# Community-Aware Opportunistic Routing in Mobile Social Networks

Mingjun Xiao, *Member*, *IEEE*, Jie Wu, *Fellow*, *IEEE*, and Liusheng Huang, *Member*, *IEEE*

**Abstract—**Mobile social networks (MSNs) are a kind of delay tolerant network that consists of lots of mobile nodes with social characteristics. Recently, many social-aware algorithms have been proposed to address routing problems in MSNs. However, these algorithms tend to forward messages to the nodes with locally optimal social characteristics, and thus cannot achieve the optimal performance. In this paper, we propose a distributed optimal Community-Aware Opportunistic Routing (CAOR) algorithm. Our main contributions are that we propose a home-aware community model, whereby we turn an MSN into a network that only includes community homes. We prove that, in the network of community homes, we can still compute the minimum expected delivery delays of nodes through a reverse Dijkstra algorithm and achieve the optimal opportunistic routing performance. Since the number of communities is far less than the number of nodes in magnitude, the computational cost and maintenance cost of contact information are greatly reduced. We demonstrate how our algorithm significantly outperforms the previous ones through extensive simulations, based on a real MSN trace and a synthetic MSN trace.

**Index Terms—**Community, delay tolerant networks, mobile social networks, opportunistic routing

## 1 Introduction

Mobile social networks (MSNs) are a special kind of delay tolerant network (DTN), in which mobile users move around and communicate with each other via their carried short-distance wireless communication devices. Typical MSNs include pocket switch networks, mobile vehicular networks, mobile sensor networks, etc. [1]. As more users exploit portable short-distance wireless communication devices (such as smart phones, iPads, mobile PCs, and sensors in vehicles) to contact and share data between each other in a cheap way, MSNs attract more attention. Since MSNs experience intermittent connectivity incurred by the mobility of users, routing is a mainly concerning and challenging problem.

Recently, some social-aware routing algorithms that are based on social network analysis have been proposed, such as Bubble Rap [2], SimBet [3], and algorithms in [4]–[7], etc. Two key concepts in social network analysis are: (i) *community*, which is a group of people with social relations; (ii) *centrality*, which indicates the social relations between a node and other nodes in a community. Based on the two concepts, these algorithms detect the communities and compute the centrality value for each node. Messages are delivered via the nodes with good centralities. Since social relations of mobile users generally have long-term characteristics and are less volatile than node mobility, social-aware algorithms outperform traditional DTN algorithms, such as flooding-based algorithms [8], [9] and probability-based algorithms [10]–[14]. Despite this, these algorithms tend to forward messages to the nodes with locally best centralities.

In this paper, we focus on the single-copy routing problem in MSNs. In many real MSNs, mobile users that have a common interest generally will visit some (real or virtual) location that is related to this interest. For instance in Fig. 1, students with a common study interest will visit the same classrooms to take part in the same courses; customers with the same shopping interests often visit the same shops; friends generally share some resources through facebook, and so on. Based on this basic social characteristic, we propose a home-aware community model. Mobile users with a common interest autonomously form a community, in which the frequently visited location is their common "*home*." Moreover, like [1], we assume that each home supports a real or virtual throwbox [15], a local device that can temporarily store and transmit messages.

Under the home-aware community model, we propose a distributed optimal Community-Aware Opportunistic Routing algorithm (CAOR). We first turn the routing between lots of nodes to the routing between a few community homes. Then, we adopt the optimal opportunistic routing scheme by maintaining an optimal relay set for each home. Each home only forwards its message to the node in its optimal relay set, and ignores other relays. Since this scheme solves the problem of whether a home should select a visited node as the relay of message delivery or ignore this visited node to wait for those better relays, it can achieve the optimal performance. More specifically, our major contributions are summarized as follows:

1. We present a home-aware community model and extend the centrality concept from a single node to a group of nodes. Unlike existing community models, each

- M. Xiao and L. Huang are with the School of Computer Science and Technology, Suzhou Institute for Advanced Study, University of Science and Technology of China, Hefei, 230027, China. E-mail: {xiaomj, lshuang}@ustc.edu.cn.
- J. Wu is with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122. E-mail: jiewu@temple.edu.

*Manuscript received 02 Oct. 2012; revised 13 Feb. 2013; accepted 25 Feb. 2013. Date of publication 19 Mar. 2013; date of current version 27 June 2014. Recommended for acceptance by K. Li.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TC.2013.55*

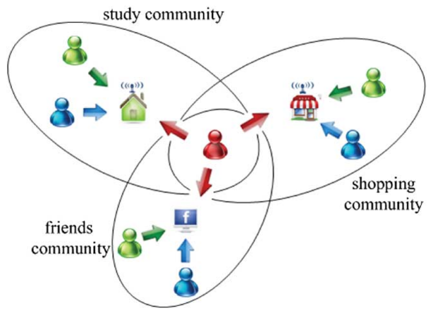



Fig. 1. An example of mobile social network.

community home in our model is assumed to have a throwbox to store and transmit messages.

2. We present a rule of optimal opportunistic routing through a theoretical analysis. We design a reverse Dijkstra algorithm to determine the optimal relays and compute the minimum expected delivery delay. Based on this, the CAOR algorithm can achieve the optimal opportunistic routing performance.
3. We turn the routing in $|V|$ mobile nodes into a routing in $|L|$ ($|L| \ll |V|$) community homes by virtue of the home-aware community model. Moreover, we prove that the simplification will not sacrifice the routing performance. As a result, the network scale and the maintaining costs are reduced significantly.
4. We first design the CAOR algorithm for the case that each home has a real throwbox. Then, we extend it to the case of virtual throwbox by letting the members of a community with high centralities act as the home of this community. Simulation results show that it can achieve a nearly optimal performance.

The rest of the paper is organized as follows. We introduce the network model in Section 2. Section 3 is the social network model. The overview, detailed implementation, and extension of CAOR are presented in Sections 4, 5, and 6, respectively. In Section 7, we evaluate the performance of our algorithms through extensive simulations. After reviewing related work in Section 8, we conclude the paper in Section 9. All proofs are presented in the Appendix.

## 2 Network Model & Assumptions

We consider an MSN composed of $|V|$ nodes $V = \{v|v \in V\}$ moving among $|L|$ locations $L = \{l|l \in L\}$ ($|L| \ll |V|$). Each mobile node visits a few locations frequently, while visiting the others rarely. A typical MSN is the Wi-Fi campus network at Dartmouth College [16]. In this network, over 6,000 different users (i.e., students and faculty equipped with devices such as PDAs, laptops, and phones) move among about 500 recorded access points (APs) installed in over 190 buildings, including the college's academic buildings, the library, and the student residences, etc. Moreover, previous works observed that 50% of mobile users in this network spent 74.0% of their time at a single AP to show the characteristic of frequently visiting a few locations [17]. Another MSN that follows this characteristic is the mobile vehicular network, in which lots of buses and taxies move among bus stations and taxi stops.

TABLE 1
Description of Auxiliary Variables

| | |
|---|---|
| $v, l, d, S$ | $v$ is a node, $l$ is a location or home, $d$ is a destination, and $S$ is a set of nodes or homes. |
| $\lambda_{v,l}, \lambda_{S,l}$ | the exponential distribution parameter of that node $v$ and (any node in) node set $S$ visit home $l$. |
| $\tilde{R}_i$ | the optimal relay set for the opportunistic routing from a message sender ($i$) to destination $d$. |
| $\tilde{S}_{i,j}$ | the optimal betweenness set for the message delivery from home $i$ to $j$. When the context is clear, the subscripts are removed. |
| $D_{i,d}(S)$ | the minimum expected delivery delay from $i$ to $d$ via relay set $S$, where $i$ may be a node, or a home. Specifically, $D_{i,d} = D_{i,d}(\tilde{R}_i)$ when $S = \tilde{R}_i$. |
| $B_{l,l'}(S)$ | the betweenness of $S$, i.e., the expected delivery delay from home $l$ to $l'$ only via node set $S$. |

In this paper, we call the frequently visited locations *homes*. Like in the previous work, we assume that the behavior of each mobile node visiting homes follows the Poisson process. In other words, the time interval that each node visits a home follows an exponential distribution (the unit of time is time slot). This is consistent with most real traces. Many previous works, such as [5], [14], also make a similar assumption. Moreover, if a node stays at a home for a long time during each visit, it will be approximately treated as that the node visits the home multiple times, and stays for a unit time during each visit. Besides, message transmissions happen only when nodes visit homes. Here, we assume that the occasional meeting outside of the homes cannot provide the message transmission, due to limited meeting time.

We also assume that each home has a "throwbox" [15], which has the ability to store and transmit messages. In fact, many MSNs follow this assumption. For instance, the Road Side Units (RSUs) in mobile vehicular networks are a kind of real throwboxes. The APs in the Wi-Fi campus network can also be seen as a type of real throwboxes since each user can upload/download data from network storages via these APs. For other MSNs without real throwboxes, we let the nodes that frequently visit or reside in a home act as the virtual throwboxes of this home. In the following, we first consider the case of real throwbox. And then, we extend our solution to the case of virtual throwbox in Section 6. In addition, we assume that there are no stable wired links between these throwboxes, especially for those virtual throwboxes. In fact, if there is a wired link between two throwboxes and the cost of message delivery between them is negligible, we can combine them as one throwbox.

Based on the above descriptions, we assume that each node $v \in V$ only visits a few homes most of the time, and the interval of $v$'s visit home $l \in L$ follows the exponential distribution with the parameter $\lambda_{v,l}$. Then, our *objective* is to design a single-copy routing algorithm with the minimum expected delivery delay. For ease of presentation, we list the main auxiliary variables in Table 1.

## 3 Social Network Modeling

Before presenting our algorithm, we build the community and define two social metrics in this section.

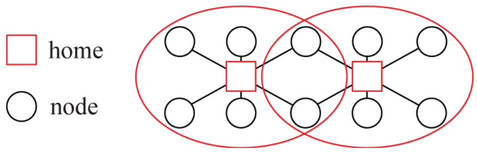



Fig. 2. Example: home-aware communities.

