

# A Self-Regulating Protocol for Efficient Routing in Mobile Delay Tolerant Networks

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

**Abstract**—In this paper, we present a novel routing protocol for mobile delay tolerant networks, called Community-based Adaptive Spray (CAS). How to maximize routing performance (delivery ratio) and minimize resource consumption (number of message copies) is a common goal in these networks. Our protocol considers the following two aspects toward this goal: 1) selecting the intermediate nodes that are closest to the destination, based on their mobility patterns; 2) dynamically controlling the number of message copies according to the time-to-live of a message. Experiment results demonstrate that our protocol can improve routing performance and resource consumption compared to the state-of-the-art Spray-and-Wait and BUBBLE protocols.

**Index Terms**—delay tolerant networks; digital ecosystem; mobility; routing; community

## I. INTRODUCTION

Mobile Delay Tolerant Networks (Mobile DTNs) are composed of a finite set of mobile devices, such as cell phones and personal digital assistants, which can communicate with each other via short range wireless protocols (e.g., Bluetooth). These mobile devices construct a digital ecosystem for the carriers to interact with each other in their living environment [1]. The characteristics of DTNs include frequent and long-term partitions which make message delivery one of the major challenges in these networks. Hence, message dissemination is often performed in a “store-carry-and-forward” manner [2], where a message is stored by intermediary nodes and forwarded to nodes closer and closer to the destination node until the latter is eventually reached or the message expires.

In the literature, a number of routing protocols have been proposed in DTNs. Nevertheless, most of them only achieve either good performance in terms of delivery ratio or delivery cost. For instance, flooding-based routing protocols achieve the good performance of delivery ratio at the cost of huge resource consumption. Whereas forwarding-based routing protocols consume the smallest resources, however, they sacrifice the delivery ratio. Hence, how to balance the routing performance of delivery ratio and the resource consumption in terms of the number of message copies is a common goal in DTNs [3].

In order to reasonably allocate resources, considerable research works [4][5] utilize the mobility pattern of nodes, which has been theoretically and experimentally proved to influence the routing performance of routing protocols [6][4]. On the other hand, the message dissemination is performed within the Time-To-Live (TTL) of a message. Consequently, an ideal resource allocation should involve two aspects: 1) selecting the intermediate nodes that close to the destinations, according to

the mobility pattern of nodes; 2) dynamically controlling the number of message copies, based on the TTL of a message.

In this paper, we present a novel routing protocol in mobile DTNs, called Community-based Adaptive Spray (CAS) routing protocol. The goal of our protocol is to achieve a good trade-off between the routing performance and the resource consumption. To this end, our protocol considers both of the above two aspects of resource allocation. The protocol is based on communities, since the mobile device carriers generally do not act alone in real life, but tend to belong to communities [4]. Furthermore, the utilization of community is proved to improve the routing performance [4]. Moreover, compared to the number of nodes in a network, the number of communities is so small that the maintenance of information about communities is inexpensive. Hence, it is reasonable to organize the nodes in the network in terms of community [5].

The main idea of our proposal is depicted as follows. The contact information (i.e., inter-contact time, contact times) between nodes and the information about encountered nodes (i.e., the identifier of encountered nodes, the community identifier of the encountered nodes) are exchanged and aggregated, when nodes encounter. The inter-contact time between two nodes is the time interval between two successive contacts. Hence, after a sufficient period of time, each node can establish a graph which reflects the topology of communities in the network.

In our proposal, the routing process is divided into multiple sub-processes. In each sub-process, a message is routed to an intermediate target node, which could be a gateway node between communities or the destination node of the message. At the beginning of each sub-process, our proposal utilizes the local knowledge of nodes to dynamically select the intermediate target node and calculate the minimum number of message copies, which are all based on the mobility pattern of nodes and the TTL of the message. The sub-process is repeated until the message either is delivered or expires. Therefore, the trade-off between the routing performance and the resource consumption can be achieved by minimizing the number of message copies utilized in each sub-process.

