

An efficient Multi-Clusters based Routing algorithm in Mobile Opportunistic Networks

Jie Peng¹, Feng Zeng^{1,*}, Wenjia Li^{2,*}

¹ School of Software, Central South University, Changsha, China

² Department of Computer Science, New York Institute of Technology

* Drs Feng Zeng and Wenjia Li are both corresponding authors for this paper

Abstract. Mobile Opportunistic Networks (MONs) are formed by random moving nodes. In MONs, there is no fixed link between any two nodes, the node can only transfer the message based on “storage-carry-and-forward” data transmission mode. Therefore, a key challenge is to find a routing that can provide a good message delivery performance in MONs. In this paper, we use a phenomenon in real life to design an efficient Multi-Clusters based routing algorithm MCR. At first, we described a phenomenon that most of us have circle of friends, but also exists in multiple friends circles. Then, we used the encounter history to set up building and updating mechanism for the circles. After that, the multi-clusters had been formed with these friends circles. And we used the multi-clusters to design MCR and make some experiments on real mobility traces. Finally our experiments results had shown that MCR outperforms the other three routing algorithms.

Keywords: Mobile opportunistic networks; Multi-Clusters; Routing algorithm.

1 Introduction

Mobile Opportunistic Networks (MONs)[1] comes from Delay-Tolerant Network(DTN), and can be seen as an application of DTN. In MONs, the node can only transfer message based on “storage-carry-and-forward” data transmission mode, that means the node stores the incoming message into its buffer and then delivers the messages towards destination node whenever it is within the range of wireless coverage. When the destination node can not be directly accessed, the node carrying the messages can only store the messages until the lifetime is over or transfer the messages to neighbor nodes.

The way of data transmission mode in MONs makes it possible that use the phenomenon in real life to improve the network efficiency. Here is a typical example, that people have their own circle of friends, and spend lots of time with friends. Besides, most of us have several friend circles rather than one. Maybe the circles are composed of our classmates, participants of the the school’s English club, or even our relatives. Whereas, these circles are likely to intersect. For instance, John is my classmates, we participated in the English club at the same time, and he is my cousin too. We can see that John exists in many friend circles of mine, it is obvious that John and I is on intimate terms. It is same in MONs, as is shown in the following pictures. At time T1, each node has no contact with other nodes, and there is no relation between any nodes. At time T2, due to the movement of nodes, many nodes connected, and different nodes constitute several different clusters. At time T3, with the relations between

nodes have changed, there are some intersections between clusters. These intersections can be seen as consisting of people who are intimate with others, such as John in the real life.

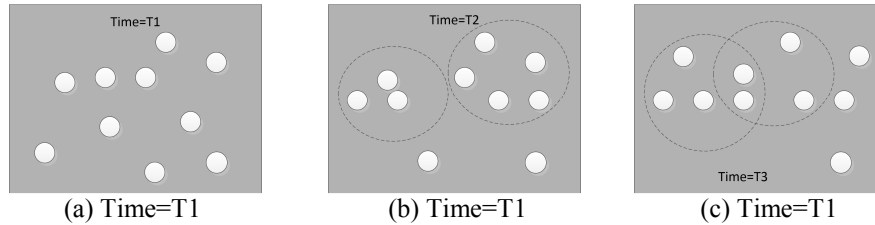


Fig. 1. The state of the cluster at different times.

In order to prove whether there is any intersection between the clusters, we use real data sets for DRAFT[2] to simulate, the experimental results are as follows. Through the experimental results we can see that with the change of time, the mean size of the cluster gradually stable, and the number of clusters that each node belongs to is also gradual stable. Which means that the intersections between clusters are existing in MONs. Therefore, in this paper, we use the intersections to determine the relations between nodes and design the message delivery strategy with the relations.

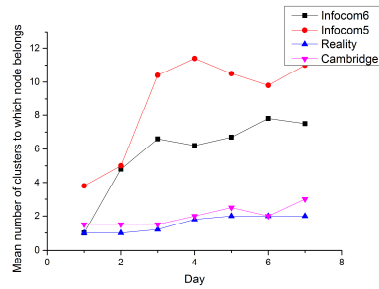


Fig. 2. The mean size of the clusters.

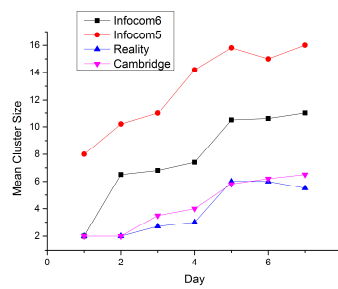


Fig. 3. The number of clusters that each node belongs to.