### 3.1 Building Home-Aware Communities

In this paper, we propose a concept of home-aware community. A home-aware community is a community of nodes that frequently visit a given home. The frequently visited home is the common home of the community members, i.e., the community home. Moreover, if a node visits several homes frequently, it can belong to multiple communities and have multiple homes. Concretely, we define the community as follows:

**Definition 1.** Home-Aware Community: *a home-aware community $C_l$ is a set of nodes that frequently visit home $l$ (beyond a given threshold). That is:*

$$C_l = \{v | \lambda_{v,l} \geq \varepsilon, v \in V\}. \tag{1}$$

*Moreover, home $l$ is equipped with a real or virtual throwbox, so that it can be used as a relay of message delivery.*

Based on the definition, each community has a star topology where its home is the center. The whole network is composed of some overlapped star-topology communities, as shown in Fig. 2. A community can easily be detected. Each node $v$ only needs to estimate the parameter $\lambda_{v,l}$ for each home $l$, based on the history records. Then, it can find the communities that it belongs to according to Definition 1. Moreover, if a node belongs to a community, it means that the node frequently visits the community home due to some interest. Each community exactly contains a group of nodes that have the common interest to the community home.

### 3.2 Centrality Metric

In an MSN, the centrality metric is generally used to measure the importance of nodes during message delivery. A node with a better centrality value means that it has a stronger capability of connecting with other nodes. Previous works mainly adopt three centrality measures: degree centrality, closeness centrality, and betweenness centrality [3]. Degree centrality is measured as the number of direct links between a given node and other nodes. Closeness centrality is a measure of how long it will take to deliver a message from a given node to other nodes. Betweenness centrality measures the extent to which a node lies on the paths linking other nodes. In this paper, we model the whole network into some overlapped star-topology communities. The message delivery thus can be turned into the delivery within and/or between these communities. Accordingly, we only present an intra-community centrality metric and an inter-community betweenness metric for nodes, to measure their importance in the message delivery within and between these communities, respectively.

#### 3.2.1 Intra-Community Centrality

In intra-community routing, the most concern is measuring the capability of each community member to meet and deliver messages to other members. Since intra-community message deliveries happen only when nodes visit community homes, the smaller the expected delay to visit a community home, the higher capability to deliver messages a community member would have. In fact, the expected delay for a node $v$ to visit a community home $l$, denoted by $D_{v,l}$, can be simply derived from the parameter $\lambda_{v,l}$. That is, $D_{v,l} = 1/\lambda_{v,l}$. Therefore, we can directly use this value to measure the intra-community centrality of the node to the community.

**Definition 2. Intra-Community Centrality.** *$I_l(v)$ is the reciprocal of the expected delay for node $v$ visiting a community home $l$, i.e., $I_l(v) = 1/D_{v,l} = \lambda_{v,l}$.*

According to Definition 2, the node with the largest intra-community centrality in a community has the best capability to deliver messages. In our extension (Section 6), we will use such nodes to act as virtual throwboxes in the communities.

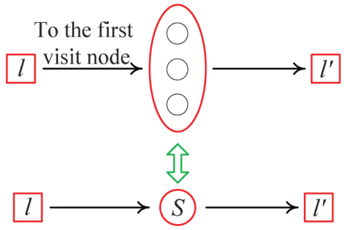



Fig. 3. Opportunistic routing between communities (opportunistic routing via a set of nodes can be seen as a message delivery via a virtual node).

#### 3.2.2 Inter-Community Betweenness

In this paper, we adopt the opportunistic routing scheme, in which multiple nodes cooperatively deliver messages. It can be defined as follows:

**Definition 3. Opportunistic Routing.** *Each message sender (home or node) has a relay set (homes or nodes). Once a relay in the set meets the message sender, the sender will let this relay deliver messages. In other words, the first relay in the set to meet the message sender will act as the real relay.*

This dynamical routing scheme is more general than the routing based on a single fixed relay. Based on this scheme, we extend the concept of betweenness from a single node to a node set. Concretely, we define the inter-community betweenness to measure the ability of a node set to be taken as a communication bridge between communities. Moreover, we use the delivery delay to evaluate the inter-community betweenness of a set of nodes.

**Definition 4. Inter-Community Betweenness.** *$B_{l,l'}(S)$ is the expected delivery delay that it takes for a relay set $S$ ($S \subseteq C_l \cap C_{l'}$) to cooperatively deliver messages from community home $l$ to $l'$. The smaller the $B_{l,l'}(S)$, the better the delivery ability of $S$ will be.*

According to the opportunistic routing scheme, a node in the relay set acts as the real relay only when it is the first node in the set to meet the message sender, as shown in Fig. 3. Thus, we can compute the inter-community betweenness of a given node set by the following theorem.

**Theorem 1.** *For two overlapped communities $C_l$, $C_{l'}$ and an arbitrary relay set $S$ ($S \subseteq C_l \cap C_{l'}$), we have:*

$$B_{l,l'}(S) = \frac{1}{\sum_{v \in S} \lambda_{v,l}} + \frac{\sum_{v \in S} \lambda_{v,l}/\lambda_{v,l'}}{\sum_{v \in S} \lambda_{v,l}}. \tag{2}$$

**Proof.** See Appendix A. □

In Eq. (2), the first part $B^{(1)}_{l,l'} = \frac{1}{\sum_{v\in S}\lambda_{v,l}}$ actually indicates the expected delay that the first node in $S$ visits $l$, and the second part $B^{(2)}_{l,l'} = \frac{\sum_{v\in S}\lambda_{v,l}/\lambda_{v,l'}}{\sum_{v\in S}\lambda_{v,l}}$ is the average delay of the nodes when $S$ forwards a message to $l'$. For simplicity, we can see the whole set as one entity, and treat it just as a virtual node $S$, whose $\lambda$ parameter is $\lambda_{S,l} = \sum_{v\in S}\lambda_{v,l}$ and the average delivery delay to $l'$ is $B^{(2)}_{l,l'}$, as shown in Fig. 3. Note that the betweenness values of the virtual node is the same as that of the original relay set, i.e., $B_{l,l'}(S) = \frac{1}{\lambda_{S,l}} + B^{(2)}_{l,l'}$.

For an arbitrary pair of overlapped communities $C_l$ and $C_{l'}$, there are $2^{|C_l\cap C_{l'}|}$ different relay sets. There must be a relay set that has the smallest betweenness. We define this set as the optimal betweenness set.

**Definition 5. Optimal Betweenness Set.** *$\tilde{S}_{l,l'}$ is the relay set with the smallest betweenness for the message delivery from community home $l$ to $l'$. Concretely,*

$$\tilde{S}_{l,l'} = \underset{S\subseteq C_l\cap C_{l'}}{argmin}\, B_{l,l'}(S). \tag{3}$$

In Section 5, we present an algorithm (Algorithm 1) to determine the optimal betweenness set for each pair of overlapped communities. These optimal betweenness sets will be used to deliver messages by CAOR.

## 4 Overview of CAOR

In this section, we introduce the methodology and basic idea of CAOR. Here, we assume that the source (and relays) knows which communities that the destination $d$ belongs to. That is, the message consists of the source, the destination information, and the data to be delivered. This assumption is reasonable because the source generally knows some basic information about the destination in most message delivery tasks. In fact, the source has many ways to know the basic information of the destination. For example, the destination can broadcast this information to all community homes, or a naming service is used to distribute this information, etc. A similar assumption is also adopted in previous work [5].

### 4.1 Methodology: Optimal Opportunistic Routing

The optimal opportunistic routing scheme means that each message sender delivers messages via its optimal relay set (i.e., delivers messages via the first encountered relay in this set). The key problem is to determine whether a relay belongs to the optimal relay set for each message sender.

To this end, we derive an optimal opportunistic routing rule. Without loss of generality, we consider an opportunistic routing from a message sender $i$ to the destination $d$ via some candidate relays $\{u|\lambda_{i,u} > 0\}$. Here, the message sender $i$ might be a mobile node or a home. Each $u$ is a one-hop relay of $i$, i.e., $\lambda_{i,u} > 0$, but it does not must be a one-hop relay of the destination. The optimal relay set, denoted by $\tilde{R}_i$, is given by the following formula:

$$\tilde{R}_i = \underset{S\subseteq\{u|\lambda_{i,u}>0\}}{argmin}\, D_{i,d}(S). \tag{4}$$

In Eq. (4), $D_{i,d}(S)$ is the expected delay for $i$ delivering messages to $d$ via the relay set $S$. Moreover, for simplicity, we let

$$D_{i,d} = D_{i,d}(\tilde{R}_i). \tag{5}$$

Then, the optimal opportunistic routing rule is presented as follows.

**Theorem 2.** *Optimal Opportunistic Routing Rule: the message sender always delivers messages to the encountered relay that has a smaller minimum expected delay to the destination than itself. Concretely, a relay $u$ belongs to the optimal relay set $\tilde{R}_i$ for the delivery from $i$ to $d$, if and only if, $D_{u,d} < D_{i,d}$, i.e.:*

$$u \in \tilde{R}_i \Leftrightarrow D_{u,d} < D_{i,d}. \tag{6}$$

**Proof.** See Appendix B. □

According to Theorem 2, we only need to compute and compare the minimum expected delivery delays from the message sender and the relay to the destination. Then, we can determine whether the relay belongs to the optimal relay set of the sender.

### 4.2 The Basic Idea

The CAOR algorithm consists of two phases: the initialization phase and the routing phase. The initialization phase simplifies the network with $|V|$ nodes to the network with $|L|$ community homes through the social network modeling. Then, under the simplified network, the routing phase delivers messages based on the optimal opportunistic routing rule. Based on the social network modeling and the optimal opportunistic routing rule, CAOR can achieve the optimal performance with a small cost. More specifically, the basic idea of CAOR is presented as follows.

#### *4.2.1 The Initialization Phase*

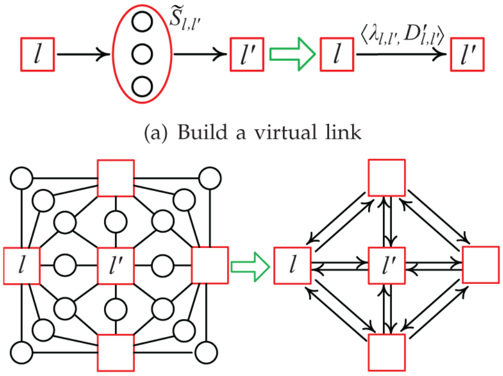
In the initialization phase, each community home $l$ collects the optimal betweenness set of each pair of community homes and uses these information to locally construct a contact graph of these homes.