The contributions of this paper are twofold:

- We propose an approach to select the gateway node.
- We dynamically calculate the needed number of message copies, according to the TTL of a message.

The remainder of the paper is organized as follows. Section II discusses related works on routing in DTNs. In Section III, we introduce the system model and the information

maintained by each node. In Section IV, we describe our proposal in detail. The simulation and results are presented in Section V. Finally, we conclude the paper in Section VI.

## II. RELATED WORK

In the literature, a variety of routing protocols have been proposed in DTNs. The existing routing protocols can be classified from different aspects [7]. From the aspect of the number of message copies utilized in the routing process, the existing protocols can be classified into two categories: forwarding-based and replication-based. In forwarding-based routing protocols, such as Direct Delivery [8] and First Contact [9], only one copy for each message exists in the routing process. Therefore, these routing protocols achieve the minimum transmission overhead in terms of the number of message copies. However, due to the inherent characteristic of long-term network partitions in DTNs, these protocols often suffer from low delivery ratio and long delivery latency, even if a message is eventually delivered to the destination. Hence, the forwarding-based protocols are generally considered to be inefficient when the TTL of a message is limited [10].

In order to improve the routing performance of delivery ratio and delivery latency, replication-based protocols distribute multiple copies of a message to the network. Based on whether the number of message copies is limited or not, the replication-based routing protocols can be further divided into flooding-based and quota-based [11]. Epidemic [12] is the delegation of flooding-based protocols. It maximizes the routing performance of delivery ratio and delivery latency at the cost of huge resource consumption. However, the resources (e.g., battery and bandwidth) in mobile devices are traditionally limited and scarce. Therefore, flooding-based routing protocols are generally considered to be not applicable in reality [10].

The quota-based routing protocols [11][13] restrict the number of message copies for each message. These routing protocols are proved to achieve a good trade-off between the routing performance and the resource consumption [11][13]. Moreover, these protocols potentially assume that all nodes have the same capability of visiting every location in the network. However, recent studies [4][14] on the characteristics of human mobility from real world traces demonstrate that people usually roam in some relative small regions rather than the whole network space. Hence, the copies of a message are mainly sprayed in the same region as the source node by this kind of routing protocols. In other words, the existing quota-based protocols cannot efficiently deliver the message whose source node and destination node are in different regions.

In order to accommodate the characteristics of human mobility, social-based routing protocols, which can belong to the class of either forwarding-based or replication-based, are proposed in recent years. The utilization of social information (i.e., community, centrality) is proved to improve the routing performance [4]. In the existing social-based protocols, the decision of message forwarding relies on node's centrality, which indicates the relative importance of a node in the network. Nevertheless, the centrality of node cannot accurately

steer a message to its destination node. To address this issue, Dang and Wu [5] utilize gateway nodes to deliver the messages whose source node and destination node belong to different communities. The gateway nodes are the nodes which have the highest probability of encountering the nodes in other communities. However, in [5], messages are routed under Direct Delivery protocol, which is generally considered to be inefficient when the TTL of a message is limited.

## III. SYSTEM MODEL

In this section, we first present the network model. We then introduce the information maintained by each node. Finally, we explain how the information is maintained by each node.

### A. Delay Tolerant Network

**Node.** We define a node as a mobile device. We assume that each node is equipped with a radio interface (e.g., Bluetooth) for short-range communication, and that the transmission range of all nodes is the same. Two nodes are considered to encounter, if they are in the transmission range of each other. Two encountering nodes can exchange messages with each other. We also assume that each node has a unique identifier.

**Community.** A community is defined as a set of nodes which frequently co-exist in a common location. The encounter frequency of nodes in the same community is considered to be much higher than that of nodes in different communities. We also assume that each community has a unique identifier. Let the set of nodes in a community  $C$  be  $V_C = \{u_i | 1 \leq i \leq n\}$ , where  $n = |V_C|$ .

**Inter-contact time between two nodes.** An inter-contact time between two nodes is the time interval between any two successive contacts.