The rest of this paper is structured as follows. In Section 2, related work in the area of routings in MONs are discussed. In Section 3, definitions and computing methods are presented for the cluster building and updating. In section 4, the relations between nodes and the routing algorithm is discussed. In section 5, simulation results are presented. The last section concludes this paper.

2 Related Work

In MONs, the main direction of many researchers lies in the routing protocol for MONs. But many protocols only consider the “storage-carry-and-forward” data transmission mode in MONs, while some others take into account social factors in opportunistic routing due to human participation. Therefore, routing protocols in MONs can be mainly classified into two categories: social oblivious routings and social characteristics aware routings, and the division is based on whether or not to consider the social attributes in the routings.

For social oblivious routings, the authors proposed Epidemic routing algorithm in [3]. In their algorithm, the node who carry messages will flooded all the messages to all nodes that are within the communication range. This kind of message forwarding scheme greatly improves the efficiency of data transfer but consumes too much network resources, so it may not be useful in the real life. In [4], the Direct Transmission were proposed and it can be considered as the simplest routing algorithm in MONs, because in Direct Transmission, the messages carried by the source node can be only transferred to the destination node, which means the source node will keep moving until the destination node is met. Another classical routing is PROPHET[5], it uses the history of previous encounters with other nodes, and through the history PROPHET can get the delivery predictability and transitive property, which can help to calculate routing.

For social characteristics aware routings, SimBet[6] is a routing that exploits the exchange of pre-estimated “betweenness” centrality metrics and locally determined social “similarity” to the destination node. In [7], the authors thought that the frequent usage of repeated nodes considered as the most likely to deliver the messages could cause faster consumption of its battery resources, thus they proposed SOCLEER routing. SOCLEER is based on routing BubbleRap[8], but it could make a node traffic load redistribution when sending messages to other intermediary nodes which were not being so much demanded. [9] proposed a routing named EDR, EDR depends on two parameters: the number of encounters with other nodes and the distance between the source node and the destination node.

From the existing research works we can conclude that social relationships among nodes have a major impact on the performance of opportunistic routing, which is also proved to be true in [10,11]. In this paper, we get the way from the phenomenon in section 1 to find the friend circles (which we use clusters to replace in this paper) and intersections, then use the intersections to measure the relationship between nodes. To the best of our knowledge, there is no study has measure the relationship between nodes in MONs in the way we proposed in this paper.

3 Cluster Building And Updating

In this section, we will introduce how to build clusters (circles of friends) and how to update them. Since the cluster of nodes is a set of nodes that are closely related, the cluster is established based on the relationship between nodes. In previous work, many authors added the node to the cluster that was closely related, but didn't take in to account the relationship between the joined node and the nodes within the cluster. Therefore, in order to make the structure of the cluster more stable and less prone to large changes, there will be a pre-processing step before the node adding to the cluster. If the result of the pre-processing step is that the entire cluster structure is more stable, the node will be added to the cluster. In order to achieve this process, we made the following definition.

It is supposed that the node i has a cluster C_i , and C_i contains n nodes. In order to make the relationship more compact, we define the $Q(C_i)$ of cluster C_i to measure the relationship strength to ensure the cluster is always in a state of stable relationship.

$$Q(C_i) = \frac{\sum_{i,j \in C_i} r(ij)}{\sum_{i,j \in C_i} r(ij) + \sum_{k \in C_j} r(ik)} \quad (1)$$

Where, $r(ij)$ represents the number of encounter times between the node i and node j , and if they never met, the value is 0. The n represents the number of the nodes that cluster C_i

contains. The $r(ik)$ represents the number of encounter times between the node i and node k , node k belongs to cluster C_j , n_j is the number of the nodes that cluster C_i contains.

3.1 Building Clusters

We assume that when the network is initialized, the clusters of each node are empty. And each node will keep a table to record the number of encounters with other node, the table is used to record the historical data of the encounter times between current node and other nodes, the specific structure is as follows.

Table 1. Structure of encounter history.

Node_id	Encounter_Times
id1	1
...	...
id10	20

Where, Node_id represents the id of the node encountered. The Encounter_times represents the number of encounter times between the current node and the node with node id are Node_id. And if they have never met before, there will not be a record in the table.

