Leveraging Node Centrality and Regularity for Efficient Routing in Mobile Peer-to-Peer Networks

Jingwei Miao, Omar Hasan, and Lionel Brunie

University of Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France
{jingwei.miao, omar.hasan, lionel.brunie}@insa-lyon.fr

Abstract. Solutions for data dissemination in traditional peer-to-peer networks are not suitable for mobile peer-to-peer networks due to the special characteristics of mobile environments, particularly highly variable connectivity, and disconnection. Mobile peer-to-peer data dissemination over Delay Tolerant Networks (DTNs) is a promising paradigm since they can tolerate frequent and long network partitions. DTNs exploit collaborative data storage and node mobility to bridge disconnected nodes and enable communication between them. Recent studies based on real world traces reveal that node mobility exhibits certain patterns influenced by the centrality and the regularity of nodes in the network. Many existing routing algorithms for DTNs exploit only one of the mobility properties, e.g., only node centrality, or only node regularity to route messages from a source node to a destination node. In this paper, we present an adaptive routing algorithm that exploits either centrality or regularity according to the situation to achieve the best possible routing performance in delay tolerant networks. Simulations performed on real mobility traces show that our algorithm outperforms the existing routing algorithms that utilize only one mobility property.

Keywords: mobile computing, peer-to-peer, delay tolerant networks, routing, centrality, regularity.

1 Introduction

A mobile peer-to-peer (P2P) network is composed of a finite set of mobile peers that communicate with each other via short range wireless protocols, such as IEEE 802.11, Bluetooth, Zigbee, or Ultra Wide Band (UWB) [1]. The special characteristics of mobile environments, such as highly variable connectivity, disconnection, location-dependency, energy and resource sensitivity, and the diversity in wireless networks as well as carrier-grade performance requirements bring new challenges for research in mobile P2P computing [2]. Routing of information from a source node to a destination node in mobile P2P networks is particularly challenging due to these characteristics since traditional routing algorithms assume the existence of an end-to-end path. Using Delay Tolerant Networks (DTNs) as a foundation for mobile P2P networks has been proposed as one of the directions for improving the performance of algorithms in mobile P2P networks [1]. In [3], Fall first proposed delay tolerant networks, which can tolerate frequent and long network partitions. DTNs exploit collaborative data storage and node mobility to bridge disconnected nodes and enable communication between them. For example, a message can be forwarded to intermediate nodes instead of requiring an end-to-end path to the destination. The intermediate nodes then forward the message closer to the destination. When the destination is within the transmission range of one of the intermediate nodes, the routing process can be achieved.

There are a number of works in the literature on routing in DTNs. In order to maximize the chances of reaching the destination, the early routing algorithms in DTNs relied on flooding the network with copies of the same message [4]. Although they have a high delivery ratio, algorithms based on flooding have a high overhead, which undesirably exhausts mobile node resources (e.g., battery, bandwidth) and generates unnecessary contention [5]. In order to minimize the resources consumed in the routing process, other algorithms such as *Direct Transmission* and *Seek and Focus* [6] only utilize one copy of a message in the entire network. The main shortcomings of this kind algorithm are low delivery success rate and high delivery delay [7].

In order to make a tradeoff between delivery rate and overhead, a number of studies focus on the analysis of real mobility traces to rationally choose the intermediary nodes [8] [9] [10] [11] [12]. These studies show that the mobility of nodes is influenced by their owner's social relationships, and exhibits a high degree of repetition. 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 [13]. 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 [14].

Building on these observations a number of routing algorithms have been proposed in the literature. Among these algorithms, RANK [15] relies on node centrality, Habit [16] builds on regularity, and BubbleRap [14] utilizes node community and centrality.

A major drawback of these approaches is that they assume that a given node in the network has the same social properties all the time. Reality is different. For example, a node can be in/out of 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. Moreover, a node can have more than one social property (e.g., centrality and regularity) at the same time. For instance, the secretary of a department can have high centrality and regularity at the same time. She can have high centrality because of interactions with a large number of coworkers. She may also exhibit high regularity due to frequent inter-departmental meetings. In order to leverage these dynamics, we present the first routing algorithm that dynamically adapts to the user's social properties. 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.

