An Adaptive Routing Algorithm for Mobile Delay Tolerant Networks

Jingwei Miao, Omar Hasan, Sonia Ben Mokhtar, Lionel Brunie
University of Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
{jingwei.miao, omar.hasan, sonia.benmokhtar, lionel.brunie}@insa-lyon.fr

***Abstract*—Delay tolerant networks (DTNs) are wireless mobile networks in which the existence of an end-to-end path from the source to the destination of a message cannot be guaranteed. This makes message delivery as one of the major challenges in DTNs. Recent studies based on real world traces show that nodes in DTNs exhibit mobility properties such as their centrality in the network or regularity patterns. To the best of our knowledge, existing routing algorithms exploit only some of the nodes mobility properties (e.g., only nodes centrality, or only nodes regularity) while excluding the others. We present in this paper the first dynamic routing algorithm in DTNs that exploits the most appropriate mobility property (among which node centrality and regularity) at the specific time and location. Our algorithm dynamically learns nodes mobility properties in order to appropriately select the best route to the destination on a per-node and per-situation basis. Simulations performed on real mobility traces show that our algorithm has a better delivery ratio and a lower overhead than existing state-of-the-art routing algorithms that rely on a single mobility property.**

***Index Terms*—delay tolerant networks, mobility, routing, centrality, regularity.**

I. Introduction

Delay Tolerant Networks (DTNs) are wireless mobile networks in which an end-to-end routing path cannot be assumed to exist between the source and the destination of a message [1]. This makes message delivery as one of the major challenges in these networks. In order to deal with the lack of end-to-end connectivity between nodes, routing is often performed in a "store and forward" way, where a message is stored by intermediary nodes and forwarded to nodes closer and closer to the destination until the latter is eventually reached.

In order to maximize the chances of reaching the destination, the first routing algorithms in DTNs relied on flooding the network with copies of the same message [2]. Followed algorithms that try to limit the number of copies of the same message in the network [3]. Although they have a high delivery ratio, algorithms based on flooding have a high overhead, which undesirably exhaust mobile node resources (e.g., battery, bandwidth) and generate unnecessary contention.

In order to better choose intermediary nodes and thus reduce the routing overhead, a number of studies of real mobility traces have been carried out [4]–[6]. These studies show that the mobility of nodes is influenced by their owner's social relationships. A number of social properties characterizing nodes' mobility have thus been defined, which can be classified into three categories: 1) *Centrality*, indicates the relative importance of a node in a network. For instance, *betweenness*, which is a type of centrality, measures the number of times a node falls on the shortest path between two other nodes [7]. 2) *Regularity*, expresses the probability of a given event (e.g., an encounter between two nodes at a given time slot) to be repeated over time. 3) *Community*, is traditionally defined as a group of interacting people co-existing in a common location. People in a community are believed to have a high probability to meet each other [8].

Building on these observations a number of routing algorithms have been proposed in the literature. Among these algorithms, RANK [9] relies on node centrality, Habit [10] builds on regularity, SimBet [11] utilizes node similarity[1] and betweenness and BubbleRap [8] utilizes nodes communities and betweenness.

A major drawback of these approaches is that they assume that a given node in the network has the same social properties all over time. Reality is different. Indeed, a node can be in/out a community during specific periods of time, it may have a central position in the network or be completely isolated at given times and may exhibit a regular or a completely irregular mobility pattern during specific times of the day/specific days of the week. In order to leverage this dynamics, we present the first routing algorithm that dynamically adapts to the user's social properties at the very specific time and location. In this paper, we focus on two social properties: node centrality and regularity. Our algorithm firstly exploits the contact history between nodes to estimate the delivery latency and overhead of a centrality-based and a regularity-based routing algorithm. It then selects the route that has the lowest estimated latency and overhead among the routes provided by the two algorithms.

Summarizing, the contributions of this paper are three fold:

1) We demonstrate the drawbacks of algorithms that are based only on one social property by studying the algorithms based on centrality and regularity (Section II).
2) We present an abstract model for delay tolerant networks. Using this model we generalize centrality-based and regularity-based routing algorithms (Section III).
3) We propose an adaptive routing algorithm for delay tolerant networks, which dynamically exploits nodes centrality and regularity according to the specific situation of the user (IV).