With these prerequisites, we will simulate the scenarios for building clusters when the two nodes meet. It is supposed that node i meets node j at time t , and then the cluster of i will determine whether to add node j to the cluster according to the following steps.

- (1) The node i will determine whether they have met before, if it is, then go to the step 2. If they never met, the node i will record the encounter record into the table and set the Encounter times to 1, and stop the process.
- (2) Update the table and make the encounter times between node i and j plus 1. If the number of encounter times is less than the threshold, stop the process. If it is greater than the threshold, then go to the third step.
- (3) The third step is the pre-processing step. First, it will be determined whether node j belongs to C_i , and if node j belongs to cluster of node i , then it is ignored; if node j does not belong to node i , then node j is added to the local cluster, and then the $\Delta Q(j)$ will be calculated as follows.

$$\Delta Q(j) = Q(C + j) - Q(C). \quad (2)$$

If the value of $\Delta Q(j)$ is bigger than 0, it means that the cluster is more stable after the participation of node j , then the node j will be add to the cluster of node i . If the value is less than 0, then node j will be removed from the cluster and then stop the process.

3.2 Updating Clusters

Because of the random movement of nodes in MONs, the relationship is changing with time. In order to ensure the stability of the cluster, the cluster should update constantly. Here are some steps to update clusters.

We defined the clusters update every t time period, the specific updating steps are as follows:

- (1) From the network initialization, every t time period, the number of encounters in the record table multiplied by γ .

- (2) After multiplied by γ , if the value of encounter times is less than the threshold, the node will be added to the deletion list.
- (3) At the beginning of every t time period, the deletion list will update. The node in the deletion list will be preprocessed with the Equation 3. If $\Delta Q(k)$'s value of a node is greater than 0, then the node will be removed from the deletion list, otherwise, the node will be added to the cluster again.

$$\Delta Q(k) = Q(C) - Q(C - k). \quad (3)$$

Where k represents the node in deletion list.

4 Multi-Cluster based routing

In this section, we will use the clusters to measure the relationship between nodes and use the relationship to design the routing algorithm to make the message delivery more efficient.

4.1 Node relationship measurement using multi-clusters

The main purpose in MONs is to transfer the messages to the destination more correctly and more effectively. So it is useful to choose a proper relay node that has a good relationship with the destination node. Therefore, in this part, we are going to introduce the measurement of node relationship using multi-clusters between the node that the source node has met and the destination node.

As described above, each node may belong to many clusters, but the number of the clusters that one node belongs to is not certain, maybe it is one, or it is more than one. In order to determine the relationship between nodes more correctly, we divided the measurement into two part. The first part is for the situation that the destination node only belongs to one cluster (its own cluster), and the other part is for the situation that the destination node belongs to many clusters.

(1) The destination node only belongs to one cluster

When the destination node d only belongs to cluster C_d which is maintains by node d itself, then the relationship between node i and the node d can be divided in to two categories.

--If the node i is in the node d 's cluster C_d , it means the two node have a good relationship already, so the strength of the relationship is the encounter times between them.

--If the node i is not in the cluster C_d , then the relationship strength between the two nodes can be measured by the Equation(4) as follows.

$$w(id) = \frac{|C_i \cap C_d|}{|C_i \cup C_d|} \quad (4)$$

Where, $|C_i \cap C_d|$ means the number of common friends of node i and node d ; $|C_i \cup C_d|$ represents the number of all friends of two people.

(2) The destination node belongs to multiple clusters

When the destination node d belongs to multiple clusters, the clusters can not be ignored in the process of the measurement between node i and node d . Actually, in this situation, the measurement between them can be also divided into two categories: the node i belongs to the node d 's cluster C_d and node i is not in the C_d . But the first

category is the same with the part (1), we will not repeat them here. The rest situation is supposed that the destination node d belongs to multiple clusters, here we set the clusters are named C_1, C_2, \dots, C_k , then the relationship strength between node and the clusters can be measured by the formulas as follows.

$$w(i1) = \frac{|Cd \cap C1|}{cd} \times \frac{|Ci \cap C1|}{Ci \cap (C1 \cup C2 \cup \dots \cup Ck)} \quad (5a)$$

$$w(i1) = \frac{|Cd \cap C1|}{cd} \times \frac{|Ci \cap C1|}{Ci \cap (C1 \cup C2 \cup \dots \cup Ck)} \quad (5b)$$

$$w(ik) = \frac{|Cd \cap Ck|}{cd} \times \frac{|Ci \cap Ck|}{Ci \cap (C1 \cup C2 \cup \dots \cup Ck)} \quad (5c)$$

