Community-home-based Multi-copy Routing in Mobile Social Networks

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

Abstract—A mobile social network (MSN) is a special kind of delay tolerant network (DTN) composed of mobile nodes that move around and share information with each other through their carried short-distance wireless communication devices. A main characteristic of MSNs is that mobile nodes in the networks generally visit some locations (namely, community homes) frequently, while visiting other locations less frequently. In this paper, we propose a novel zero-knowledge multi-copy routing algorithm, *homing spread* (HS), for homogeneous MSNs, in which all mobile nodes share all community homes. HS is a distributed and localized algorithm. It mainly lets community homes spread messages with a higher priority. Theoretical analysis shows that HS can spread a given number of message copies in an optimal way when the inter-meeting time between any two nodes and between a node and a community home follows independent and identical exponential distributions, respectively. We also extend HS to the heterogeneous MSNs, where mobile nodes have different community homes. In addition, we calculate the expected delivery delay of HS, and conduct extensive simulations. Results show that community homes are important factors in message spreading. By using homes to spread messages faster, HS achieves a better performance than existing zero-knowledge MSN routing algorithms, including Epidemic (with a given number of copies), and Spray&Wait.

Index Terms—Community, delay tolerant networks, mobile social networks, routing.

1 Introduction

Mobile social networks (MSNs) are composed of mobile users that move around and use their carried wireless communication devices to share information via online social network services, such as Facebook, Twitter, etc. Recently, the short-distance communication model has also been adopted by encountered mobile users in MSNs to share information, such as multimedia, large-size files, etc., at a low cost. Such MSNs can be seen as a special kind of delay tolerant network (DTN). Fig. 1 shows a simple example. Like other DTNs, there are generally no stable end-to-end delivery paths in an MSN, due to the mobility of nodes. Therefore, delivering messages is a challenging issue. Many routing algorithms that are based on store-carry-and-forward schemes have been proposed to address this issue. The existing algorithms can simply be divided into two categories.

One category is *knowledge-based routing algorithms,*

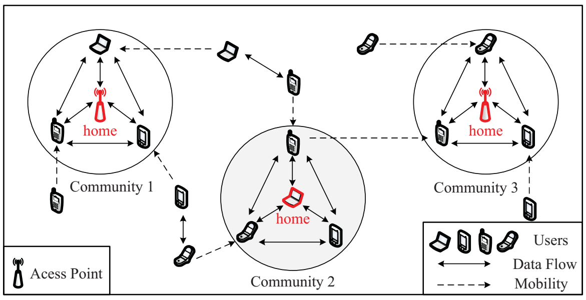


Fig. 1. An example of MSN: mobile users move around to form three communities, each of which can support a real or virtual throwbox through an access point in its home or a mobile user that visits its home most frequently.

which mainly includes probability-based algorithms (e.g., [2]–[6]) and social-aware algorithms (e.g., [7]–[12]). The nodes in these algorithms are assumed to have known some contact probabilities between nodes or some social characteristics of nodes, and then they use this knowledge to guide their message deliveries. However, it is difficult for each node to get to know the contact probabilities or social characteristics of other nodes in real MSNs.

Another category consists of *zero-knowledge routing algorithms*, which do not require any prior knowledge on the contact probabilities or social characteristics of nodes. The typical algorithms include Epidemic [13] and Spray&Wait [14]. Epidemic spreads messages to each encountered node through the flooding strategy. To avoid producing too many message copies, Epidemic in the real implementation generally limits the

- *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, 1805 N. Broad Street, Philadelphia, PA 19122. E-mail: jiewu@temple.edu*

This paper is an extended version of the conference paper [1] published in Infocom 2013. This research was supported in part by the National Grand Fundamental Research 973 Program of China (Grant No.2011CB302905); the National Science and Technology Major Project (Grant No. 2011ZX03005-004-04, 2012ZX03005009); the National Natural Science Foundation of China (NSFC) (Grant No. 61379132, 60803009, 61003044, 61170058), the NSF of Jiangsu Province in China (Grant No. BK20131174, BK2009150); and NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167.

maximum number of copies. Spray&Wait also limits the number of copies. Moreover, it adopts a binary splitting method to spread copies into the network until one message holder encounters the destination. Both of the algorithms assume that all nodes just randomly walk in a given area, and that nodes visit all locations in a uniformly random way. However, real MSNs generally do not follow this assumption, making them less efficient, as we will show later in this paper.