First, each community home $l$ determines the optimal betweenness sets for the message deliveries from itself to other community homes. Given the parameters $\lambda_{v,l}$ and $\lambda_{v,l'}$ for each node $v$ in community $C_l$ and another overlapped community $C_{l'}$, i.e., $\{\lambda_{v,l}, \lambda_{v,l'}|v \in C_l \cap C_{l'}\}$, the optimal betweenness set $\tilde{S}_{l,l'}$ can be determined through a greedy algorithm (Algorithm 1 in Section 5). Only the nodes in optimal betweenness sets will be used to deliver messages.

Second, home $l$ treats the whole optimal betweenness set $\tilde{S}_{l,l'}$ as a virtual node, and builds a virtual link $\overrightarrow{ll'}$ for the message delivery from $l$ to $l'$ via $\tilde{S}_{l,l'}$. Moreover, this link is attached with a weight $\langle\lambda_{l,l'}, D'_{l,l'}\rangle$, as shown in Fig. 4(a). The time interval that the virtual link $\overrightarrow{ll'}$ emerges follows the exponential distribution with the parameter $\lambda_{l,l'}$. The expected delivery delay via the virtual link is $D'_{l,l'}$ (i.e., the delay for the optimal betweenness set to forward the messages to home $l'$). Moreover, the weight satisfy:

$$\lambda_{l,l'} = \lambda_{l,\tilde{S}_{l,l'}} = \sum_{v\in\tilde{S}_{l,l'}}\lambda_{l,v}, \tag{7}$$

$$D'_{l,l'} = B^{(2)}_{l,l'} = \frac{\sum_{v\in\tilde{S}_{l,l'}}\lambda_{l,v}D_{v,l'}}{\sum_{v\in\tilde{S}_{l,l'}}\lambda_{l,v}}. \tag{8}$$




(a) Build a virtual link

(b) Simplify the whole network

Fig. 4. An example of network simplification.

Finally, home $l$ sends the link weight $\langle \lambda_{l,l'}, D'_{l,l'} \rangle$ to, and receives link weights from, other homes. Based on these link weights, home $l$ constructs a direct weighted graph $G = \langle L, W \rangle$, where $W = \{\langle \lambda_{l,l'}, D'_{l,l'} \rangle | l, l' \in L\}$, as shown in Fig. 4(b).

In addition, when the link weight $\langle \lambda_{l,l'}, D'_{l,l'} \rangle$ changes by more than a threshold along with the time eclipse, home $l$ also will send the updated weight to other homes.

#### 4.2.2 The Routing Phase

The routing phase computes the minimum expected delivery delays and makes the routing decision based on the optimal opportunistic routing rule when a node $v$ visits a community home $l_i$.

First, node $v$ gets the direct weighted graph $G$ from home $l_i$, and extends this graph to $G^+ = \langle L^+, W^+ \rangle$ by adding the destination and itself. For simplicity, we treat $v$ and $d$ as two virtual homes, i.e., $l_n = v$ ($n = |L| + 1$) and $l_0 = d$. Moreover, we let $L^+ = L + \{l_n, l_0\}$, and $W^+ = W + \{\langle \lambda_{l_n,l}, 0 \rangle, \langle \lambda_{l,l_n}, 0 \rangle, \langle \lambda_{l,L_0}, 0 \rangle | l \in L\}$.

Second, in the extended graph $G^+$, home $l_n$ (i.e., node $v$) computes the minimum expected delivery delays $D_{l_n,l_0}$ and $D_{l_i,l_0}$ through a reverse Dijkstra algorithm (Algorithm 3 in Section 5). The basic idea is to compute the minimum expected delivery delays from homes or nodes in $L^+$ to the destination $l_0$ in an ascending order. Without loss of generality, we assume that $L^+ = \{l_0, l_1, \ldots, l_n\}$, and the minimum expected delivery delays for each home in set $S = \{l_0, l_1, \ldots, l_k\}$ (i.e., $D_{l_0,l_0}, \ldots, D_{l_k,l_0}$) have been calculated, as shown in Fig. 5. Then, we compute the expected delay from each home $l \in L^+ - S = \{l_{k+1}, \ldots, l_n\}$ to home $l_0$ via $S$, according to the following formula[1].

$$D_{l,l_0} = \sum_{l' \in S} \int_0^{\infty} \lambda_{l,l'} e^{-\sum_{l' \in S} \lambda_{l,l'} t} (t + D'_{l,l'} + D_{l',l_0}) dt$$
$$= \frac{1 + \sum_{l' \in S} \lambda_{l,l'} (D'_{l,l'} + D_{l',l_0})}{\sum_{l' \in S} \lambda_{l,l'}}. \quad (9)$$

Assume that $D_{l,l_0}$ is the smallest when $l = l_{k+1}$. Then, this delay is exactly the minimum expected delivery delay of $l_{k+1}$. In the same way, $D_{l_{k+2},l_0}, \ldots, D_{l_n,l_0}$ can be iteratively derived.

1. $\lambda_{l,l'} e^{-\sum_{l' \in S} \lambda_{l,l'} t}$ is the probability density for the message delivery via $l'$; $D'_{l,l'} + D_{l',l_0}$ is the expected delay for the optimal betweenness set of $l$ and $l'$ to forward the message to $l'$ plus the expected delay from $l'$ to $l_0$.

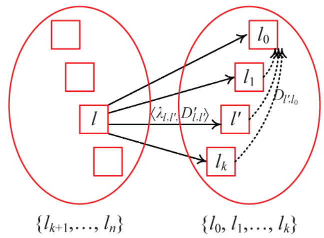



Fig. 5. Iteratively compute the minimum expected delivery delay.

Finally, home $l_n$ (i.e., node $v$) compares $D_{l_n,l_0}$ and $D_{l_i,l_0}$ and lets the one with a smaller delay value receive the message.

### 4.3 Optimality of CAOR

First, we show that CAOR can achieve the minimum expected delivery delay in the simplified network. As the description in Section 4.2, CAOR uses a reverse Dijkstra algorithm to calculate the minimum expected delivery delay from each home to the destination in the extended graph $G^+$. The minimum expected delivery delays will be derived in an ascending order. When we compute the $(k+1)$-th minimum expected delivery delay, the delays smaller than this one have been derived out. According to the optimal opportunistic routing rule in Theorem 2, only these homes can be candidate relays of the home with the $(k+1)$-th minimum expected delivery delay. Thus, Eq. (9) can correctly derive the minimum expected delivery delays of all homes to the destination.

Second, we can get that the minimum expected delivery delays that are derived in the simplified network are equal to the corresponding values in the original network. In fact, this is ensured by the following theorem.

**Theorem 3.** *Assume that community $C_l$ has $m$ overlapped communities $C_{l_1}, \ldots, C_{l_m}$. Then, the optimal relay set $\tilde{R}_l$ of home $l$, and the optimal betweenness sets $\tilde{S}_{l,l_i}$ $(1 \le i \le m)$ satisfy:*

1) if $v \notin \bigcup_{i=1}^{m} \tilde{S}_{l,l_i}$, then $v \notin \tilde{R}_l$;
2) $\tilde{S}_{l,l_i} \subseteq \tilde{R}_l$, otherwise $\tilde{S}_{l,l_i} \cap \tilde{R}_l = \emptyset$ for $\forall i \in [1, m]$.

**Proof.** See Appendix C. □

Theorem 3 shows that either all nodes in the optimal betweenness set $\tilde{S}_{l,l_i}$, or none of them, belong to the optimal relay set $\tilde{R}_l$; furthermore, if a node does not belong to any optimal betweenness set, it also does not belong to the optimal relay set. This means that the simplified network still contains all the nodes that belong to the optimal relay sets, and that the each optimal betweenness set can act as a relay as a whole. Thus, the minimum expected delivery delays that are derived in the simplified network are equal to the results in the original network.

Based on the above analysis and the optimal opportunistic routing rule in Theorem 2, we can straightforwardly get the optimality of CAOR. That is,

**Corollary 1.** *CAOR can achieve the minimum expected delivery delay.*

### 4.4 Discussion: Overhead of CAOR

CAOR makes the routing decision on a simplified network. The simplification process has two advantages.

On the one hand, the network with $O(|V|^2)$ edges is turned into a network with $O(|L|^2)$ edges. The computation, storage and communication costs are significantly reduced, since $|L| \ll |V|$ (e.g., the $|V|^2/|L|^2$ of the real MSN trace used by our evaluation in Section 7 is about 10,000).

On the other hand, the edge weights of a simplified network only depend on the optimal betweenness sets. The behavior of the nodes outside optimal betweenness sets would not result in the update of edge weights. In general, each optimal betweenness set only includes several nodes that frequently visit the community homes. The occasional visit behavior would not result in an update of the edge weights. In fact, these frequent visit behaviors indicate the potential social relationships between communities, which show long-term characteristics. For example, the dependency between an educational course community and a research community in a campus social network is a long-term social characteristic; students in a research community would periodically take part in corresponding fundamental courses; this even would not be violated by the enrollment of new students and the graduation of old students. The relationship between departments in a company or institution is also a long-term social characteristic, which would not be violated by employment variation, and so on. Thus, the updating and management costs of the shared network information are also significantly reduced.

These two advantages make CAOR able to achieve the optimal performance with a small cost, which is very important to real MSNs.

## 5 Detailed Implementation of CAOR

This section introduces the detailed implementation of CAOR, including the initialization phase and the routing phase. The first phase simplifies the network (Algorithm 2), and the second phase computes the minimum expected delivery delay (Algorithm 3) and makes the routing decision (Algorithm 4). Both of the phases are implemented in the distributed way.

### 5.1 Initialization Phase

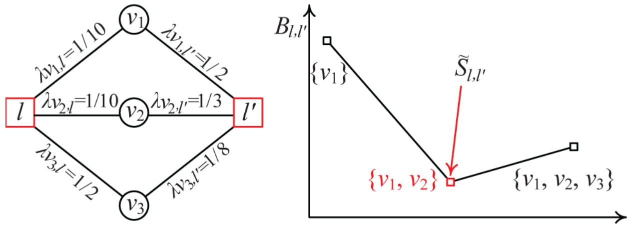
The key of the initialization phase is to determine the optimal betweenness sets for each pair of communities. To this end, we introduce a theorem, by which these optimal betweenness sets can be efficiently derived out.

