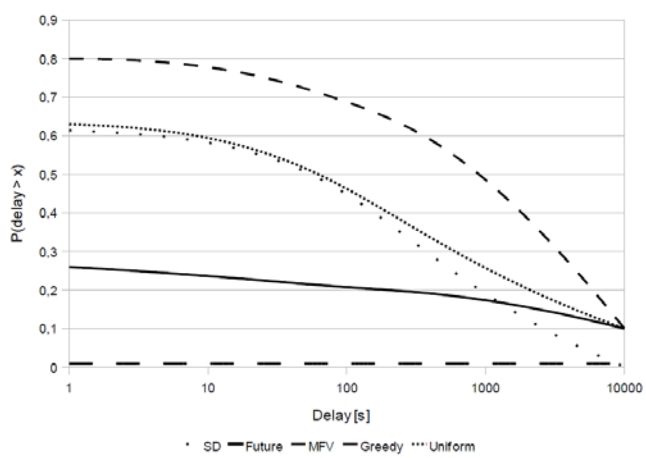


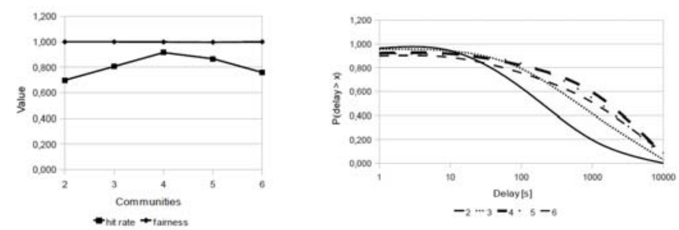
Fig. 3. Delay CCDF for Social Dissemination and ContentPlace.

After analyzing the results of the Social Dissemination algorithm in comparison to the seven ContentPlace policies, we can state that there is no policy that outperforms our technique from the standpoint of all evaluated four metrics. The best two policies from ContentPlace are considered by the authors to be Future and Most Likely Next [9], which do slightly outperform the Social Dissemination technique in regard to hit rate and latency, but not from the standpoint of bandwidth overhead. However, the Social Dissemination algorithm can obtain a hit rate close to 100%.

We next continued our experiments, looking into the scalability of the proposed algorithm. Figure 4(a) presents the hit rate and fairness, and Figure 4(b) presents the latency CCDF, when the number of communities varies from 2 to 6. These metrics are computed as described above, with the hit rate computed as an average of hit rate values for all the four channels. This was done because, as seen in Figure 4(a), the fairness for all tests was very close to 100%, so the hit rate values are almost equal for all channels.

There is an interesting conclusion that can be reached by analyzing Figure 4(a), namely that the highest hit rate value is achieved when the number of channels is equal to the number of communities. This happens because the data objects from each channel are uniformly spread uniformly across communities when there as many channels as there are communities. It is also clear from the same figure that the hit rate for 6 communities is better than the hit rate for 2 communities, so we can state that, when increasing the number of communities, the Social Dissemination algorithm scales relatively well. Another advantage of the Social Dissemination technique is that it maintains a fairness of very close to 100% towards all channels, no matter how many communities there are in the network.

Figure 4(b) shows that the delay of receiving a data object increases with the number of channels, but not very drastically. However, it is important to note that opportunistic networks are a type of Delay Tolerant Networks (DTNs), where the delay is a factor that is not such as important as hit rate. The maximum latency in these experiments was obtained when testing with 6 communities, and had a value of about 2500 seconds.

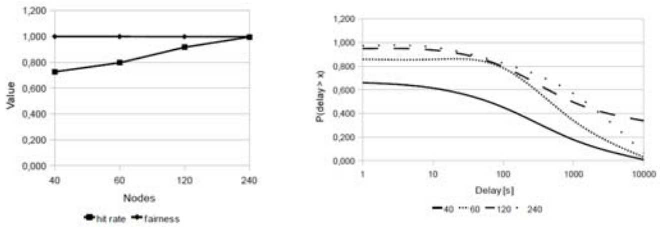


(a) Hit rate and fairness (b) Delay CCDF

Fig. 4. Hit rate, fairness and delay CCDF for Social Dissemination when increasing the number of communities.

For the second experiment the number of communities is 4 (and kept fixed), while the number of nodes varies (with values of 40, 60, 120 and 240). Figure 5 presents the hit rate, fairness and latency CCDF, as above. As seen in Figure 5(a), similar to the experiment where the number of communities is increased, the fairness remains very close to 100%, meaning that every community receives an almost equal percentage of data objects from the ones they requested. Moreover, the same figure shows that the hit rate obtained when testing with 4 communities grows with the number of users in the network, reaching 100% for 240 nodes.