Where, $w(i1)$ and $w(ik)$ represents the relationship between node i and cluster C_1 and cluster C_k relative to the destination node respectively. And C_d represents the node d 's cluster and C_i represents the node i 's cluster. With the relationship between node i and clusters that destination node d belongs to, the relationship strength between node i and node d can be calculated by Equation (6).

$$w(id) = \frac{\sum_{j=1}^k w(ij)}{k} \quad (6)$$

Where, k is the number of clusters that the node d belongs to.

4.2 Routing Algorithm Based on Multi-Clusters

Here we propose the multiple clusters based routing algorithm (MCR), with the intention of bringing in a concise concept of multi-clusters in to MONs forwarding to achieve significant improvement of forwarding efficiency. The MCR uses the knowledge of cluster structure and the history of encountering times to make forwarding decision. As described above, there are two intuitions behind this algorithm. The first one is that each node maintains a cluster to store the nodes with good relationship, here we allow single node clusters to exist. And the second is that each node may belong to multiple clusters.

Forwarding in MCR is carried out as follows. If a node carries a message destined for another node, this node first transfer the message using the relationship strength between the destination node (which can be calculate with Equation (4) and formulaEquation (4)), until it reaches a node which is in the same cluster as the destination node. Within the cluster, the node will transfer the messages to the nodes that have encountered the destination node with more times. The detail forwarding process is shown as follows.

It is supposed that node i carries messages to forward and encounters node j at time t . If node j is the destination node of the message, then node i will deliver the message to node j and delete the delivered message in its buffer. Otherwise, the transmission will be going on as follows.

- (1) If the destination node is in both node j 's cluster and node i 's cluster, then if the node j met the destination node more times before, then node i will copy the message to node j . Otherwise, the node i won't transfer the message.
- (2) If destination node is in node j 's cluster and not in node i 's cluster, then node i will transfer the message to node j , and delete the message from its buffer to prevent further dissemination. Otherwise, the node i won't transfer the message.
- (3) If both the clusters of node i and node j don't contains the destination node, then the forwarding decision will be made by the relationship strength with the destination node of node i and node j . The calculation of the relationship strength between destination node and node i and j is shown in Equation (4) and (6). If the relationship strength between node j and the destination node is greater than that of node i , then

the message will be transferred to the node j from node i . Otherwise, the node i won't transfer the message.

5 Simulations

This section represents the experimental results obtained when running the MCR and several classical routings on real mobility traces. And we will evaluate MCR Routing and compare its performance to Epidemic, DRAFT and Bubble Rap with the results.

5.1 Simulation Setup

In order to obtain more accurate results, we have taken advantage of the real data set in the opportunistic network simulator. The simulation platform used in this paper is MobEmu [12], the same as [13]. MobEmu is an opportunistic network simulator that can replay the encounter trajectory of a node based on a data set and apply the required routing algorithm when the nodes meet, and finally collects and gives the statistics of the experimental results. The real datasets Infocom2006 is used for node activities driving, which can be downloaded from CRAWDAD[14].

In order to highlight the performance of MCR, we used two identical types of routing protocols, BubbleRap and DRAFT, and used a classical routing Epidemic in MONs as reference. We compare the performance of the routings in the same simulation environment. And the following performance metrics are used in the performance comparison..

- (1) **Hit Rate** : the ratio of the number of messages arrived at the destination to the total number of messages sent.
- (2) **Delivery Cost** : the ratio of the total number of messages exchanged to the number of messages generated.
- (3) **Delivery Latency** : the average amount of time passed when messages transmitted from source to destination.
- (4) **Hop Count** : the number of intermediate nodes in the shortest path from source to destination for a successful transmission.

5.2 Performance Comparison

This part shows the simulation results and the detailed analysis of these results. The experiments are mainly carried out from two aspects, the first aspect of the experiments is used to examine the performance of routings over time, and the second aspect is used to examine the performance of routings as the node storage capacity changes. In the first aspect of the experiments, the node storage capacity is set to 50, and the results are obtained over time. And in the second aspect of the experiments, the node storage capacity is not fixed, the simulation time of each experiment is set to the duration of the dataset.