In previous work [14], Conan et al. exploit the fixed point technique to design an efficient algorithm, which can be used to determine the optimal betweenness set $\tilde{S}$. In fact, the results, including a property and the corresponding algorithm, also can be derived from our optimal opportunistic routing rule in Theorem 2. For the integrity of this paper, we introduce them here, as shown in Corollary 2 and Algorithm 1.

**Corollary 2.** *Assume that* $\lambda_{v_1,l'} \geq \lambda_{v_2,l'} \geq \cdots \geq \lambda_{v_n,l'}$, *then the optimal betweenness set* $\tilde{S}_{l,l'}$ *satisfies:*

1. $v_1 \in \tilde{S}_{l,l'}$;
2. *if* $v_{i+1} \in \tilde{S}_{l,l'}$, *then* $v_i \in \tilde{S}_{l,l'}$. *That is,* $\exists k \in [1,n]$ *s.t.* $\tilde{S}_{l,l'} = \{v_1, \ldots, v_k\}$;
3. *if* $\tilde{S}_{l,l'} = \{v_1, \ldots, v_k\}$, *then* $B_{l,l'}(\{v_1, \ldots, v_i\}) > B_{l,l'}(\{v_1, \ldots, v_i, v_{i+1}\})$ *for any* $i \in [1, k-1]$.

**Proof.** See Appendix D. □




(a) A simple MSN (b) Betweenness vs. set

Fig. 6. Example: determine optimal betweenness set.

The algorithm to determine $\tilde{S}_{l,l'}$, based on Corollary 2, is greedy. The basic idea [14] is to add nodes $v_1, v_2, \ldots, v_n$ into $\tilde{S}_{l,l'}$ in turn, to extend the relay set until the corresponding delivery delay $B_{l,l'}(\tilde{S}_{l,l'})$ increases; then, end the extending operation and get $\tilde{S}_{l,l'}$. Concretely, $\tilde{S}_{l,l'}$ is initialized in step 1, and is extended in step 3. Step 4 computes the minimum expected delay according to Eq. (2). The correctness of this algorithm is provided by Corollary 2. The first property of the theorem ensures the correctness of the initialization of $\tilde{S}_{l,l'}$, the second property ensures the correctness of the extended $\tilde{S}_{l,l'}$, and the third property ensures the correctness of ending the extending operation. The computational overhead of this algorithm is $O(|C_l \cap C_{l'}|)$.

---

**Algorithm 1** Determine optimal betweenness set

---

**Require** $\{\langle\lambda_{v_1,l}, \lambda_{v_1,l'}\rangle, \ldots, \langle\lambda_{v_n,l}, \lambda_{v_n,l'}\rangle\}$ $(\lambda_{v_1,l'} \geq \cdots \geq \lambda_{v_n,l'})$

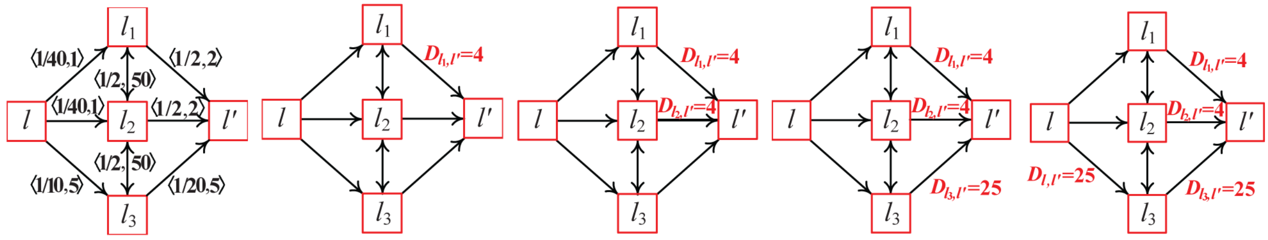
**Ensure** $\tilde{S}_{l,l'}$, $\langle\lambda_{l,l'}, D'_{l,l'}\rangle$

1: Initialize: $S = \{v_1\}$ and $D_{l,l'}(S) = \frac{1}{\lambda_{v_1,l}} + \frac{1}{\lambda_{v_1,l'}}$;
2: **for** $i = 2, \ldots, n$ **do**
3: $\quad S = S + \{v_i\}$;
4: $\quad$Incrementally compute $D_{l,l'}(S)$ by Eq. (2);
5: $\quad$**if** $D_{l,l'}(S)$ increases **then**
6: $\quad\quad$Break;
7: **return** $\tilde{S}_{l,l'} = S - \{v_i\}$ and corresponding $\langle\lambda_{l,l'}, D'_{l,l'}\rangle$;

---

Fig. 6 shows an example of greedily determining the optimal betweenness set. There are three relays $v_1, v_2, v_3$ for message delivery from $l$ to $l'$, as shown in Fig. 6(a). The optimal betweenness set $\tilde{S}$ is determined by adding nodes $v_1, v_2, v_3$ into $\tilde{S}_{l,l'}$ in turn, and finding the minimum $D_{l,l'}(\tilde{S}_{l,l'})$. When $\tilde{S}_{l,l'} = \{v_1\}$, the expected delay is 12. When node $v_2$ is added into $\tilde{S}_{l,l'}$, the expected delay decreases to be 15/2. However, after node $v_3$ is added into $\tilde{S}_{l,l'}$, the expected delay increases to become 55/7, as shown in Fig. 6(b). Then, the optimal relay set $\tilde{S}_{l,l'} = \{v_1, v_2\}$. Note that the optimal relay for single path routing might not be the optimal relay for opportunistic routing. In this example, though node $v_3$ is the best relay for the single path delivery from $l$ to $l'$, it does not belong to $\tilde{S}_{l,l'}$.

Now, we present the implementation of the initialization phase by Algorithm 2. Each community home first collects the $\lambda$ parameters of its community members in Step 1. Then, the home exploits Algorithm 1 to determine




(a) simplified netwrok (b) comp. $D_{l_1,l'}$ (c) comp. $D_{l_2,l'}$ (d) comp. $D_{l_3,l'}$ (e) comp. $D_{l,l'}$

Fig. 7. An example of iteratively computing minimum expected delays: in each round, the minimum expected delay of a home is determined according to Eq. (9) ($\tilde{S}_{l_1,l'} = \tilde{S}_{l_2,l'} = \tilde{S}_{l_3,l'} = \{l'\}$, $\tilde{S}_{l,l'} = \{l_1, l_2\}$).

the optimal betweenness sets for the message deliveries from itself to other community homes in Step 2. In Step 3, the home produces the virtual links for these deliveries and sends the corresponding weights to other community homes. Next, the home receives the link weights of other pairwise community homes to locally construct the contact graph of homes in Steps 4 and 5. Note that the algorithm is a distributed one. The correctness is provided by Corollary 2. The computational overhead is dominated by the cost of determining the optimal betweenness sets in Step 2.

**Algorithm 2** CAOR: initialization

**Ensure** $G = \langle L, W\rangle$, where $W = \{\langle\lambda_{l,l'}, D'_{l,l'}\rangle | l, l' \in L\}$

**For** each community home $l \in L$ **do**

1: Collect $\lambda_{v,l}$, $\lambda_{v,l'}$ for each $v \in C_l$ and $l' \in L - \{l\}$;

2: Use Algorithm 1 to produce $\tilde{S}_{l,l'}$ and $\langle\lambda_{l,l'}, D'_{l,l'}\rangle$;

3: Create the virtual link $\overrightarrow{ll}'$ : $\langle\lambda_{l,l'}, D'_{l,l'}\rangle$ for each $l' \in L - \{l\}$ and send the link weights to other homes;

4: Receive the link weights from other homes;

5: Construct the contact graph $G = \langle L, W\rangle$;

### 5.2 Routing Phase

The routing phase extends the graph, uses the reverse Dijkstra algorithm to compute the minimum expected delays for each home in the extended graph, and then makes the routing decision. The reverse Dijkstra algorithm is shown in Algorithm 3. Steps 1 and 2 are the initialization. In each round, i.e., Steps 4–9, the minimum expected delivery delay of a home is determined. An example is shown in Fig. 7. The correctness is provided by Eq. (9) and the optimal opportunistic routing rule. The computational overhead is $O(|L|^2)$.

The routing decision of CAOR is shown in Algorithm 4. When a node $v$ visits a community home $l$, it first construct the extended contact graph of homes $G^+$ in Step 4 by adding $v$ and $d$ into the graph $G$, which is generated by home $l$ in the initialization phase. Then, node $v$ uses Algorithm 3 to compute the minimum expected delivery delays $D_{v,d}$ and $D_{l,d}$ in Step 5. The routing decision is made in Steps 6–9. The computational overhead of this algorithm is dominated by the execution of Algorithm 3 in Step 5. Moreover, the rule of optimal opportunistic routing ensures that this algorithm can achieve the minimum expected delay for each message delivery.

**Algorithm 3** Compute minimum expected delay

**Require** $G^+ = \langle L^+, W^+\rangle$, $i$, $l_0 = d$

**Ensure** $D_{i,d}$

1: Set $S = \emptyset$;

2: Let $D_{l_0,l_0} = 0$, $S \leftarrow l_0$, and $L^+ = L^+ - l_0$;

3: **for** each $l \in L^+$ **do**

4: Compute $D_{l,l_0}(S)$ according to Eq. (9);

5: Select the smallest one, and let $D_{l,l_0} = D_{l,l_0}(S)$;

6: **if** $l$ is $i$ **then**

7: Break;

8: **else**

9: $S \leftarrow l$, and $L^+ = L^+ - l$;

10: **return** $D_{i,d} = D_{l,l_0}$;

## 6 Extension

In this section, we extend the CAOR algorithm to the case that community homes have no ability to store messages. Generally, there are many nodes that reside in these homes of very high probabilities, according to the real traces of MSNs. We thus can use these nodes to act as the message relays, i.e., virtual throwboxes, for this case. For simplicity, we assume that the average residual time of each node for each visit is $\tau$. Here, the residual time $\tau$ also means that if the time interval of two nodes that are visiting the same home is no more than $\tau$, then they can exchange messages. Under this model, we can compute the residual probability of a node $v$ visiting a home $l$, denoted by $r_{v,l}$, by the following formula:

$$\begin{aligned} r_{v,l} &= \frac{1}{t}\sum_{k=1}^{\infty}\frac{e^{-\lambda_{v,l}t}(\lambda_{v,l}t)^k}{k!}k\tau \\ &= \lambda_{v,l}\tau. \end{aligned} \tag{10}$$

In this formula, $\frac{e^{-\lambda_{v,l}t}(\lambda_{v,l}t)^k}{k!}$ is the probability that $v$ visits $l$ for $k$ times. Note that $\lambda_{v,l}$ is actually the visit frequency of node $v$ to the community home $l$ in a unit time, and $\tau$ is the stay time of each visit. Thus, we always have $r_{v,l} = \lambda_{v,l}\tau \leq 1$. Moreover, the direct delivery delay of two community members, $v$ and $v'$, satisfy the following formula:

$$D_{v,v'} = \int_0^{\infty} t\, r_{v',l} \lambda_{v,l} e^{-r_{v',l}\lambda_{v,l} t} dt = \frac{1}{\lambda_{v,l} r_{v',l}} = \frac{1}{\lambda_{v,l}\lambda_{v',l}\tau}. \quad (11)$$

Here, it should be stated that even though the average residual time is assumed to be the same for the sake of simplicity, the formulas are still correct if they are different.