The remainder of this paper is organized as follows. In section 2, two examples are given to demonstrate the problems of algorithms based only on one of the properties of centrality and regularity. In section 3, we firstly present an abstract model for delay tolerant networks. Using this model we generalize centrality-based and regularity-based routing algorithms. Based on the above two models, 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. A comparison between our algorithm and other algorithms is conducted in section 4. Finally, section 5 concludes this paper and describes our future work.

2 Problem Description

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

2.1 The Problem of Centrality-based Algorithms

The routing mechanism of centrality-based algorithms (e.g. [15] [14]) 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.

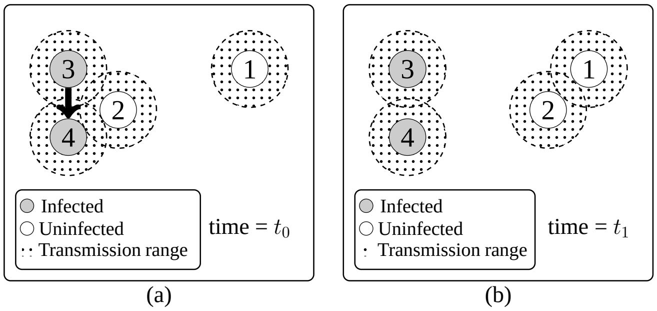

Fig. 1. An example where algorithms based on centrality fail

An example to demonstrate this phenomenon is described as follows (See Fig.1). The identifier of a node also indicates its centrality. For example, the centrality of node 3 is 3. The grey node means that it has already taken the given message. The white node means that it has not taken the message. The dotted region indicates the transmission range. If a node is within the transmission range of another one, it can receive messages from that one. The arrow indicates the transmission direction of the message. Initially, node 3 generates the message whose destination is node 1. Node 3 meets node 4 and node 2 at time t_0. Since the centrality of node 3 is lower than that of node 4, it forwards a copy of the message to node 4 instead of node 2. In this example, node 2 has a higher probability of meeting node 1 in the future than node 4. At time t_1 (which is later than t_0), node 2 meets node 1. The delivery of the message fails, since node 2 does not get a copy of the message, even though it has a high probability of meeting the destination. The example shows that the delivery ratio of centrality-based algorithms is affected by the above phenomenon.

2.2 The Problem of Regularity-based Algorithms

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 length of time. For instance, time slots can be considered as 4 hour intervals and the length of time can be considered as a week. Let's assume that 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 1. 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 2. 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 (e.g. [16] [17] [18]) 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 the routing process based on regularity and its drawbacks. To clearly exhibit the delivery process of a message, regularity tables of the current node's (node A's) friends are merged into the regularity table of node A (See Table 1 and Table 2). "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 1. 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 2. 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.

3 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 these types of algorithms. 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.

3.1 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”.

3.2 Delay Tolerant Network Model

Some recent research works [15] [16] 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. [19] and Hossmann et al. [20], 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.

3.3 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 [15] [14] 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. $weight(h)$ 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 [15] [14], 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.

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

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.

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

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

3.4 A Generalized Model of Regularity-Based Algorithms

Regularity-based algorithms [16] [17] [18] 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.

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

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.

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

3.5 The Workflow 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. $\alpha = SelectAlgorithm(v_s, v_d, u, l)$
2. $ExecuteAlgorithm(m, v_s, v_d, \alpha)$

$\boldsymbol{SelectAlgorithm(v_s, v_d, u, l)}$ 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. Secondly, the algorithm which can achieve the lowest cost is preferred. Thus, when two algorithms can achieve the 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).

$\boldsymbol{ExecuteAlgorithm(m, v_s, v_d, \alpha)}$ 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.