**Inter-contact time between a node and a community.** An inter-contact time between a node  $u$  and a community  $C$  is the time interval between any two successive contacts between node  $u$  and any node in community  $C$ .

**Gateway.** A node  $w$  in a community  $C_i$  is defined as the gateway to another community  $C_j$ , if node  $w$ 's average inter-contact time with community  $C_j$  is the shortest with respect to the other nodes in its community  $C_i$ .

**Network.** Let the set of all nodes in the environment be given as the set  $V$ . Let the set of all communities in the environment be  $M = \{C_i | 1 \leq i \leq k \wedge V = \bigcup_{i=1}^k V_{C_i}\}$ . Edges  $e_{i,j}$  and  $e_{j,i}$  exist between two communities  $C_i$  and  $C_j$  (where  $1 \leq i \leq k$ ,  $1 \leq j \leq k$ , and  $i \neq j$ ), if the nodes in them have encountered at least once. The weight of edge  $e_{i,j}$  denoted as  $\varpi_{i,j}^w$  is the average inter-contact time between the gateway  $w$  in community  $C_i$  with the community  $C_j$ . Let  $E$  be the set of all edges in the network. Therefore, a DTN can be represented as the directed weighted graph  $G(M, E)$ .

**Message.** A message can be considered as a tuple  $\langle S, D, I, C, L, T \rangle$ , where  $S$  is the source node,  $D$  is the destination node,  $I$  is the intermediate target node, which is a gateway node connecting the current community to another community (or the destination node).  $C$  is the identifier of the community connected by the gateway node (or null).  $L$  is the

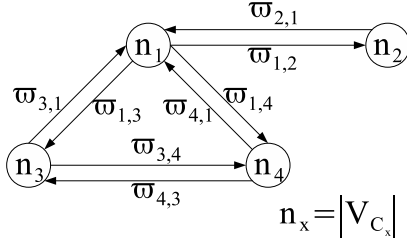


Fig. 1: Community Graph

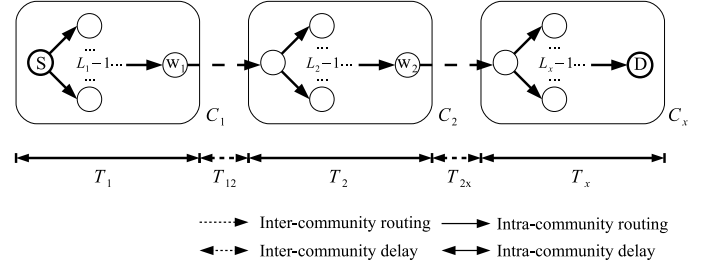


Fig. 2: Routing Protocol Overview

number of message copies permitted in the current community,  $T$  is the TTL.

### B. Information Maintained by a Node

Each node, e.g., node  $u$ , in the network maintains six types of information: its node ID  $u$ , its community ID  $C^u$ , a community table, a gateway table, a contact table, and a community graph.

**Community table.** The community table of node  $u$  holds the node ID and the community ID of all nodes who are encountered by the nodes of  $u$ 's community. The node also maintains a timestamp which indicates the time when the table was last updated.

**Gateway table.** The gateway table of node  $u$  contains the following fields for each known community  $C_i$ : community ID  $C_i$ , node ID of a gateway node (e.g.  $w$ ) in community  $C_i$ , the ID (e.g.  $C_j$ ) of the community connected by the gateway node  $w$ , and the average inter-contact time between gateway  $w$  and community  $C_j$  given as  $\varpi_{i,j}^w$ . Node  $u$  also maintains a timestamp which records the last updated time of this table.

**Contact table.** The contact table of node  $u$  maintains the following fields for each encountered node  $v$ : the node ID  $v$ , the total contact duration  $d_{u,v}$ , the number of encounters  $\sigma_{u,v}$ , the start time of the ongoing encounter  $\tau_{u,v}$  (if there is one).