**Algorithm 4** CAOR: routing

```
For each node v ∈ V do
    if v visits a community home l ∈ L then
    for each message of v and l, v do
        Extract destination (d) information;
        Get G+ by adding v and d to G in home l;
        Compute D_{v,d} and D_{l,d} through Algorithm 3;
        if D_{v,d} < D_{l,d} then
            Let v hold the message;
    else
        Let l hold the message;
```

Compared to the case where the community homes have storage abilities, the probability parameter for node $v$ visiting the home $l$ becomes $\lambda_{v,l} r_{v',l}$ from $\lambda_{v,l}$. It is multiplied by a coefficient $r_{v',l}$ that is near to 1. Thus, after replacing the $\lambda_{v,l}$ in the CAOR algorithm by $\lambda_{v,l} r_{v',l}$, the algorithm can still achieve a nearly optimal performance.

## 7 Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performance of the CAOR algorithm using the MSN trace from the Wi-Fi campus network of Dartmouth College [16]. Since the other widely used MSN traces, such as the Infocom trace and the UMassDieselNet trace, have not provided the location information, we also extend the evaluation to a synthetic trace based on a Time-variant Community Model [18]. We compare the CAOR algorithm with the existing social-aware algorithms: SimBet [3] and Bubble rap [2]. Three performance metrics commonly used are examined: delivery ratio, delivery delay and delivery hops. Simulation results demonstrate that the CAOR algorithm can significantly improve MSN routing performance.

### 7.1 Algorithms in Comparison

**SimBet** [3] is an MSN routing algorithm that is based on small world dynamics. In this algorithm, a novel metric, SimBet Utility, composed with betweenness and similarity, is proposed; the ego network analysis technique is used to estimate the values of the betweenness centrality metric and the similarity metric for each node, based on local information. When some nodes encounter each other, the algorithm lets the node with the maximum utility value deliver messages.

**Bubble rap** [2] is an MSN routing algorithm based on $k$-clique community detection. This algorithm uses the weighted network analysis technique to detect the $k$-clique community, and computes centrality ranks for each node. Messages are forwarded (bubble) up along the hierarchical tree using the global rank, and then bubble up by using the local rank when they reach nodes that are in the same community as the destinations.

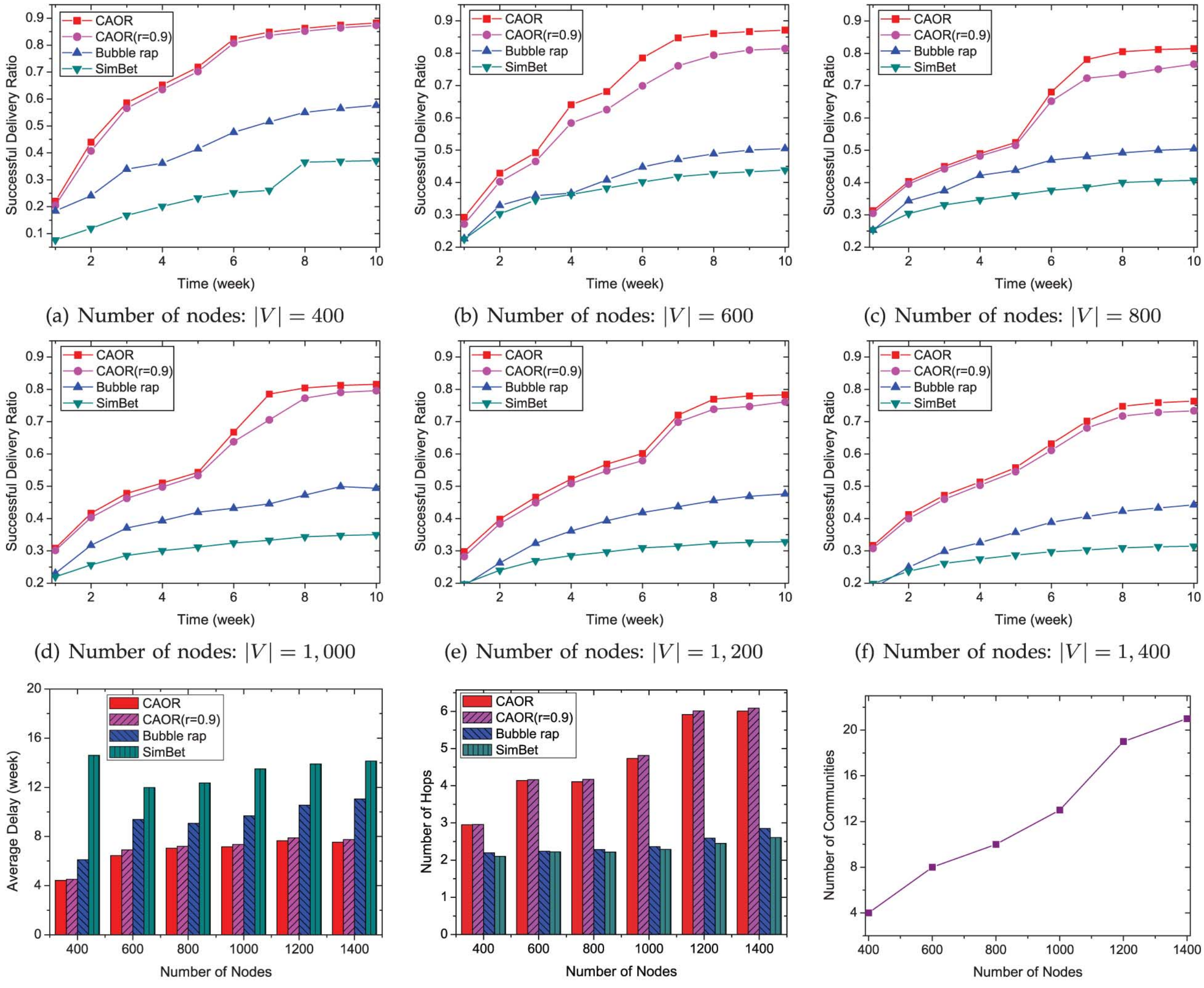
**CAOR** ($r = 0.9$) is the extended version of CAOR, where $r = 0.9$ means that each community has a node with a residual probability of 0.9. Here, we only let $r = 0.9$ for simplicity. In fact, we get similar simulation results for other residual probabilities.

### 7.2 Simulations on the Real Trace

We adopt the real experimental trace of the Wi-Fi campus network in Dartmouth College [16] in our simulations, since it is one of the most extensive and widely exploited data traces. This trace includes 507 valid APs and uses 6,022 log files to list the records for each node's visit to the APs from 2001–2003. Due to the limit of our PC memory and computation ability, we randomly select partial mobile nodes and APs from the trace to construct an MSN, while ensuring adequate connectivity of the network. Concretely, the network is built as follows.

1. We first derive the number of related nodes for each AP from the trace, and randomly select an AP with enough related nodes as the seed AP, denoted by $l_1$. Here, a node is said to be related to an AP if there are records about its visit to this AP in the trace. In addition, we use $C_{l_i}$ to denote the set of related nodes of AP $l_i$.
2. Secondly, we assume that $l_1, \ldots, l_{i-1}$ have been determined, and then we determine the $i$-th AP. We randomly select an AP from the remaining APs, and compute $|C_i \cap C_k|$ for each $k \in [1, i-1]$. If there exists a $k \in [1, i-1]$ to make $|C_i \cap C_k| > 80\%|C_k|$, we consider this AP be very close to $l_k$, and then we drop it and reselect a new AP. If $|C_i \cap C_k| < 10\%|C_k|$ for all $k \in [1, i-1]$, we consider this AP to not have enough connectivity to other APs, and thus we drop it and select a new AP. We repeatedly select APs and test the conditions until a suitable AP is found, and then let it be $l_i$. If an AP that satisfies the conditions cannot be found, then we return and restart from step (1).
3. We repeat step (2) until $|C_1 \cup \cdots \cup C_i| \geq |V|$. If $|C_1 \cup \cdots \cup C_i| > |V|$, we randomly remove some nodes to make $|C_1 \cup \cdots \cup C_i| = |V|$. Then, we get $L = \{l_1, \ldots, l_i\}$, and $V = C_1 \cdots \cup C_i$. Finally, we compute the $\lambda$ parameter for each node's visit to each AP, according to the trace.

After building the MSN, we conduct the simulations by generating 10,000 messages for randomly selected source nodes, and by executing the above-mentioned algorithms to forward these messages to their destinations, while recording the delivery delay, delivery ratio, and delivery hops. Moreover, we give each message a time-to-live (TTL) value. Messages would be dropped if their TTLs are exhausted. For the fairness of comparison, we set the delays of failed message deliveries as the maximum delay value, i.e., TTL, though their delivery hops might be very small.




(a) Number of nodes: $|V| = 400$ (b) Number of nodes: $|V| = 600$ (c) Number of nodes: $|V| = 800$

(d) Number of nodes: $|V| = 1,000$ (e) Number of nodes: $|V| = 1,200$ (f) Number of nodes: $|V| = 1,400$

(g) Comparison of average delay (h) Comparison of average hops (i) Communities vs. nodes

Fig. 8. Performance comparisons of CAOR, CAOR ($r = 0.9$), SimBet, and Bubble rap on the real trace.