5.2.1 Performance Comparison of routings over time

Figure 4 shows the results of the routings' performance over time on data set Infocom2006. From the pictures we can see, that the MCR has the highest message hit rate in the four routings, and performs better than others in the other three aspects. The reason of the

better performance may be because the MCR chooses the best nodes to build and update the clusters, and use the clusters to get accurately locate the identity of the relay node. And this lead to less waste of the network resources and higher efficiency of message delivery.

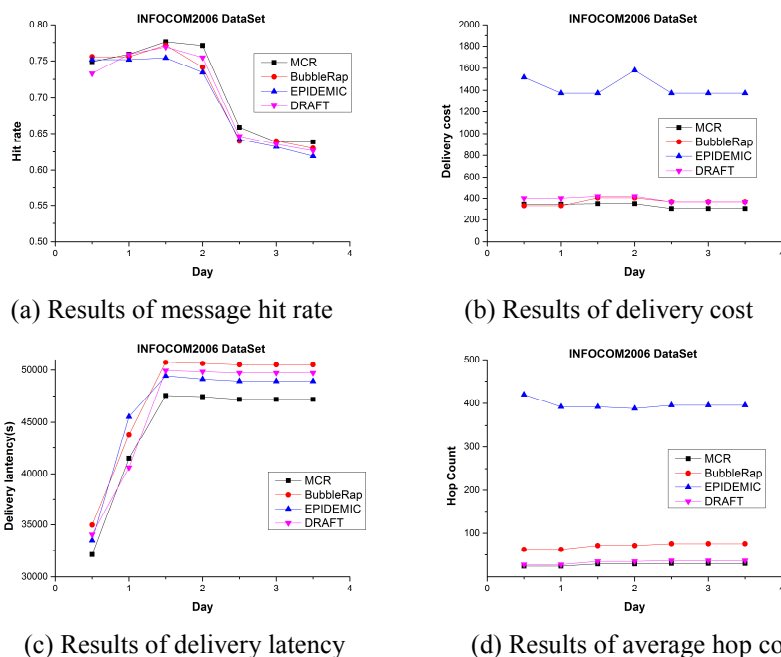
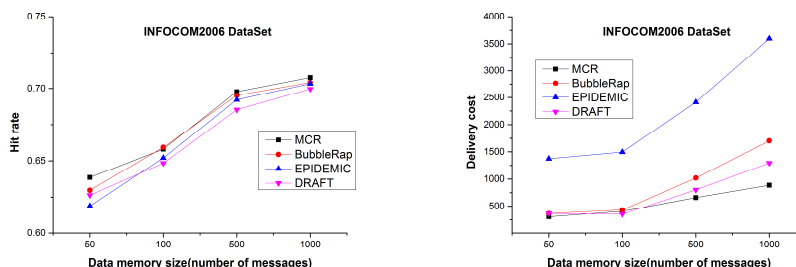


Fig. 4. The state of the cluster at different times.

5.2.2 Performance comparison with different node storage capacities

In this part, we carry out the experiments based on different node storage capacities to prove the stability and robustness of our routing under different data memory size and all the experiments are using the data set INFOCOM2006. Figure 5 shows the results of the experiment. The results show that our routing has the highest message hit rate and the lowest delivery cost and hop count. However, the delivery latency is not the best in the four routings. This may be because that in our routing, the relationship measurement have to get full knowledge of the nodes clusters and have to calculate the value, so the latency is higher in this dataset. But on the whole, our routing outperforms the other three routings.



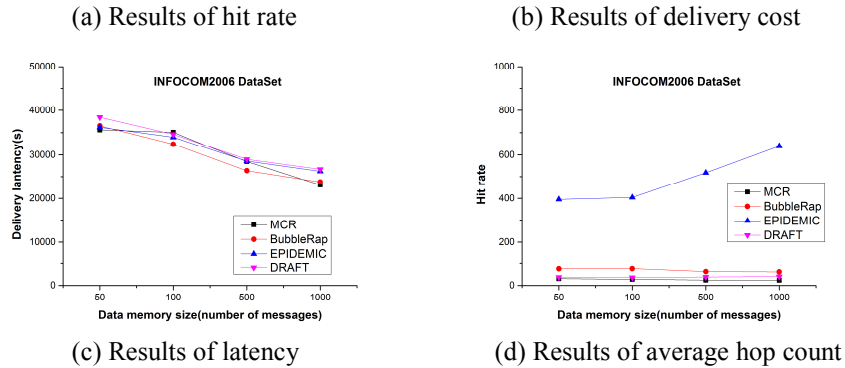


Fig. 5. Performance comparison with different node storage capacities.