The work presented in this paper complements the work in-

[1] Similarity in the context of SimBet is defined as the number of nodes that a given node and the destination node have both encountered.

troduced earlier in [12]. In this paper, we conduct experiments on the MIT Reality dataset [13] (Section V), whereas in [12] we conducted experiments on the Cambridge Haggle dataset [14]. The results of experiments on both datasets confirm that our algorithm has a better delivery ratio and a lower overhead as compared to state-of-the-art algorithms that rely on a single mobility property (i.e., RANK and Habit).

II. Problem Description

In this section, we use two examples to demonstrate the problems faced by algorithms based only on either centrality or regularity.

A. The Problem of Centrality-based Algorithms

The routing mechanism of centrality-based algorithms is to forward a message to intermediate nodes having higher centrality than the current node in the hope that the destination node will be reached. The issue with this approach is that a node cannot forward a message to the destination node through intermediate nodes that have lower centrality than the current node. These intermediate nodes with low centrality may have high probability of encountering the destination node in the future, however, this characteristic is ignored by centrality-based algorithms.

B. The Problem of Regularity-based Algorithms

Before we discuss the drawback of such algorithms, let us explain the concept of regularity in further detail. As discussed before in the introduction, the regularity of a node is defined as the probability that two nodes meet each other in a given time slot over a given time length. For example, time slots can be considered as 4 hour intervals and the time length can be considered as a week. Assume the duration of the contact history is 10 weeks. In the contact history, node A has met node B for 7 times in the time slot from Mon. 8 AM to Mon. 12 PM. In this case, the regularity between node A and node B from Mon. 8 AM to Mon. 12 PM is 0.7. Each node contains a regularity table that describes the regularity between it and its fellow nodes in given time slots. The regularity table is constructed by tuples which contain the time slot and the regularity. The number of such tuples is the ratio of the time length (e.g., a week, a month) divided by the size of the time slot.

TABLE I
TIME SLOT IS 2 HOURS, AND THE TIME LENGTH IS A WEEK

Time slot	A $\leftrightarrow$ B	A $\leftrightarrow$ C	B $\leftrightarrow$ D	C $\leftrightarrow$ D
Mon.[8 AM, 10 AM)	0.7	0.6	0.5	0.3
Mon.[10 AM, 12 PM)	0.1	0.2	0.6	0.4
…	…	…	…	…

TABLE II
TIME SLOT IS 4 HOURS, AND THE TIME LENGTH IS A WEEK

Time slot	A $\leftrightarrow$ B	A $\leftrightarrow$ C	B $\leftrightarrow$ D	C $\leftrightarrow$ D
Mon.[8 AM, 12 PM)	0.8	0.6	0.6	0.4
Mon.[12 PM, 4 PM)	0.1	0.2	0.1	0.4
…	…	…	…	…

The routing process of regularity-based algorithms is described as follows. If two nodes meet each other frequently, they are considered as friends and they exchange their regularity tables with each other. On the other hand, two nodes who do not meet each other frequently are considered as strangers so they do not exchange their regularity tables. A node can use its regularity table and the regularity tables of its friends to construct a regularity graph. Using these regularity graphs, regularity-based algorithms try to find a path with an optimal delivery probability to forward a message to the destination. We give an example to demonstrate this routing process and its drawbacks. To clearly exhibit the delivery process of a message, regularity tables of the current node's (node A) friends are merged into the regularity table of node A (See Table I and II). "A $\leftrightarrow$ B" means that nodes A and B meet each other. The numbers in the cells represent the regularity between two nodes in a time slot. For instance, the regularity between node A and node B in time slot from Mon. 8 AM to Mon. 12 PM is 0.8. The minimum regularity in a path is used to express the delivery probability. At Mon. 8 AM, node A generates a message whose destination is node D, and time-to-live (TTL) is 5 hours. When two hours and one week are selected as the size of the time slot and the time length, the content of the regularity table in node A is shown in Table I. Based on this regularity table, node A selects the path A $\rightarrow$ B $\rightarrow$ D, since the delivery probability of this path is the best which is 0.6. When the size of the time slot is changed to be 4 hours, the content of the regularity table in node A is shown in Table II. Therefore, node A selects the path A $\rightarrow$ C $\rightarrow$ D, whose delivery probability is 0.4, to delivery the message. To summarize this example, the size of the time slot and the time length strongly influence the performance of such algorithms. Moreover, if the destination of a message is out of the regularity graph of a node, the node cannot construct a path to deliver the message. Therefore, the delivery ratio of regularity-based algorithms is affected by the above factors.