In fact, MSNs have social characteristics, compared to traditional DTNs. Nodes in an MSN generally visit some locations frequently, while visiting other locations less frequently, due to their different interests. The nodes that frequently visit the same location will form a community with a common interest, as shown in Fig. 1. The location is seen as the *home* of the community. Many mobility models from real MSN traces have captured this characteristic of skewed location visiting preferences [15]–[18]. Moreover, each community home (or simply, home) in real traces can support a *real throwbox*, a device that can locally store and forward messages, or can let the nodes that visit it most frequently act as *virtual throwboxes* [19]. Such social characteristics can be utilized to guide message deliveries so as to improve the routing performance.

To this end, we propose a zero-knowledge multi-copy MSN routing algorithm, *homing spread* (HS), in this paper. The objective is to minimize the expected delay of delivering each message from its source to its destination, while the copies of each message are no more than a given threshold. The algorithm consists of three phases. In the first phase, the source spreads copies quickly to community homes. In the second phase, the homes that have received more than one copy spread the message to other homes and mobile nodes (or simply nodes). Then, in the third phase, the destination fetches the message from any encountered *message holder*, which is either a mobile node or a home that has message copies. This algorithm makes use of the unbalanced location visiting characteristic and uses homes as special message holders. Thus, it can achieve a better performance than existing zero-knowledge routing algorithms. The main contributions are summarized as follows:

1) We first propose the HS algorithm for homogeneous MSNs, in which all mobile nodes share all community homes. Moreover, we show that HS is optimal in homogeneous MSNs when the inter-meeting time between any two nodes and between a node and a home follows independent and identical exponential distributions.
2) We also extend the HS algorithm to the heterogeneous MSNs, in which mobile nodes might have different community homes. We show that HS can still achieve good message delivery performance in the heterogeneous MSNs.
3) We construct a continuous Markov chain to calculate the expected delivery delay of HS and derive an upper bound. Moreover, we calculate the number of message copies required to bound the expected delivery delay to a given threshold.
4) We conduct extensive simulations on a synthetic MSN trace to evaluate HS. The results show that HS significantly outperforms the existing zero-knowledge multi-copy routing algorithms, including Epidemic with a given number of copies, and Spray&Wait.

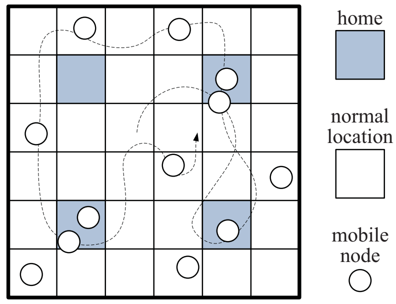


Fig. 2. The network model.

The remainder of the paper is organized as follows. We introduce the network model and problem in Section 2. Section 3 is the overview of HS. The detailed HS for homogeneous MSNs and the extended HS for heterogeneous MSNs are presented in Sections 4 and 5, respectively. Section 6 gives a theoretical performance analysis on HS. In Section 7, we evaluate the performance of HS through extensive simulations. After reviewing the related work in Section 8, we conclude the paper in Section 9.

2 Network Model & Problem

In this section, we introduce the network model, followed by the problem.