6 Conclusions and future work

In this paper, we have shown that using multi-cluster to measure the relationship between nodes and deliver the messages can reduce the delivery cost and the number of hops in the message delivery, and message hit rate and delivery latency are not affected at the same time.

At first, we describe a phenomenon in real life, everyone has a circle of friends, but also exists in multiple friends circles. And we have experimented with the routing protocol in the published paper on the real dataset, confirming that this phenomenon is also existed in the opportunistic network and may have a role in determining the relationship between nodes. Then, inspired by this, we use the number of encounters as a basis to establish a circle of friends for each node and set up an update mechanism for the circles. After that, the multi-cluster has been formed with these friends circles. And we use it designed the node relationship measurement mechanism and routing algorithm MCR. Finally, we made some experiments to prove the effectiveness of our routing. Our experiments are mainly carried out from two aspects, the first aspect of the experiments are used to examine the performance of routings over time, and the second aspect are used to examine the performance of routings as the node storage capacity changes. The experiments results show that MCR not only outperforms BubbleRap routing, EPIDEMIC routing and DRAFT routing in delivery cost and average hop count, but also has a good performance in message hit rate and delivery latency. Therefore, using the multi-cluster to determine the node relationship and help delivery messages is useful in MONs.

For future work, we wish to design a more effective routing algorithm that the difference between the multiple clusters can be considered. For example, in the real life, we may belongs to different circles of friend, and the people in these circles may belongs to different types, such as family or colleagues. For this phenomenon, we want the routing can distinguish between different types of clusters or use it to set different priorities for the cluster to deliver the messages.

References

- [1] Fall K. A delay-tolerant network architecture for challenged internets[C]// ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 25-29, 2003, Karlsruhe, Germany. DBLP, 2003:27-34.
- [2] Orłowski M, Filer N. The rise and fall of spatio-temporal clusters in mobile ad hoc networks[J]. *Ad Hoc Networks*, 2013, 11(5):1641-1654.
- [3] Amin Vahdat and David Becker. Epidemic routing for partially connected ad hoc networks. Master Thesis, 2000.
- [4] Grossglauser M, Tse D N C. Mobility increases the capacity of ad hoc wireless networks[J]. *IEEE/ACM Transactions on Networking*, 2002, 10(4):477-486.
- [5] Lindgren A, Doria A, Schelén O. Probabilistic Routing in Intermittently Connected Networks[J]. *Sigmobile Mobile Computing & Communication Review*, 2003, 7(3):19-20.
- [6] Elizabeth M, Haahr. Social network analysis for routing in disconnected delay-tolerant MANETs[J]. *Mobi-Hoc 2007*, 2007:32-40.
- [7] Junior N M, Campos C A V. SOCLEER: A social-based energy-efficient forwarding protocol for opportunistic networks[C]// *Wireless Communications and Mobile Computing Conference*. IEEE, 2015:757-762.
- [8] Hui P, Crowcroft J, Yoneki E. BUBBLE Rap: Social-Based Forwarding in Delay-Tolerant Networks[J]. *IEEE Transactions on Mobile Computing*, 2011, 10(11):1576-1589.
- [9] Dhurandher S K, Borah S, Woungang I, et al. EDR: An Encounter and Distance Based Routing Protocol for Opportunistic Networks[C]// *IEEE, International Conference on Advanced Information NETWORKING and Applications*. IEEE, 2016:297-302.
- [10] Mei A, Morabito G, Santi P, et al. Social-aware stateless forwarding in pocket switched networks[C]// *IEEE INFOCOM*. IEEE, 2015:251-255.
- [11] Costa P, Mascolo C, Musolesi M, et al. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks[J]. *IEEE Journal on Selected Areas in Communications*, 2008, 26(5):748-760.
- [12] Ciobanu R I, Dobre C, Cristea V. Social aspects to support opportunistic networks in an academic environment[C]// *International Conference on Ad-Hoc, Mobile, and Wireless Networks*. Springer-Verlag, 2012:69-82.
- [13] Ciobanu R I, Marin R C, Dobre C, et al. ONSIDE: Socially-aware and Interest-based dissemination in opportunistic networks[C]// *Network Operations and Management Symposium*. IEEE, 2014:1-6.
- [14] CRAWDAD. Available online: <http://www.crowdad.org>.