III. The ARo Adaptive Routing Algorithm

In this section, firstly, we discuss our hypothesis. Secondly, we construct the network model used in this paper. Thirdly, we develop generalized models of centrality-based algorithms and regularity-based algorithms. We develop these models in order to calculate the expected delivery performance (e.g., delivery latency, delivery cost) for any type of algorithm. Finally, based on the above two models, we propose our adaptive routing algorithm, which takes advantage of the characteristics of the above two types of algorithms to improve routing performance.

The idea of our algorithm is to select the algorithm which is the best-adapted for the actual situation. Firstly, the algorithm exploits the contact history between nodes to calculate the expected values of the routing performance metrics (e.g., delivery latency, delivery cost, etc), based on our generalized models of centrality-based and regularity-based algorithms. Then, the algorithm compares the expected values of the metrics to select the best algorithm to route a message.

A. Our Hypothesis

The above two types of algorithms exploit only one social property to forward a message. However, a node can have more than one social property (e.g., centrality and regularity) at the same time. An algorithm that exploits multiple properties can avoid drawbacks associated with algorithms based on only one property. Based on this observation, we propose to investigate the following hypothesis: "an adaptive routing algorithm that can switch between centrality-based and regularity-based algorithms can provide better routing performance".

B. Delay Tolerant Network Model

Some recent research works [9] [10] show that the contact between nodes in DTNs is not random but follows patterns which are repetitive to a certain extent. Therefore, the contact history of a node can be exploited to predict its future contacts. Inspired by Jain et al. [15] and Hossmann et al. [16], we integrate the contact history into our model of DTNs. The elements of our model are described as follows:

Nodes and Edges. Let $V = \{v_1, \ldots, v_n\}$ be the set of all the nodes of a network. An edge e_{ij} exists between two nodes v_i and v_j (where, $1 \leq i \leq n$, $1 \leq j \leq n$, $i \neq j$), if they have contacted each other at least once. The inter-contact time between two nodes is the time interval between two successive contacts. The weight of the edge e_{ij} denoted as w_{ij} is the mean of all instances of inter-contact times between the two nodes. A DTN is represented as the undirected weighted graph $G(V, E)$.

Message. A message can be considered as a tuple (v_s, v_d, t, l), where v_s is the source node, v_d is the destination node, t is the time stamp of creation and l is the time-to-live (TTL).

Routing Algorithm. A routing algorithm for the DTN is responsible for routing a message from its source node to its destination node via intermediate nodes within the given TTL in the absence of an end-to-end path between the source and the destination.

C. A Generalized Model of Centrality-based Algorithms

As mentioned in the introduction, centrality is a metric that calculates the relative importance of a node in a network. Centrality-based algorithms [9] [8] always forward a message from a node with lower centrality to a node with higher centrality in the hope that the destination will be reached. We develop a generalized model of these centrality-based algorithms. This model will allow us to calculate the expected delivery performance metrics (e.g., delivery latency, delivery cost) of these centrality-based algorithms for a given message.

We utilize a vector of nodes to denote a path in $G(V, E)$. The weight of a path is the sum of the weights of the edges that form the path. $wgt(h)$ denotes the weight of path h.

Let $h^l(v_s)$ be any path which originates with v_s such that the weight of the path is no greater than l. Every node in $h^l(v_s)$ has a higher centrality than the preceding nodes in the path.

Let $h^l(v_s, v_d)$ be any path $h^{l'}(v_s), v_d$, where $l > l'$. That is, any path $h^{l'}(v_s)$ followed by the node v_d. In a path $h^l(v_s, v_d)$, the centrality of v_d may be lower than its previous node. However, the condition that each node has higher centrality than its preceding nodes still holds for the path $h^{l'}(v_s)$.

In a network, it is possible that more than one such path exists. Let $H^l(v_s)$ be the set of all possible paths $h^l(v_s)$. Let $H^l(v_s, v_d)$ be the set of all possible paths $h^l(v_s, v_d)$. The dissemination of a message m in the centrality-based algorithms that we consider [9] [8], always follows the shortest path in terms of edge weights from source node v_s to a destination node v_d with time-to-live l. If there is no path from v_s to v_d within l, the expected delivery latency can be considered as infinite. Otherwise, the expected delivery latency is the weight of the path. Thus, the expected delivery latency of the message can be expressed as Equation 1. The subscript c indicates the centrality-based algorithms.