**Community graph.** In the community graph of node  $u$ , each vertex denotes a known community e.g., community  $C_i$ . Each vertex is labeled by the number of its member nodes. The weight of the directed edge  $e_{ij}$  denoted by  $\varpi_{i,j}^w$  is the average inter-contact time between the gateway  $w$  in community  $C_i$  and the community  $C_j$ . The community graph of node  $u$  locally reflects the topology of its known communities in the network, which is illustrated in Fig. 1.

### C. Maintenance of Information

The maintenance of the above information is driven by events. There are three kinds of events in the protocol: (1) *connect event*, (2) *disconnect event*, and (3) *update event*. We assume that nodes in the network honestly synchronize the maintained information.

**Connect Event.** It happens at the moment when two nodes enter the transmission range of each other. When connect event happens, two encountered nodes honestly exchange and update their corresponding values in each field of community table, community graph, and gateway table, according to the value of timestamp. They also set the timestamp to the current time.

**Disconnect Event.** It happens at the moment when two nodes  $u$  and  $v$  go out of the transmission range of each other. They calculate the contact duration as  $\Delta d_{u,v} = \tau^c - \tau_{u,v}$ , where  $\tau^c$  is the current time. The cumulative contact duration between nodes  $u$  and  $v$  is  $d_{u,v} = d'_{u,v} + \Delta d_{u,v}$ , where  $d'_{u,v}$  is the cumulative contact duration of them before this contact. After that, the  $\sigma_{u,v}$  increases by 1.

**Update Event.** Update event is periodically invoked at the side of all nodes after every  $\gamma$  time units. When update event happens, a node (e.g., node  $u$  in community  $C_i$ ) utilizes (1) calculates the average inter-contact time  $\varpi_{i,j}^u$  between itself and each community  $C_j$  in its community table. For community  $C_j$ , if node  $u$  achieves a shorter average inter-contact time with community  $C_j$  than the current gateway  $w$  does, that is  $\varpi_{i,j}^u \leq \varpi_{i,j}^w - \epsilon$ , where  $\epsilon$  is a threshold. The fields of the node ID of gateway, inter-contact time and timestamp in the entry are then updated to  $u$ ,  $\varpi_{i,j}^u$  and  $\tau^c$ , respectively.

$$\varpi_{i,j}^u = \frac{\tau^c - \sum_{v \in C_j} d_{u,v}}{\sum_{v \in C_j} \sigma_{u,v}} \quad (1)$$

## IV. PROTOCOL DESIGN

In this section, we first briefly describe the mechanism of CAS. We then present the design of CAS in detail.

### A. Overview of CAS

In this section, we give an overview of our routing protocol. A routing example is depicted in Fig. 2. The figure shows when a node (e.g., node  $u$ ) generates a message (or receives a message from another node in a different community), node  $u$  first finds the shortest path from its community to the community of the destination, according to the community graph. After that, node  $u$  extracts the community (e.g. community  $C$ ) which is next to the community of node  $u$  on the path. It then checks up the gateway table to find the gateway (e.g., node  $w$ ) connecting to community  $C$ . Following that, node  $u$  utilizes CAS to calculate the optimal number of message copies needed to route the message to gateway node  $w$ . It afterwards sets gateway node  $w$  (or the destination node of the message, if the destination node is in the same community as node  $u$ ) as the intermediate target node, and routes the message with the specified number of message copies to the intermediate target node. When the intermediate target node receives the message, if it is the destination node, the

routing process terminates; otherwise, it waits to forward the message to the nodes in community  $C$  (or the community of the destination). The previous process is repeated until the message is delivered or expires.

### B. Design of CAS

In our proposal, the routing process is divided into multiple sub-processes. Each sub-process is consisted of the following two components: 1) optimization of message copies and 2) message routing.