Again, this shows that our proposed algorithm scales extremely well even when the number of nodes in the network is increased. This happens because, the more nodes there are in a community, the greater the number of nodes that travel between two communities is. When a node leaves its community, it takes content generated in its home community with it and spreads it in the communities it encounters along its way. Therefore, this content reaches other communities, where it is disseminated according to the algorithm presented in Section 3. The latencies for this set of experiments, presented in Figure 5(b), show an increase as the number of nodes increases, but this increase is not very significant, and (as stated before) it is acceptable in DTNs.



(a) Hit rate and fairness (b) Delay CCDF

Fig. 5. Hit rate, fairness and delay CCDF for Social Dissemination when increasing the number of nodes.

5. Conclusions and future work

In this paper we have proposed a social-based dissemination algorithm for opportunistic networks entitled Social Dissemination. This algorithm is based on the social aspect of opportunistic networking, assuming nodes belong to users grouped into communities. We have compared the results of Social Dissemination with the ones obtained in ContentPlace, and showed that there is no ContentPlace policy that outperforms the Social Dissemination algorithm in all analyzed categories. Furthermore, we have proven that the proposed dissemination technique scales well when the number of nodes or communities is increased, obtaining good hit rates and an acceptable growth in node request latency.

As future work we plan to extend these results and modify the utility function such that to also consider the communities a node is more likely to visit in the future, as they may lead to better results in regard to hit rate. Furthermore, we plan to implement the algorithm for real-life use, using smartphones.

Acknowledgments

The research presented in this paper is supported by national project ``DEPSYS - Models and Techniques for ensuring reliability, safety, availability and security of Large Scale Distributes Systems", Project CNCSIS-IDEI ID: 1710. The work has been co-funded by national project ``TRANSYS – Models and Techniques for Traffic Optimizing in Urban Environments", Contract No. 4/28.07.2010, Project CNCSIS-PN-II-RU-PD ID: 238, and 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.

6. References

[1] Lilien, L., Z. H. Kamal, and A. Gupta, "Opportunistic Networks: Challenges in Specializing the P2P Paradigm", in *Proc. of the 17th Int. Conf. on Database and Expert Systems Applications* (DEXA '06). IEEE Computer Society, Washington, DC, USA, pp. 722-726, 2006.

[2] Pelusi, L., A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks", *IEEE Communications Magazine*, 44, pp. 134-141, November 2006.

[3] Boldrini, C., M. Conti, and A. Passarella, "Context and resource awareness in opportunistic network data dissemination", in *Proc. of the Int. Symp. on a World of Wireless, Mobile and Multimedia Networks* (WoWMoM 2008), Newport Beach, CA, USA, pp. 1-6, 2008.

[4] Wu, J., and D. J. Watts, "Small worlds: the dynamics of networks between order and randomness", *SIGMOD Rec.*, 31, pp. 74-75, December 2002.

[5] Conti, M., S. Giordano, M. May, and A. Passarella, "From opportunistic networks to opportunistic computing", *IEEE Communications Magazine*, 48, pp. 126-139, September 2010.

[6] Yoneki, E., P. Hui, S.Y. Chan, and J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks", in *Proc. of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems* (MSWiM '07), New York, NY, USA, pp. 225-234, 2007.

[7] Lenders, V., M. May, G. Karlsson, and C. Wacha, "Wireless ad hoc podcasting", S*IGMOBILE Mob. Comput. Commun. Rev.* 12, 1 (January 2008), pp. 65-67, 2008.

[8] Greifenberg, J., and D. Kutscher, "Efficient publish/subscribe-based multicast for opportunistic networking with self-organized resource utilization", in *Proc. of the 22nd International Conference on Advanced Information Networking and Applications*, Washington, DC, USA, pp. 1708-1714, 2008.

[9] Boldrini, C., M. Conti, and A. Passarella, "Design and performance evaluation of ContentPlace, a social-aware data dissemination system for opportunistic networks", *Computer Networks*, 54, pp. 589-604, March 2010.

[10] Musolesi, M., and C. Mascolo, "A community based mobility model for ad hoc network research", in *Proc. of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality* (REALMAN '06), New York, NY, USA, pp. 31-38, 2006.

[11] Boldrini, C., M. Conti, and A. Passarella, "User-Centric Mobility Models for Opportunistic Networking. In Bio-Inspired Computing and Communication", Lecture Notes In Computer Science, Vol. 5151. Springer-Verlag, Berlin, Heidelberg, pp. 255-267, 2008.
.