$$Lat_c(v_s, v_d, l) = \begin{cases} +\infty, & \text{if } H^l(v_s, v_d) = \emptyset \\ \min wgt(h), & h \in H^l(v_s, v_d) \end{cases} \quad (1)$$

The expected delivery cost of the routing process for the message can be considered as the number of copies of the message in the network at the time when the TTL for the message expires. Let $N(H^l(v_s))$ be the set of all the nodes in all the paths in the set $H^l(v_s)$ (see Equation 2). Thus the expected delivery cost for delivering the message m in centrality-based algorithms can be expressed as Equation 3.

$$N(H^l(v_s)) = \{v | v \textit{ is a node in } h, \textit{ and } h \in H^l(v_s)\} \quad (2)$$

$$Cost_c(v_s, v_d, l) = |N(H^l(v_s)) - \{v_d\}| \quad (3)$$

D. A Generalized Model of Regularity-Based Algorithms

Regularity-based algorithms [10] [17] always forward a message along the path which can achieve the best delivery probability. We develop a generalized model of these centrality-based algorithms. This model will allow us to calculate the expected delivery performance metrics (e.g., delivery latency, delivery cost) of these centrality-based algorithms for a given message.

Let $p^u(v_i, v_j)$ be the regularity between two nodes v_i and v_j in a given time slot u. If the maximum regularity between two nodes is greater than a threshold δ, they can be considered as friends; otherwise, they are strangers and remove the edge between them. Each node contains a regularity table which describes the regularity between it and its friends.

We utilize a vector to denote a path in $G(V, E)$. The time slot of two adjacent nodes in a path increases along with the index of the node in the path. Since the regularity between two nodes is different in different time slots, the paths constructed to deliver a message are different. Let $k^l(v_s, v_d, u)$ be any path from v_s to v_d, which starts in the time slot u of the creation time of the message m. In a network, it is possible that more than one such path exists, Let $K^l(v_s, v_d, u)$ be the set of all possible paths $k^l(v_s, v_d, u)$. The expected delivery probability of a path is expressed as the minimum regularity in the path. Let $k_b^l(v_s, v_d, u)$ be the path which can achieve the best expected delivery probability. If the path $k_b^l(v_s, v_d, u)$ does not exist, the expected delivery latency can be considered as infinite. Otherwise, the expected delivery latency is the weight of the path. Thus, the expected delivery latency of the message can be expressed as Equation 4. The subscript r indicates the regularity-based algorithms.

$$Lat_r(v_s, v_d, u, l) = \begin{cases} +\infty, & \text{if } K^l(v_s, v_d, u) = \emptyset \\ wgt(k), & k = k_b^l(v_s, v_d, u) \end{cases} \tag{4}$$

The expected delivery cost of the routing process for the message can be considered as the number of copies of the message in the network at the time when the TTL for the message expires. Thus the expected delivery cost for delivering the message m can be expressed as Equation 5.

$$Cost_r(v_s, v_d, u, l) = |k_b^l(v_s, v_d, u)| - 1 \tag{5}$$

IV. The Work flow of Our Adaptive Routing Algorithm

In this section, we exploit the expected routing performance metrics for the models of centrality-based and regularity-based algorithms to propose our adaptive routing algorithm for delay tolerant networks. We call our algorithm the *ARo* (Adaptive Routing) algorithm, pronounced as "arrow".

The objective of the *ARo* algorithm is to select the best routing algorithm (from centrality-based and regularity-based algorithms) for the given message. The ARo algorithm uses the generalized models that we have developed to estimate the expected routing performance of the centrality-based and regularity-based algorithms. When a message $m(v_s, v_d, t, l)$ is created, the following two steps are executed:

1) $\alpha = SelectAlgorithm(v_s, v_d, u, l, \varepsilon)$
2) $ExecuteAlgorithm(m, v_s, v_d, \alpha)$

1) $SelectAlgorithm(v_s, v_d, u, l, \varepsilon)$: The goal of this function is to select the algorithm which can provide the best delivery performance for a message. It calculates the expected delivery performance (e.g., delivery latency, delivery cost) based on the developed models. By comparing these expected delivery performance parameters, it selects the algorithm which can achieve the best delivery performance. There are two intuitions behind this function. Firstly, the messages are hoped to be delivered as soon as possible. Thus, this function selects the algorithm which can achieve the shortest delivery latency. If the gap between these delivery latencies don't exceed the threshold ε, they are considered approximately the same. Secondly, the algorithm which can achieve the lowest cost is preferred. Thus, when two algorithms can achieve approximately same delivery latency, this function selects the algorithm which assumes the lowest resources in terms of the copies of messages created. This function returns the name of the selected algorithm, which will be added into a message as the message header. The symbols for centrality-based and regularity-based algorithms are α_c and α_r. The pseudo code of the function is listed as follows (see Algorithm 1).