1) *Optimization of message copies*: In Spray and Wait routing protocol [13], if  $L$  copies of a message are distributed in the network and the TTL of the message is  $T$ , the expected delivery probability of the message can then be calculated as  $p_d = 1 - e^{-\alpha LT}$ , where  $\alpha$  is the inverse of the average inter-contact time between nodes [15]. Hence, in order to meet the expected delivery probability  $p_d$ , the relationship between the number of message copies  $L$  and the TTL  $T$  can be expressed as (2). Moreover, in order to spend the minimum time to spray  $L - 1$  copies of a message [10], our proposal utilizes Binary Spray and Wait to route a message inside a community.

$$L \times T = -\frac{\ln(1 - p_d)}{\alpha} \quad (2)$$

From the previous section (Section III part A) we can see that a DTN can be denoted as  $G(M, E)$ . We utilize a vector of communities to denote a path  $h$  in  $G(M, E)$ . Consider node  $u$  carries a message denoted by  $\langle S, D, I, C, L, T \rangle$ . Let  $h^*$  be the shortest path from the community of node  $u$  to the community of the destination node. Let  $L_i$  denote the number of message copies distributed in a community  $C_i$ . Let  $w$  be the gateway node connecting community  $C_i$  to community  $C_{i+1}$ . Let  $p_d$  be an expected delivery probability of the message. With notations given above, the optimization of delivery cost can be formulated as (3)-(5).

$$\min \sum_{i=1}^n L_i, \text{ where } n = |h^*| \quad (3)$$

$$s.t. \sum_{i=1}^n \frac{\ln(1 - p_d)}{\alpha L_i} + \sum_{i=1}^{n-1} \varpi_{i,i+1}^w \leq T \quad (4)$$

$$L_i \leq |V_{C_i}|, \text{ where } 1 \leq i \leq |h^*| \quad (5)$$

The objective function (3) is to minimize the number of message copies utilized to route a message. Constraint (4) represents that the sum of the time spent to route the message in communities and the total average inter-contact time between communities should be less than the TTL of the message. Constraint (5) expresses that the number of message copies permitted in a community should be less than the number of nodes in the community.

Since all  $L_i$  are positive integers, we can enumerate all possible values for  $L_i$  to solve (3). The pseudo code of the solution is illustrated in Fig. 3.

---

### Algorithm: OptimizeCost

**Participants:** Node  $u$ .

**Input:** (1)  $m$ , a message carried by node  $u$ . (2)  $D_m$ , the destination node of message  $m$ . (3)  $T$ , the TTL of message  $m$ . (4)  $p_d$ , the expected delivery probability of message  $m$ .

**Output:** A tuple of  $C^*$  and  $L^*$ .

**Notations:** (1)  $L_{min}$ , the minimum number of message copies. (2)  $h^*$ , the shortest path from the community of node  $u$  to the community of node  $D_m$ . (3)  $C^*$ , the ID of the community next to node  $u$ 's community on the path  $h^*$ . (4)  $L^*$ , the number of message copies permitted to route the message to the intermediate target node.

```

1:  $L_{min} \leftarrow \infty$ 
2:  $C^* \leftarrow h^*[2]$ 
3:  $L^* \leftarrow \infty$ 
4:  $\varpi_{sum} \leftarrow \sum_{k=k_1}^{|h^*|-1} \varpi_{k,k+1}$ 
5: for  $i \leftarrow 1$  to  $|h^*|$  do
6:   for  $j \leftarrow 1$  to  $|h[i]|$  do
7:      $L' \leftarrow j$ 
8:      $T' \leftarrow -\frac{\ln(1-p_d)}{\alpha L'}$ 
9:      $R[i][j] \leftarrow \langle L', T' \rangle$ 
10:  end for
11: end for
12:  $getAllPossibleCombination(\nu[1, \dots, 1], Q, h^*)$ 
13: for all  $q \in Q$  do
14:    $T_{sum} \leftarrow 0$ 
15:    $L_{sum} \leftarrow 0$ 
16:   for  $i \leftarrow 1$  to  $|q|$  do
17:      $j \leftarrow q[i]$ 
18:      $\langle T', L' \rangle \leftarrow R[i][j]$ 
19:      $T_{sum} \leftarrow T_{sum} + T'$ 
20:      $L_{sum} \leftarrow L_{sum} + L'$ 
21:   end for
22:   if  $T_{sum} \leq T - \varpi_{sum}$  and  $L_{sum} < L_{min}$  then
23:      $L_{min} \leftarrow L_{sum}$ 
24:      $k \leftarrow q[1]$ 
25:      $\langle T', L' \rangle \leftarrow R[1][k]$ 
26:      $L^* \leftarrow L'$ 
27:   end if
28: end for
29: if  $L^* > |h^*[1]|$  then
30:    $L^* \leftarrow |h^*[1]|$ 
31: end if
32: return  $\langle C^*, L^* \rangle$ 