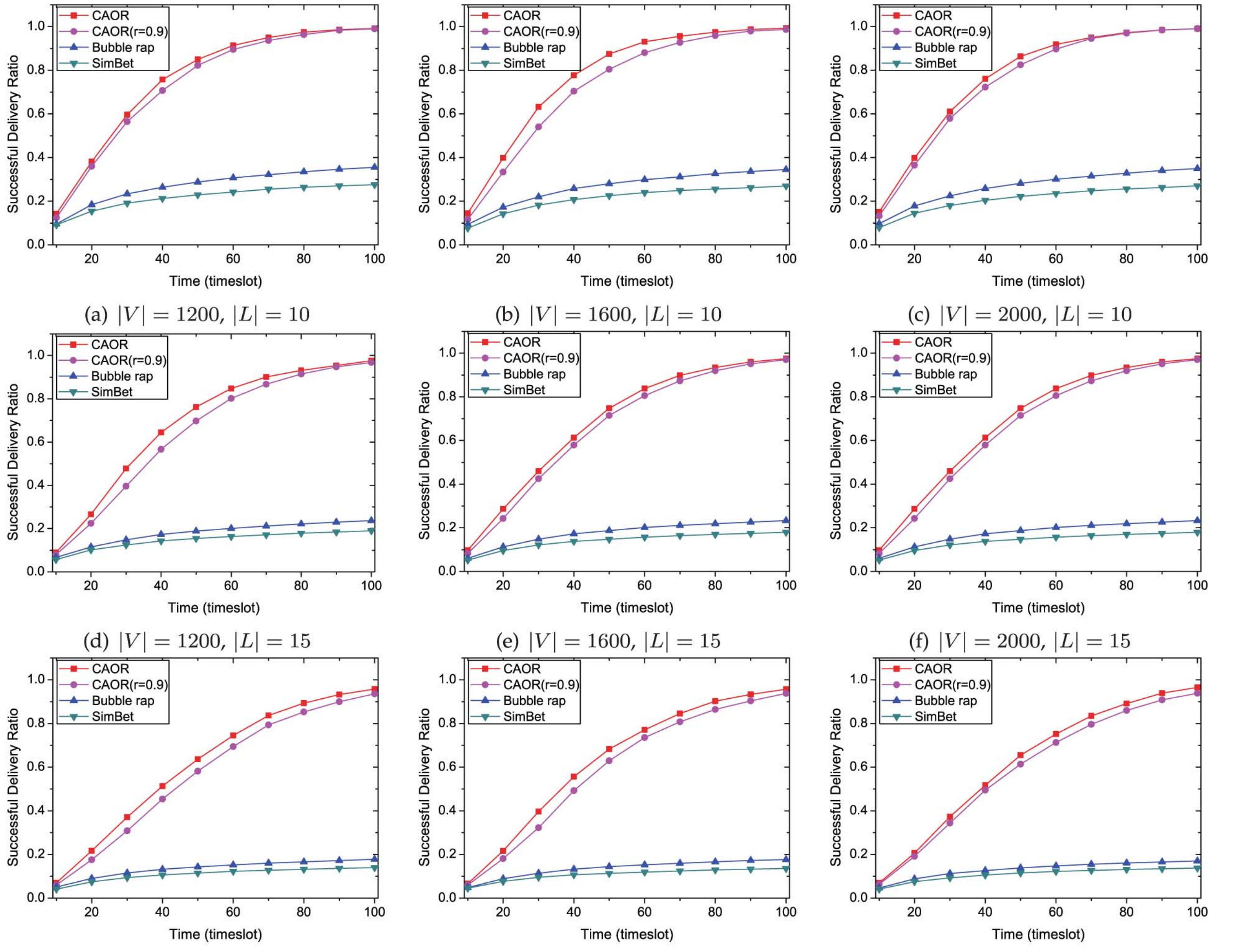
In simulations on evaluating the delivery ratio, we set the number of nodes to $|V| = 400, 600, 800, 10{,}00, 1{,}200,$ and $1{,}400$, respectively. The TTLs are set from 1 week to 10 weeks. Simulation results are shown in Fig. 8(a)–(f). In simulations on evaluating the average delivery delay and average delivery hops, the TTL is 20 weeks. The results are shown in Fig. 8(g)–(h).

These results show that CAOR significantly outperforms SimBet and Bubble rap. Compared with SimBet and Bubble rap, CAOR increases the delivery ratio by about 89.5% and 35.8%, and reduces the delivery delay by about 49.6% and 22.7%, respectively. We also list the contrast between the number of communities and nodes in Fig. 8(i). It shows that the number of communities is far less than the number of nodes. Since the routing decision of CAOR mainly depends on the number of communities, the computational cost and maintenance cost are very low. Moreover, along with the increasing of nodes, the number of communities increases. Due to the increasing network scale, the delivery ratios of these algorithms have a little decreases, and the average delivery delay of CAOR has a little increase. However, the increasing of communities has no effect on the average delivery delays of SimBet and Bubble rap. This is because many message deliveries in both of the algorithms converge at the homes with local high centrality values. The average delivery hops of the two algorithms are about two-hops, which also proves their property of local convergency. In contrast, there is no local convergency in CAOR.

In addition, we also compare the performance of the extended CAOR algorithm, i.e., CAOR ($r = 0.9$) with SimBet and Bubble rap, as show in Fig. 8(a)–(h). The results show that the performances of CAOR ($r = 0.9$) are very close to CAOR.

### 7.3 Simulations on the Synthetic Trace

The synthetic trace in our simulations is produced by a Time-variant Community Model in [18]. In this model, each node is randomly assigned to several community homes on a plane. Time is equally divided into timeslots. In each timeslot, nodes perform random waypoint trips with a probability $1 - p$ of roaming outside and a probability $p$ of staying inside (or getting back to) their homes. By choosing different probabilities $p$ for each node, a large range of heterogeneous node behaviors can be reproduced. More specifically, we set $10 \sim 20$ homes, and let each node be randomly related to $1 \sim 5$ homes. The probability $p$ of each node visiting a home is set to be a random value in [0.1,0.2]. Note that this community model is a discrete version of our network model described in Section 2, in which the parameter $\lambda$ is actually equal to the




(a) $|V| = 1200$, $|L| = 10$ (b) $|V| = 1600$, $|L| = 10$ (c) $|V| = 2000$, $|L| = 10$

(d) $|V| = 1200$, $|L| = 15$ (e) $|V| = 1600$, $|L| = 15$ (f) $|V| = 2000$, $|L| = 15$

(g) $|V| = 1200$, $|L| = 20$ (h) $|V| = 1600$, $|L| = 20$ (i) $|V| = 2000$, $|L| = 20$

Fig. 9. Comparisons of delivery ratios of CAOR, CAOR ($r = 0.9$), SimBet, and Bubble rap on the synthetic trace.

visiting probability $p$. Thus, the produced trace is consistent with our network model.

We execute algorithms CAOR, CAOR ($r = 0.9$), Bubble rap, and SimBet on this synthetic trace, respectively. In the simulations, 10,000 messages are generated by randomly selecting source and destination nodes. The average delivery delay, delivery ratio, and delivery hops are recorded. In simulations on evaluating the delivery ratio, we set $|V| = 1200$, 1600, and 2,000, respectively. The number of community homes is set to be $|L| = 10$, 15, and 20. The TTLs are set from 10 to 100 timeslots. Simulation results are shown in Fig. 9. In simulations on evaluating the average delivery delay and average delivery hops, we let $|V| = 1{,}000$, $1{,}200, \cdots, 2{,}000$ nodes. The number of community homes and TTL are the same as the simulations on the delivery ratio. The results are shown in Fig. 10.

These results also show that, compared with SimBet and Bubble rap, CAOR has a significantly large delivery ratio and a low average delivery delay, and CAOR ($r = 0.9$) has a performance close to CAOR. Moreover, the average delivery delay of CAOR is nearly irrelative to the number of nodes. This is consistent with our theoretical analysis. In fact, our theoretical analysis shows that message deliveries mainly depend on the number of communities and the nodes in optimal relay sets. Other nodes will not participate in the message deliveries. As a result, there is only a very small variation with the number of nodes when we fix the number of communities.

## 8 Related Work

So far, many traditional DTN routing algorithms have been proposed. These algorithms include flooding-based algorithms (e.g., [8], [9]) and probability-based algorithms (e.g., [10], [11], [13], [14], [19], [20]). Among these algorithms, the $\mathrm{MH}^*$ algorithm [14] adopts the optimal opportunistic routing strategy, based on global contact information. Compared with this algorithm, the CAOR algorithm adopts the home-aware community model and turns the routing problem among mobile nodes into the routing problem among static communities, and therefore, achieves the optimal routing performance only based on community contact information. The maintenance cost of the contact information is far less than the $\mathrm{MH}^*$ algorithm. This is important because it means that the mobility behaviors of most nodes would not affect the routing performance of the whole network. Moreover, since the network is simplified to be a static network, many previous routing algorithms in static networks, such as wireless sensor networks, can be applied.