Algorithm 1 $SelectAlgorithm(v_s, v_d, u, l, \varepsilon)$

$\alpha \leftarrow \alpha_c$
if $Lat_r(v_s, v_d, u, l) < Lat_c(v_s, v_d, l) - \varepsilon$ **then**
    $\alpha \leftarrow \alpha_r$
else if $Lat_r(v_s, v_d, u, l) > Lat_c(v_s, v_d, l) + \varepsilon$ **then**
    $\alpha \leftarrow \alpha_c$
else
    if $Cost_r(v_s, v_d, u, l) < Cost_c(v_s, v_d, l)$ **then**
        $\alpha \leftarrow \alpha_r$
    else
        $\alpha \leftarrow \alpha_c$
    end if
end if
return α

2) $ExecuteAlgorithm(m, v_s, v_d, \alpha)$: Once the algorithm α is selected by the previous step, the source node v_s executes the selected algorithm to route the message to the destination node v_d. Each intermediate node only extracts the name of the algorithm from the message header and executes the selected algorithm to route the message.

V. Simulation and Results

In this section, a trace from the real world is used to compare the routing performance (i.e., delivery ratio and delivery cost) of our routing algorithm and two state of the art routing algorithms which represent the centrality-based and regularity-based algorithms respectively.

A. Simulation Setup

To evaluate our algorithm, we used real trace-driven simulations based on MIT Reality data [13]. This data consists of the contacts for 97 smart phones which were carried by students and staffs at MIT over nine months. In our simulations, we used the contacts from September 1^{st} to December 1^{st}. Since it is the time of the first academic semester, human relationships are relatively stable.

In this experiment, each simulation is repeated 20 times with different random seeds for statistical confidence. Since 4 hours are selected as the size of time slot and the contact history is mapped into a week plan, there are 42 time slots in each simulation. The threshold ε is 12 minutes. At the beginning of each time slot, 5% nodes are randomly chosen as the source nodes, and each source node sends messages to other nodes. A message contains the identifiers of the source and the destination nodes, the start time and a given TTL. Thus, there are 16128 messages created for each simulation.

B. Metrics & Routing Algorithms

For all the simulations we have conducted for this work, we have measured the following metrics:

Delivery ratio: The proportion of messages that have been delivered out of the total unique messages created.

Delivery cost: The total number of messages (incl. duplicates) transmitted in the simulation. To normalize this, we divide it by the total number of unique messages created.

We compare ARo against RANK and Habit which represent the centrality-based and regularity-based algorithms.

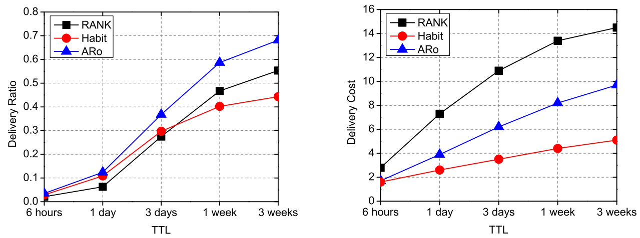

Fig. 1. Delivery ratio (left) and delivery cost (right) comparison of several algorithms on reality data set

RANK: A node forwards a message to the destination node or intermediate nodes whose centrality are higher than its. The C-Window strategy is used to calculate the centrality of a node. It cumulates the unique people encountered by a node in the previous time windows whose sizes are 4 hours.

Habit: The source of a message uses its regularity table to calculate the path which can achieve the best delivery probability. The size of the time slot and the time length to calculate the regularity are 4 hours and 1 week respectively. The threshold δ is 0.3. The minimum regularity in a path is employed to denote the delivery probability of the path.

C. Simulation Result

The delivery ratios of these algorithms increase, as TTL becomes longer. With a longer TTL, the messages which need long latency can be delivered (see Fig.1). When TTL is shorter than three days, the delivery ratio of Habit is better than that of RANK. The reason is that with a short TTL there are few paths, which start with a source node, for RANK, thus the drawback of centrality-based algorithms is prominent in such case. However, the regularity-based algorithms can still deliver messages in such case. When TTL is longer than one week, the delivery ratio of Habit is not as good as that of RANK. The reason is that RANK exploits much more paths to deliver a message than Habit does, and the drawback of centrality-based algorithms is not prominent in such case. The delivery ratio of our algorithm is always better than those of other algorithms. When TTL is three days, our algorithm can achieve about 8% delivery increment than Habit does. When TTL is three weeks, our algorithm can achieve about 13% delivery increment than RANK does.