```

$getAllPossibleCombination(\nu[ ], Q, h^*)$

```

1:  $Q \leftarrow Q \cup \{\nu[ ]\}$ 
2: for  $1 \leq i \leq |\nu[ ]|$  do
3:   if  $\nu[i] < |h^*[i]|$  then
4:      $\nu[i] \leftarrow \nu[i] + 1$ 
5:     return  $getAllPossibleCombination(\nu[ ], Q, Z)$ 
6:   end if
7: end for

```

Fig. 3: Algorithm:OptimizeCost

---

2) *Message Routing*: Routing process is carried out as follows. Consider a message  $m$  with  $L > 1$  copies carried by a node (e.g., node  $u$ ). When node  $u$  encounters another node (e.g., node  $v$ ), it checks whether node  $v$  is inside its community. If nodes  $u$  and  $v$  are in the same community, it checks whether node  $v$  is the intermediate target node. If so, node  $u$  forwards message  $m$  to node  $v$ ; otherwise, it hands over  $\lfloor L/2 \rfloor$  copies of message  $m$  to node  $v$  and keeps

---

**Protocol: RouteMessage****Participants:** Two encountering nodes  $u$  and  $v$ .**Input:** (1)  $m$ , a message carried by node  $u$ . (2)  $D_m$ , the destination node of message  $m$ . (3)  $I_m$ , the intermediate target node of message  $m$ . (4)  $L_u$ , the number of copies of message  $m$  carried by node  $u$ . (5)  $L_v$ . (6)  $T$ , the TTL of message  $m$ . (7)  $p_d$ , the expected delivery probability of message  $m$ .**Output:** Message  $m$  is forwarded to node  $v$ , if nodes  $u$  and  $v$  are in the same community or node  $v$  is the destination node or in the community of the destination node.**Notations:** (1)  $C^*$ , the ID of the community next to the community of node  $v$  on the shortest path from the community of node  $v$  to the community of node  $D_m$ . (2)  $L^*$ , the number of message copies permitted to route the message to the intermediate target node in the community of node  $v$ .**Events and Associated Actions:****node  $u$  initiates the protocol**

```

1: if  $C^u = C^v$  then
2:   if  $v = I_m$  then
3:     node  $u$  forwards message  $m$  to node  $v$ .
4:   else if  $L_u > 1$  then
5:      $L_v \leftarrow \lfloor L_u/2 \rfloor$ 
6:      $L_u \leftarrow \lceil L_u/2 \rceil$ 
7:     node  $u$  forwards  $L_v$  copies of message  $m$  to node  $v$ .
8:   end if
9: else
10:  if  $v = D_m$  or  $C^{D_m} = C^v$  then
11:    node  $u$  forwards message  $m$  to node  $v$ 
12:  end if
13: end if

```

**upon node  $v$  receives a message  $m$  from node  $u$** 

```

1: if  $C^u \neq C^v$  then
2:    $\langle C^*, L^* \rangle \leftarrow \text{OptimizeCost}(D_m, T, p_d)$ 
3:   if  $C^{D_m} = C^v$  then
4:      $I_m \leftarrow D_m$ 
5:   else
6:     node  $v$  looks up its gateway table to find the gateway node
        $w$  to community  $C^*$ 
7:      $I_m \leftarrow w$ 
8:   end if
9:    $L_v \leftarrow L^*$ 
10: end if

```

Fig. 4: Protocol:RouteMessage

the remaining copies of message  $m$ . If nodes  $u$  and  $v$  are in different communities, it checks whether node  $v$  is the destination node or in the same community as the destination node. If so, node  $u$  forwards message  $m$  to node  $v$ .