Algorithm 1 $Select(v_s, v_d, u, l)$

```
if $Lat_r(v_s, v_d, u, l) < Lat_c(v_s, v_d, l)$ then
    $\alpha \leftarrow \alpha_r$
else if $Lat_r(v_s, v_d, u, l) > Lat_c(v_s, v_d, l)$ 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
```

4 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.

4.1 Simulation Setup

To evaluate our algorithm, we used the Cambridge Haggle dataset [21]. This trace includes Bluetooth sightings of the small devices (iMotes) by groups of users from University of Cambridge Computer Laboratory. To evaluate the message dissemination between the mobile users, we remove the data comes from fixed iMotes. The refined trace consists of the contacts for 36 iMotes which were carried by users over 11 days.

In this experiment, each simulation is repeated 20 times with different random seeds for statistical confidence. Since 1 hour is selected as the size of time slot and the contact history is mapped into a week plan, there are 168 time slots in each simulation. At the beginning of each time slot, 5% nodes are randomly chosen as the source nodes, and each source node sends messages to all other nodes. Consequently, there are 11760 messages created for each simulation. Each message contains the identifiers of the source and the destination nodes, the start time and a given TTL.

4.2 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 (including 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.

RANK: A node forwards a message to the destination node or intermediate nodes whose centrality are higher than its [15]. 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 1 hour.

Habit: The source of a message uses its regularity table to calculate the path which can achieve the best delivery probability [16]. The size of the time slot and the time length to calculate the regularity are 1 hour and 1 week respectively. The threshold δ is 0.2. The minimum regularity in a path is employed to denote the delivery probability of the path.

4.3 Simulation Result

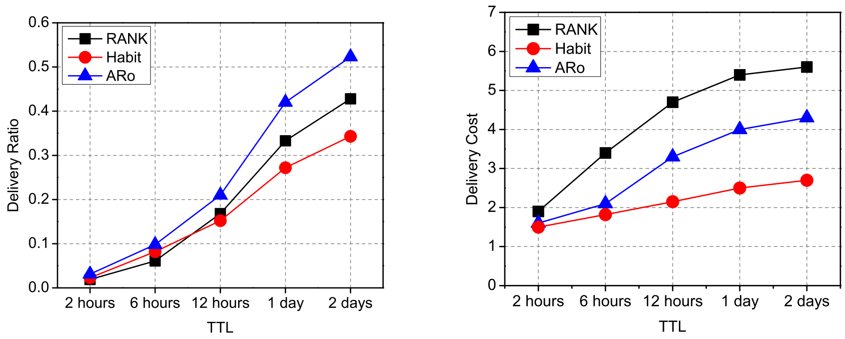

Fig. 2. Delivery ratio (left) and Delivery cost (right) comparison of several algorithms on Cambridge data set

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.2). When TTL is shorter than six hours, 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 twelve hours, 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 twelve hours, our algorithm can achieve about 5% delivery increment than *Habit* does. When TTL is two days, our algorithm can achieve about 10% delivery increment than *RANK* does.

The delivery cost of *ARo* is higher than *Habit* but much lower than *RANK*. When TTL is twelve hours, *ARo* can achieve about 1.5 delivery cost decrement than *RANK* does. When TTL is two days, *ARo* can achieve about 1.6 delivery cost decrement than *RANK* does. Since more messages can be delivered by *RANK* than by *Habit* when TTL is longer than twelve hours, *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 twelve hours.

These results show that *ARo* can exploit the advantages of centrality-based algorithms to overcome the drawbacks of regularity-based algorithms and vice versa to improve the overall routing performance. The results thus validate our hypothesis that an adaptive routing algorithm that can switch between centrality-based and regularity-based algorithms can provide better routing performance.

5 Conclusion and Future Work

In this paper, we presented the first dynamic routing algorithm for mobile peer-to-peer data dissemination. Our algorithm appropriately 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.

Our future work spans two directions. First, 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. Second, we will focus on exploiting communities and the connectors between communities to improve the performance of data dissemination in mobile peer-to-peer networks.