The delivery cost of ARo is higher than Habit but much lower than RANK. When TTL is three days, ARo can achieve about 6.0 delivery cost decrement than RANK does. When TTL is three weeks, ARo can achieve about 4.85 delivery cost decrement than RANK does. Since more messages can be delivered by RANK than by Habit when TTL is longer than three days, ARo selects RANK to delivery such messages. Moreover, RANK exploits much more intermediate nodes to deliver a message than Habit does. These result in that the delivery cost of ARo increases quickly when TTL is longer than three days.

These results show that ARo can exploit the advantage of a type of algorithms to complement the drawback of another type of algorithms to improve the routing performance. These studies we have performed support our hypothesis.

VI. Conclusion and Future Work

In this paper we have demonstrated the need for an adaptive routing algorithm in delay tolerant networks. We presented the *ARo* algorithm which selects the best routing algorithm according to the given situation. The simulation results support our hypothesis that an adaptive routing algorithm that can switch between centrality-based and regularity-based algorithms can provide better routing performance.

As future work, we would like to extend our algorithm such that the best routing algorithm can be selected not only at the source node but at each intermediate node as well. This may further improve delivery performance. Another direction for future work is to add the feature of community-based routing to our adaptive routing algorithm.

Acknowledgments

This work is partially supported by the China Scholarship Council (CSC) UT-INSA PhD program, the MDPS German-French Doctoral College and the French National Research Agency (SocEDA, Grant ANR-10-SEGI-013).

References

[1] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *SIGCOMM ’03*, pp. 27–34.
[2] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” Citeseer, Tech. Rep., 2000.
[3] T. Spyropoulos, K. Psounis, and C. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in *SIGCOMM ’05*, pp. 259–266.
[4] M. González, C. Hidalgo, and A. Barabási, “Understanding individual human mobility patterns,” *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
[5] M. Musolesi and C. Mascolo, “A community based mobility model for ad hoc network research,” in *REALMAN ’06*. ACM, pp. 31–38.
[6] C. Boldrini and A. Passarella, “Hcmm: Modelling spatial and temporal properties of human mobility driven by users’ social relationships,” *Computer Communications*, vol. 33, no. 9, pp. 1056 – 1074.
[7] L. Freeman, “A set of measures of centrality based on betweenness,” *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.
[8] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: social-based forwarding in delay tolerant networks,” *TMC*, 2010.
[9] P. Hui and J. Crowcroft, “Predictability of human mobility and its impact on forwarding,” pp. 543–547, 2008.
[10] A. J. Mashhadi, S. B. Mokhtar, and L. Capra, “Habit: Leveraging human mobility and social network for efficient content dissemination in Delay Tolerant Networks,” in *10th WOWMOM*. IEEE, 2009, pp. 1–6.
[11] E. M. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” in *MobiHoc ’07*, 2007, pp. 32–40.
[12] J. Miao, O. Hasan, and L. Brunie, “Leveraging node centrality and regularity for efficient routing in mobile peer-to-peer networks,” in *4th Globe*, ser. LNCS, 2011, vol. 6864, pp. 83–94.
[13] N. Eagle and A. S. Pentland, “CRAWDAD data set mit/reality (v. 2005-07-01),” Downloaded from http://crawdad.cs.dartmouth.edu/mit/reality.
[14] J. Scott, R. Gass, J. Crowcroft, A. Chaintreau, P. Hui, and C. Diot, “CRAWDAD trace cambridge/haggle/imote/content (v. 2006-09-15),” Downloaded from http://crawdad.cs.dartmouth.edu/cambridge /haggle/imote/content.
[15] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” *SIGCOMM Comput. Commun. Rev.*, vol. 34, pp. 145–158, 2004.
[16] T. Hossmann, T. Spyropoulos, and F. Legendre, “Know thy neighbor: towards optimal mapping of contacts to social graphs for DTN routing,” in *INFOCOM ’10*. IEEE, 2010, pp. 1–9.
[17] L. H. Vu, “Characterizing and leveraging people movement for content distribution in mobile peer-to-peer networks,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2010.