When node  $v$  receives a copy of message  $m$  from node  $u$ , it checks whether node  $u$  is in its community. If so, node  $v$  keeps the copy of message  $m$ . Otherwise, it utilizes Algorithm *OptimizeCost* to reset the field of intermediate target node  $I$  and the number of message copies  $L$  in the replica of message  $m$ . The protocol of *RouteMessage* is illustrated in Fig. 4.

## V. PERFORMANCE EVALUATION

In this section, we first present the simulation setup and the performance metrics. After that, we briefly describe the compared routing protocols. At last, we compare the performance of our proposal with the compared routing protocols.

TABLE I: Parameter Settings

Parameter Name	Value
Simulation area	$3000 \times 1500 m^2$
Transmission range	10 m
Simulation duration	1 hour + 12 hours + TTL
Warm-up period	1 hour
Message generation rate	a message per 30 seconds
Number of communities	12
Number of nodes in a community	20
Node speed	1.34 m/s
$p_l$	0.8
$p_r$	0.2
$\gamma$	1 hour
$p_d$	0.8

## A. Simulation Setup

The experiment is conducted by the Opportunistic Network Environment (ONE) simulator [9]. The simulation scenario is a  $3000 \times 1500m^2$  area. This area is equally divided into  $4 \times 3$  regions. Each region is initially deployed twenty nodes. Each node moves under the community-based mobility model presented in [16]. In the adopted mobility model, if a node is in its community, for the next movement, it has a probability of  $p_l$  to stay in its community or  $1 - p_l$  to roam outside its community. If it is outside its community, it has a probability of  $p_r$  to stay outside its community; while  $1 - p_r$  to go back to community. According to [16] and the simulation settings, the inter-contact time between nodes in the same community is 1405 s. In the experiment, each simulation is repeated 5 times with different random seeds for statistical confidence. The detailed settings are listed in TABLE I.

## B. Metrics

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 message transmissions in the simulation. To normalize this, we divide it by the total number of unique messages created.

## C. Routing Protocols

Based on the above settings, we conducted our experiment with the following routing protocols.

**Epidemic:** Each node forwards a copy of each unexpired message in its buffer to the encountered node that does not have a copy of the message.

**Direct:** The source node of a message can only forward the message to the destination node.

**Binary Spray and Wait (SaW):** Each messages has  $L$  copies. A message carrier forwards half of the message copies to an encountered node, if its  $L > 1$  and the latter does not have the message. The message, which has only one message copy left, can only be forwarded to the destination. In the experiment,  $L$  is set to 10.

**Bubble:** It utilizes social information about nodes, such as their centrality and the community to which they belong. In

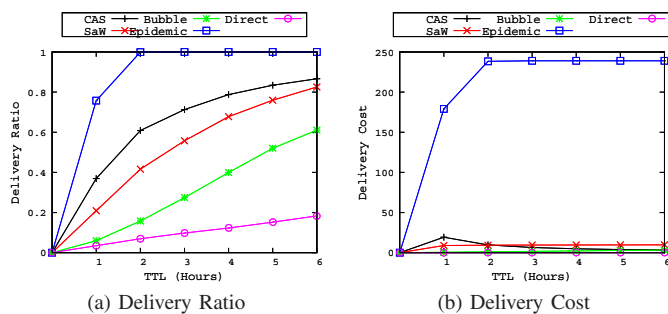


Fig. 5: Routing performance comparison of several algorithms

this protocol, a message is forwarded based on the global rankings of two encountering nodes, until it reaches a node in the community of the destination node. After that, the message is forwarded based on the local rankings of two encountering nodes, until it either reaches the destination node or expires. In the experiment, the length of time window is set to 1 hour.