Acknowledgments

This work is partially supported by the China Scholarship Council (CSC) UT-INSA PhD program and the MDPS German-French Doctoral College. Jingwei Miao acknowledges the CSC UT-INSA and the MDPS.

References

[1] Luo, Y., Wolfson, O.: Mobile p2p databases. Encyclopedia of GIS (2008) 671–677

[2] Buford, J., Li, L., Tutschku, K.: Call for papers - 6th ieee international workshop on mobile peer-to-peer computing (mp2p'09) (2009)

[3] Fall, K.: A delay-tolerant network architecture for challenged internets. In: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications. SIGCOMM '03, New York, NY, USA, ACM (2003) 27–34

[4] Vahdat, A., Becker, D.: Epidemic routing for partially connected ad hoc networks. Technical report, Citeseer (2000)

[5] Chaintreau, A., Hui, P., Crowcroft, J., Diot, C., Gass, R., Scott, J.: Impact of human mobility on the design of opportunistic forwarding algorithms. In: Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications, IEEE (April 2006) 1–13

[6] Spyropoulos, T., Psounis, K., Raghavendra, C.: Single-copy routing in intermittently connected mobile networks. In: Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on, IEEE (2004) 235–244

[7] Spyropoulos, T., Psounis, K., Raghavendra, C.: Efficient routing in intermittently connected mobile networks: the multiple-copy case. IEEE/ACM Transactions on Networking (TON) **16**(1) (2008) 77–90

[8] González, M., Hidalgo, C., Barabási, A.: Understanding individual human mobility patterns. Nature **453**(7196) (2008) 779–782

[9] Musolesi, M., Mascolo, C.: A community based mobility model for ad hoc network research. In: REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality, New York, NY, USA, ACM (2006) 31–38

[10] Boldrini, C., Passarella, A.: Hcmm: Modelling spatial and temporal properties of human mobility driven by users' social relationships. Computer Communications **33**(9) (2010) 1056 – 1074

[11] Boldrini, C., Conti, M., Passarella, A.: Users mobility models for opportunistic networks: the role of physical locations. Proc. of IEEE WRECOM (2007)

[12] Nixon, P., Dobson, S., Stabeler, M.: Exploiting human networks for delay tolerant environmental sensing. (2009)

[13] Freeman, L.: A set of measures of centrality based on betweenness. Sociometry **40**(1) (1977) 35–41

[14] Hui, P., Crowcroft, J.: Bubble rap: Forwarding in small world dtns in ever decreasing circles. Univ. of Cambridge, Computer Laboratory, Tech. Rep. UCAM-CL-TR-684 (2007) 1–44

[15] Hui, P., Crowcroft, J.: Predictability of human mobility and its impact on forwarding. In: Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on. (aug. 2008) 543 –547

[16] Mashhadi, A., Ben Mokhtar, S., Capra, L.: Habit: Leveraging human mobility and social network for efficient content dissemination in delay tolerant networks. In: World of Wireless, Mobile and Multimedia Networks Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a. (15-19 2009) 1 –6

[17] Vu, L.H.: CHARACTERIZING AND LEVERAGING PEOPLE MOVEMENT FOR CONTENT DISTRIBUTION IN MOBILE PEER-TO-PEER NETWORKS. PhD thesis, University of Illinois at Urbana-Champaign (2010)

[18] Vu, L., Do, Q., Nahrstedt, K.: 3r: Fine-grained encounter-based routing in delay tolerant networks. (2011)

[19] Jain, S., Fall, K., Patra, R.: Routing in a delay tolerant network. SIGCOMM Comput. Commun. Rev. **34** (August 2004) 145–158

[20] Hossmann, T., Spyropoulos, T., Legendre, F.: Know thy neighbor: towards optimal mapping of contacts to social graphs for DTN routing. In: INFOCOM, 2010 Proceedings IEEE, IEEE (2010) 1–9

[21] Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., Chaintreau, A.: CRAWDAD trace cambridge/haggle/imote/content (v. 2006-09-15). Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/content (September 2006)