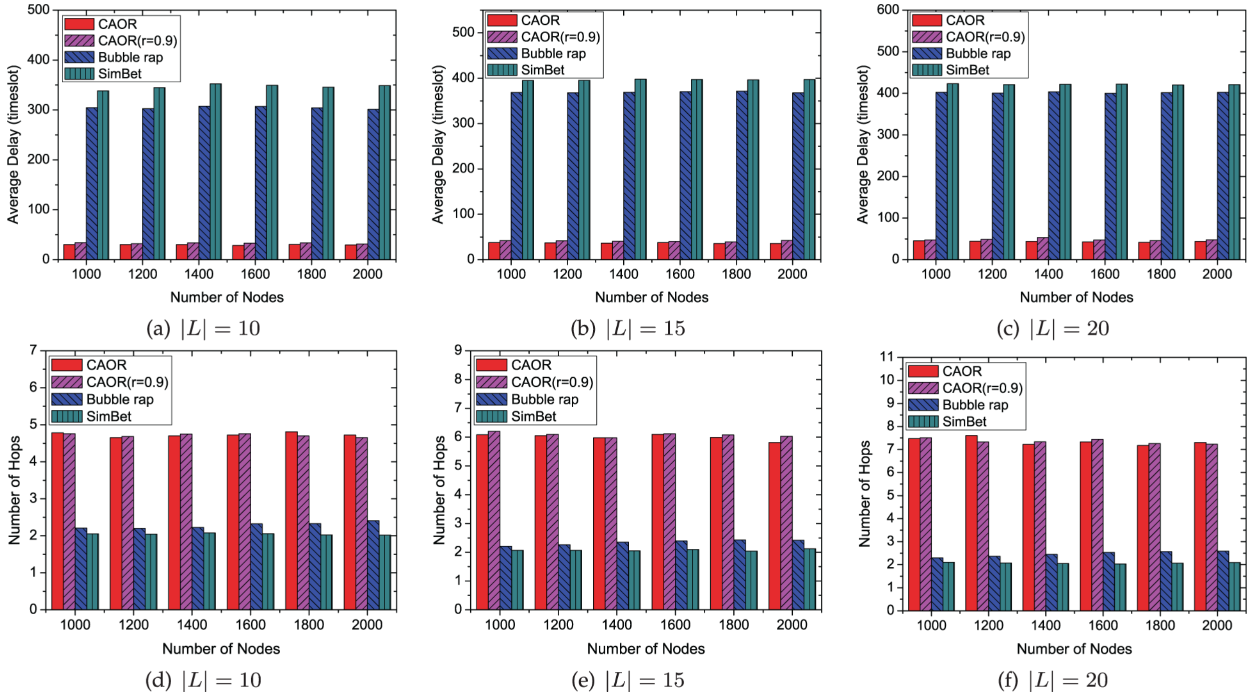



Fig. 10. Comparisons of delay and hops of CAOR, CAOR ($r = 0.9$), SimBet, and Bubble rap on the synthetic trace.

Social-aware algorithms assume that each node has some social characteristics (such as community, centrality, and similarity, etc.) and then exploits the knowledge to direct the routing decision, so as to improve the delivery ratio. The SimBet [3] algorithm exploits the ego network technique to locally compute the approximate centrality and similarity for each node. It then uses these characteristics to find bridge nodes for the message delivery. The Bubble rap [2] algorithm uses the $k$-clique algorithm to detect a community, ranks each node by calculating their centrality values, and then exploits the rank values of nodes to direct the routing decision. Besides, the algorithm in [5], a multicasting MSN algorithm, also uses the $k$-clique technique to detect the communities, and defines the cumulative contact probability of each node as its centrality, based on which, it finds the relay for message delivery. The Social-Greedy [21] algorithm calculates the social closeness for each node based on its social profile, and then greedily delivers the messages to the nodes that are socially closer to their destinations. Compared with the CAOR algorithm, these algorithms just exploit the social characteristics of nodes to improve the probability of meeting the destination for each message. However, this is still unpredictable, and thus cannot achieve the optimal result.

The CAOR algorithm is based on the home-aware community model. There are two features: one is that nodes are assumed to frequently meet at some homes and the cases that they occasionally encounter at other places are ignored; another is that the interval for each node's visit to homes follows the exponential distribution. In fact, most of the mobility research [17], [18], [22]–[24] has captured the characteristics of skewed location visiting preferences and the periodic reappearance of nodes at the same location from numerous real trace analyses. Moreover, they also point out that the intermeeting time of nodes in the real traces follows the power law distribution. However, Cai et al. [25] has proven that when the area is bounded, the distribution is the exponential distribution; otherwise, if there is no bound, the distribution becomes power law. Therefore, for simplicity, the exponential distribution is still widely adopted, as seen in [5], [14]. Compared with these mobility models, our model does not remove the homes. Instead, we utilize these homes to relay messages.

In addition, a lot of research also uses some auxiliary nodes to relay messages: research in [15] exploits "throwboxes" to relay messages, and research in [26] uses mobile "message ferries" to relay transfer messages, etc. Compared with our work, "message ferries" are mobile message relays, unlike our static community homes. The "throwbox" is just like our community home; however, their works mainly focus on the capacity and delivery delay of the Epidemic algorithm when adding "throwboxes" into the DTNs. To the best of our knowledge, this is the first work to use home-aware communities to find the optimal opportunistic routing among mobile nodes.

## 9 Conclusion

In this paper, we model an MSN into some overlapping home-aware communities, simplify the routing problem among many mobile nodes into the problem among some static communities, and propose the CAOR algorithm to achieve optimal opportunistic routing. Through theoretical analysis, we find out that optimal opportunistic routing only depends on a few nodes in the network. A change in behavior of most nodes would not affect the routing performance. We can thus achieve the optimal routing performance at a very low maintenance cost. Compared with previous social-aware algorithms, the optimal and predictable routing performance is the biggest advantage of the CAOR algorithm.

## APPENDIX

### A. Proof of Theorem 1

According to Definitions 3 and 4, we have that $B_{l,l'}(S)$ is exactly the expected delivery delay from $l$ to $l'$ via the first encountered node in $S$. Since the time interval that each node $v$ in $S$ encounters $l$ follows the exponential distribution with parameter $\lambda_{v,l}$, the probability density function of node $v$ becoming the first node meeting $l$ is $\lambda_{v,l}\prod_{v\in S}e^{-\lambda_{v,l}t}$. The delivery delay from $l$ to $l'$ via node $v$ is $t$ plus $D_{v,l'}=1/\lambda_{v,l'}$. Then, we have:

$$B_{l,l'}(S)=\sum_{v\in S}\left(\int_0^\infty \lambda_{v,l}\prod_{v\in S}e^{-t\lambda_{v,l}}(t+1/\lambda_{v,l'})dt\right)=\frac{1}{\sum_{v\in S}\lambda_{v,l}}+\frac{\sum_{v\in S}\lambda_{v,l}/\lambda_{v,l'}}{\sum_{v\in S}\lambda_{v,l}}. \quad (12)$$

### B. Proof of Theorem 2

We first prove $u\in\tilde{R}_i\Rightarrow D_{u,d}<D_{i,d}$ by contradiction. Assume that $u\in\tilde{R}_i$ while $D_{u,d}\geq D_{i,d}$. Then, we construct a new relay set $R^-=\tilde{R}_i-\{u\}$. By computing $D_{i,d}(\tilde{R}_i)$ and $D_{i,d}(R^-)$, we have:

$$D_{i,d}(\tilde{R}_i)=\sum_{v\in\tilde{R}_i}\left(\int_0^\infty \lambda_{i,v}\prod_{v\in\tilde{R}_i}e^{-t\lambda_{i,v}}(t+D_{v,d})dt\right)=\frac{1+\sum_{v\in\tilde{R}_i}\lambda_{i,v}D_{v,d}}{\sum_{v\in\tilde{R}_i}\lambda_{i,v}}, \quad (13)$$

$$D_{i,d}(R^-)=\sum_{v\in R^-}\left(\int_0^\infty \lambda_{i,v}\prod_{v\in R^-}e^{-t\lambda_{i,v}}(t+D_{v,d})dt\right)=\frac{1+\sum_{v\in R^-}\lambda_{i,v}D_{v,d}}{\sum_{v\in R^-}\lambda_{i,v}}, \quad (14)$$

Then, by comparing $D_{i,d}(\tilde{R}_i)$ and $D_{i,d}(R^-)$, we have:

$$D_{i,d}(\tilde{R}_i)-D_{i,d}(R^-)=\frac{\lambda_{i,u}}{\sum_{v\in R^-}\lambda_{i,v}}(D_{u,d}-D_{i,d}(\tilde{R}_i)). \quad (15)$$

That is:

$$D_{i,d}(\tilde{R}_i)\geq D_{i,d}(R^-)\Leftrightarrow D_{u,d}\geq D_{i,d}(\tilde{R}_i). \quad (16)$$

On the other hand, we have $D_{u,d}\geq D_{i,d}=D_{i,d}(\tilde{R}_i)$, according to the assumption. Thus, we can get $D_{i,d}(R^-)\leq D_{i,d}(\tilde{R}_i)$ from Eq. (16). This is a contradiction in that $\tilde{R}_i$ is the optimal relay set to minimize $D_{i,d}$ (if there are multiple relay sets to minimize $D_{i,d}$, we always select the one with the smallest set size in this paper). Therefore, the assumption is wrong, and we should have $D_{u,d}<D_{i,d}$.

Likewise, we can get $D_{u,d}<D_{i,d}\Rightarrow u\in\tilde{R}_i$ by the contradiction method. Assume that $D_{u,d}<D_{i,d}$ and meanwhile $u\notin\tilde{R}_i$. Then, we construct a new relay set $R^+=\tilde{R}_i+\{u\}$. By computing $D_{i,d}(R^+)$, we have:

$$D_{i,d}(R^+)=\sum_{v\in R^+}\left(\int_0^\infty \lambda_{i,v}\prod_{v\in R^+}e^{-t\lambda_{i,v}}(t+D_{v,d})dt\right)=\frac{1+\sum_{v\in R^+}\lambda_{i,v}D_{v,d}}{\sum_{v\in R^+}\lambda_{i,v}}. \quad (17)$$

Then, by comparing $D_{i,d}(R^+)$ and $D_{i,d}(\tilde{R}_i)$ in Eq. (13), we have:

$$D_{i,d}(R^+)-D_{i,d}(\tilde{R}_i)=\frac{\lambda_{i,u}}{\sum_{v\in R^+\lambda_{i,v}}}(D_{u,d}-D_{i,d}(\tilde{R}_i)). \quad (18)$$

That is:

$$D_{i,d}(R^+)<D_{i,d}(\tilde{R}_i)\Leftrightarrow D_{u,d}<D_{i,d}(\tilde{R}_i). \quad (19)$$

On the other hand, we have $D_{u,d}<D_{i,d}=D_{i,d}(\tilde{R}_i)$ according to the assumption. Thus, we can get $D_{i,d}(R^+)<D_{i,d}(\tilde{R}_i)$ from Eq. (19). This is a contradiction in that $\tilde{R}_i$ is the optimal relay set to minimize $D_{i,d}$. Therefore, the assumption is wrong, and we should have $u\in\tilde{R}_i$.