#### D. Simulation Results

In Fig. 5(a), the performance of delivery ratio of the compared routing protocols is illustrated. Epidemic and Direct always achieve the upper and lower bounds of the performance of delivery ratio, respectively. It can be seen that the delivery ratios of these algorithms increase, as TTL becomes longer. The delivery ratio of our algorithm is always better than those of other algorithms. When TTL is less than 3 hours, the delivery ratio of our proposal is much higher than that of Spray and Wait, since the allocation of message copies is based on the TTL of messages. Moreover, the delivery ratio of our proposal is much higher than that of Bubble, since our proposal can reasonably select the node which connects two given communities, when the source and destination are in different communities.

In Fig. 5(b), the performance of delivery cost of the compared routing protocols is demonstrated. Epidemic and Direct always achieve the lower and upper bounds of the performance of delivery ratio, respectively. As it may be seen, when TTL is less than 2 hours, the delivery cost of our proposal is much higher than that of Binary Spray and Wait, since our proposal replicates more message copies for the messages with a short TTL. When TTL is longer than 2 hours, the delivery cost of our proposal is less than that of Binary Spray and Wait, since our proposal reduces the needed number of message for the messages with a longer TTL.

## VI. CONCLUSION

In this paper, we present a novel routing protocol in mobile DTNs. Our protocol considers both of the mobility pattern of nodes and the TTL of a message to allocate the number of message copies utilized in the routing process. The conducted experiment demonstrates that our protocol can adjust the number of message copies based on the TTL. Moreover, the experiment results show that our protocol can improve the

routing performance compared to quota-based Spray and Wait protocol and social-based BUBBLE protocol.

Our future work spans two directions. First, we would like to conduct more experiments to evaluate the performance of our protocol. Second, we would like to prove that our protocol can express a class of routing protocols by varying the number of message copies. It can prove that our protocol is suitable to a variety of environments

## ACKNOWLEDGMENTS

This work is supported in part 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] I. Carreras, A. Francescon, and E. Gregori, "Nomadic wireless sensor networks for autonomic pervasive environments," in *Autonomic Communication*, ser. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006, vol. 3854, pp. 254–265.
- [2] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 27–34.
- [3] A. Balasubramanian, B. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 373–384, Aug. 2007.
- [4] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 11, pp. 1576–1589, 2011.
- [5] H. Dang and H. Wu, "Clustering and cluster-based routing protocol for delay-tolerant mobile networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 6, pp. 1874–1881, 2010.
- [6] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 606–620, 2007.
- [7] M. Liu, Y. Yang, and Z. Qin, "A survey of routing protocols and simulations in delay-tolerant networks," in *Wireless Algorithms, Systems, and Applications*, ser. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2011, vol. 6843, pp. 243–253.
- [8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 63–76, February 2008.
- [9] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, ser. Simutools '09. ICST, Brussels, Belgium, Belgium: ICST, 2009, pp. 55:1–55:10.
- [10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the multiple-copy case," *IEEE/ACM Trans. Netw.*, vol. 16, pp. 77–90, February 2008.
- [11] S. C. Nelson, M. Bakht, R. Kravets, and A. F. Harris, III, "Encounter-based routing in dtms," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, pp. 56–59, June 2009.
- [12] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Duke University, Tech. Rep., 2000.
- [13] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, ser. WDTN '05. New York, NY, USA: ACM, 2005, pp. 252–259.
- [14] M. Gonzalez, C. Hidalgo, and A. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
- [15] E. Bulut, Z. Wang, and B. K. Szymanski, "Cost-effective multiperiod spraying for routing in delay-tolerant networks," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 1530–1543, Oct. 2010.
- [16] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc '06. New York, NY, USA: ACM, 2006, pp. 49–60.