### C. Proof of Theorem 3

1. Since $v\notin\bigcup_{i=1}^m\tilde{S}_{l,l_i}$ means $v\in\bigcup_{i=1}^m(C_l\cap C_{l_i}-\tilde{S}_{l,l_i})$, then without loss of generality, we assume $v\in C_l\cap C_{l_i}-\tilde{S}_{l,l_i}$ and $v\in\tilde{R}_l$ to prove the first property by contradiction. Firstly, we construct a new relay set $R^-$ for the message delivery from $l$ to $d$ via $l_1,\ldots,l_m$. Let $R^-=\tilde{R}_l-(C_l\cap C_{l_i})+\tilde{S}_{l,l_i}$, and then compare the delay values, $D_{l,d}(R^-)$ and $D_{l,d}(\tilde{R}_i)$, the delivery delays from $l$ to $d$ via the new relay set $R^-$ and the optimal relay set $\tilde{R}_l$. In fact, the two delay values are the expected values of the delays via nodes in the two relay sets. Consider that a node in $R=\tilde{R}_l-(C_l\cap C_{l_i})$ first visits $l$ and is selected as the real relay. Its contributions to $D_{l,d}(R^-)$ and $D_{l,d}(\tilde{S}_{l,l_i})$ are the same. Thus, we only need to consider the contributions of the remaining nodes in $R^--R(=\tilde{S}_{l,l_i})$ and $\tilde{R}_l-R$ to $D_{l,d}(R^-)$ and $D_{l,d}(\tilde{R}_l)$, respectively. Since $\tilde{S}_{l,l_i}$ is the optimal relay set for the direct delivery from $l$ to $l_i$, we thus have $D_{l,l_i}(\tilde{S}_{l,l_i})+D_{l_i,d}<D_{l,l_i}(\tilde{R}_l-R)+D_{l_i,d}$. That is, the expected delay from $l$ to $l'$ via $R^-$ is even less than the delay via $\tilde{R}_l$. This is a contradiction in that $\tilde{R}_l$ is the optimal relay set. Therefore, the assumption about $v\in\tilde{R}_l$ is wrong, and we should have $v\notin\tilde{R}_l$.
2. We are still using the contradiction method, and assume that there exists an integer $i\in[1,m]$ that satisfies $\tilde{S}_{l,l_i}\nsubseteq\tilde{R}_l$ and $\tilde{S}_{l,l_i}\cap\tilde{R}_l\neq R=\emptyset$. We also construct a new relay set $R'=\tilde{R}_l-R+\tilde{S}_{l,l_i}$. Based on a similar analysis, as in part 1, we have that $D_{l,d}(R')$ is less than $D_{l,d}(\tilde{R}_l)$. This is a contradiction in that $\tilde{R}_l$ is the optimal relay set. Therefore, the assumption about $\tilde{S}_{l,l_i}\cap\tilde{R}_l\neq\emptyset$ is wrong, and the theorem is correct.

### D. Proof of Corollary 2

At first, we directly prove the second result, which also implies the first result. We consider the optimal opportunistic routing between $l$ and $l'$ via $\{v_1,\ldots,v_n\}$. If $v_{i+1}\in\tilde{S}_{l,l'}$, then we have $D_{v_{i+1},l'}<D_{l,l'}$ according to Theorem 2. Since $D_{v_i,l'}=\frac{1}{\lambda_{v_i,l'}}<D_{v_{i+1},l'}=\frac{1}{\lambda_{v_{i+1},l'}}$, we can get $D_{v_i,l'}<D_{l,l'}$. Using Theorem 2 again, we have $v_i\in\tilde{S}_{l,l'}$. Without loss of

generality, let the node in $\tilde{S}_{l,l'}$ with the largest expected delay to community location $l'$ be $v_k$, i.e., $v_k \in \tilde{S}_{l,l'}$. Then, $v_{k-1}, v_{k-2}, \ldots, v_1 \in \tilde{S}_{l,l'}$, i.e., $\tilde{S}_{l,l'} = \{v_1, \ldots, v_k\}$.

Now we prove the third result. Compare $D_{l,l'}(\{v_1, \ldots, v_i\})$ and $D_{l,l'}(\{v_1, \ldots, v_i, v_{i+1}\})$, we have:

$$D_{l,l'}(\{v_1, \ldots, v_{i+1}\}) < D_{l,l'}(\{v_1, \ldots, v_i\}) \Leftrightarrow D_{v_{i+1},l'} < D_{l,l'}(\{v_1, \ldots, v_i\}). \quad (20)$$

On the other hand, $v_{i+1} \in \tilde{S}_{l,l'}$, then we can get $D_{v_{i+1},l'} < D_{l,l'} < D_{l,l'}(\{v_1, \ldots, v_i\})$ according to Theorem 2. Thus, $D_{l,l'}(\{v_1, \ldots, v_{i+1}\}) < D_{l,l'}(\{v_1, \ldots, v_i\})$.

## Acknowledgment

This research was supported in part by the National Grand Fundamental Research 973 Program of China (Grant 2011CB302905); in part by the National Science and Technology Major Project (Grant 2011ZX03005-004-04, Grant 2012ZX03005009); in part by the National Natural Science Foundation of China (NSFC) (Grant 61379132, Grant 60803009, Grant 61003044, Grant 61170058); in part by the NSF of Jiangsu Province in China (Grant BK20131174, Grant BK2009150); and in part by NSF Grant ECCS 1231461, Grant ECCS 1128209, Grant CNS 1138963, Grant CNS 1065444, and Grant CCF 1028167.

## References

[1] J. Wu, M. Xiao, and L. Huang, "Homing spread: Community home-based multi-copy routing in mobile social networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2013, pp. 2319–2327.
[2] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble Rap: Social-based forwarding in delay tolerant networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2008, pp. 241–250.
[3] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant MANETS," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2007, pp. 32–40.
[4] J. Wu and Y. Wang, "Social feature-based multi-path routing in delay tolerant networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2012, pp. 1368–1376.
[5] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2009, pp. 299–308.
[6] Q. Li, S. Zhu, and G. Cao, "Routing in socially selfish delay tolerant networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2010, pp. 1–9.
[7] R. Lu, X. Lin, and X. S. Shen, "Spring: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2010, pp. 1–9.
[8] A. Vahdate and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke Univ., Tech. Rep. CS-2000–06, Jun. 2000 [Online]. Available: http://charlotte.ucsd.edu/classes/wi02/cse294/adhocepidemics.pdf
[9] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop Delay-Tolerant Netw. (WDTN)*, 2005, pp. 252–259.
[10] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic routing in intermittently connected networks," *Lect. Notes Comput. Sci.*, vol. 3126, no. 1, pp. 239–254, 2004.
[11] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2006, pp. 1–11.
[12] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. Conf. Applicat. Technol. Archit. Protocols Comput. Commun. (SIGCOMM'04)*, 2004, pp. 145–158.
[13] C. Liu and J. Wu, "Routing in a cyclic Mobispace," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2008, pp. 351–360.
[14] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 773–782, 2008.
[15] M. Ibrahim, P. Nain, and I. Carreras, "Analysis of relay protocols for throwbox-equipped DTNS," in *Proc. Int. Symp. Modeling Optim. Mobile Ad Hoc Wireless Netw. (WiOPT)*, 2009, pp. 1–9.
[16] D. Kotz, T. Henderson, I. Abyzov, and J. Yeo. (2009, Sep.) "CRAWDAD data set dartmouth/campus (v. 2009-09-09)," [Online]. Available: http://crawdad.cs.dartmouth.edu/dartmouth/ campus.
[17] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *Proc. ACM MobiCom Annu. Int. Conf. Mobile Comput. Netw.*, 2004, pp. 187–201.
[18] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2007, pp. 758–766.
[19] C. Liu and J. Wu, "An optimal probabilistic forwarding protocol in delay tolerant networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2009, pp. 105–114.
[20] 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, no. 1, pp. 63–76, 2008.
[21] K. Jahanbakhsh, G. C. Shoja, and V. King, "Social-greedy: A socially-based greedy routing algorithm for delay tolerant networks," in *Proc. Int. Workshop Mobile Opportunistic Netw. (MobiOpp)*, 2010, pp. 159–162.
[22] L. Jeremie, F. Timur, and C. Vania, "Evaluating mobility pattern space routing for dtns," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2006, pp. 1–10.
[23] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2006, pp. 49–60.
[24] P. Hui, A. Chaintreau, J. Scott, R. Gass, J. Crowcroft, and C. Diot, "Pocket switched networks and human mobility in conference environments," in *ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, 2005, pp. 244–251.
[25] H. Cai and D. Y. Eun, "Crossing over the bounded domain: From exponential to power-law inter-meeting time in MANET," in *Proc. ACM MobiCom Annu. Int. Conf. Mobile Comput. Netw.*, 2007, pp. 159–170.
[26] M. M. B. Tariq, M. Ammar, and E. Zegura, "Message ferry route design for sparse ad hoc networks with mobile nodes," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2006, pp. 37–48.

**Mingjun Xiao** received the PhD degree from University of Science and Technology of China, in 2004. He is an associate professor at the School of Computer Science and Technology, University of Science and Technology of China, Anhui. He is a visiting scholar at Temple University, under the supervision of Dr. Jie Wu, during 2012. He has served as a reviewer for many journal papers. His main research interests include delay tolerant networks and wireless sensor networks.

**Jie Wu** is chair and Laura H. Carnell professor with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA. Prior to joining Temple University, he was a program director at the National Science Foundation and a distinguished professor at Florida Atlantic University. His research interests include wireless networks, mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. He has published more than 550 papers in various journals and conference proceedings. He serves in the editorial boards of *IEEE Transactions on Computers* and *Journal of Parallel and Distributed Computing*. He was also general co-chair for IEEE MASS 2006, IEEE IPDPS 2008, and DCOSS 2009 and is the program co-chair for IEEE INFOCOM 2011. He has served as an IEEE Computer Society distinguished visitor. Currently, he is the chair of the IEEE Technical Committee on Distributed Processing (TCDP), an ACM distinguished speaker. He is the recipient of 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

**Liusheng Huang** received the MS degree in computer science from University of Science and Technology of China, Anhui, in 1988. He is a professor at the School of Computer Science and Technology, University of Science and Technology of China. His main research interests include delay tolerant networks and Internet of things. He serves on the editorial board of many journals. He has published 6 books and more than 200 papers.