2.1 Network Model

We consider a typical MSN that is composed of a number of mobile nodes and many locations. Each node frequently visits a few locations, called *community homes* or *homes*, while the other locations, called *normal locations*, are visited less frequently. Each node might have multiple homes. Many real MSNs follow this unbalanced-visiting characteristic. For example, previous works have observed that 50% of mobile users in the Dartmouth campus Wi-Fi network spent over 74.0% of their time at a location [15]–[18]. Moreover, we assume that each home has a *throwbox* that can locally store and forward messages. Many real applications can support throwboxes, such as the roadside units in vehicular ad hoc networks, the base stations in delay tolerant networks, etc. [19], [20]. Even though there are no real throwboxes in some homes, we can let the nodes that most frequently visit these homes act as the virtual throwboxes. In fact, the works in [15]–[18] have also observed that some nodes remain close to some locations about 98.7% of the time, which can be used as the virtual throwboxes of these locations. Moreover, literature [10] has pointed out that virtual throwboxes will only result in a little bit of performance degradation, compared to real throwboxes. In addition, we assume that the throwbox in each home has enough cache space to store messages from visited mobile nodes. This is reasonable since a real throwbox is generally equipped with a large cache. If a virtual throwbox has limited cache, we can let multiple nodes that frequently visit the home act as the virtual throwboxes at the same time, so that they can also provide a large cache together [10].

More specifically, we consider that n mobile nodes $V = \{1, 2, \cdots, n\}$ independently and randomly walk on a $\sqrt{m} \times \sqrt{m}$ 2D grid, among which there are h ($h \ll n$) homes $H = \{l_1, \cdots, l_h\}$ and $m - h$ normal locations $L = \{l_{h+1}, \cdots, l_m\}$, as shown in Fig. 2. Each home has a real or a virtual throwbox. The homes of node i ($i \in V$) are denoted by home set $H_i \subseteq H$. Each node visits either its home with a relatively high probability, or a normal location with a very low probability. The visited home and normal location of each node are randomly selected from its home set and normal locations, respectively.

2.2 Problem

In this paper, we consider two MSN settings: the homogeneous setting and the heterogeneous setting. They are defined as follows:

Definition 1: The *homogeneous setting* refers to that all nodes in the MSN share all of the h homes. That is to say, $H_i = H$ for each $i \in V$.

Definition 2: The *heterogeneous setting* means that each node i might have a different home set H_i, each home in which is randomly selected from H. That is, $H_i \subseteq H$. Other homes outside of H_i are seen as normal locations for node i.

Under both the homogeneous setting and the heterogeneous setting, we study the zero-knowledge multi-copy routing problem. Here, *zero-knowledge routing* means that each node in the MSN is unaware of other nodes. That is to say, each node in the MSN does not know the home sets of other nodes.

Our objective is to minimize the delivery delay for a given number of message copies C ($1 < C < n/2$; the concrete value of C will be determined in Section 6.3, which is much less than $n/2$). For simplicity, a visit to a home is known as *homing*, but when a message holder meets another node at a normal location, it is known as *roaming*. Then, we plan to address the following challenges:

- What is the optimal way for a message holder to spread copies during homing and roaming?
- Once a home receives some message copies, how should it further spread these copies?
- What is a general way for a mobile destination to obtain a copy?

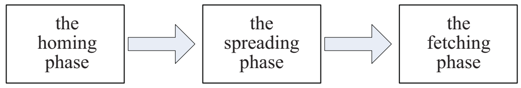


Fig. 3. The framework of Homing Spread.

3 Overview: Homing Spread (HS)

To solve the above problem, we first propose the zero-knowledge multi-copy routing algorithm, homing spread (HS), for the homogeneous setting. Then, we extend HS to the heterogeneous setting. Since each node has a relatively high probability of visiting homes, the basic idea of HS is to let the homes have a higher priority to get the copies, so as to maximize the probability that the destination meets a message holder (i.e, the homes or nodes that have message copies). More specifically, HS consists of three phases: *homing*, *spreading*, and *fetching*, as shown in Fig. 3.

1) In the homing phase, the source sends copies quickly to homes. Upon reaching the first home, the message holder (which includes the source) dumps all copies into the throwbox of the home. When roaming occurs (i.e., a message holder meets another node at a normal location before reaching a home), copies are split between the two nodes and both become message holders.
2) In the spreading phase, the homes with multiple copies spread these copies to other homes and mobile nodes. These homes first spread their copies to each node that visits them by splitting the copies between themselves and those visiting nodes. Then, each node that receives copies spreads these copies to other homes and mobile nodes. In the splitting of copies between the homes and the visiting nodes, each home always keeps at least one copy through its throwbox.
3) In the fetching phase, the destination fetches the message when it meets any message holder for the first time, which can be either a home or a mobile node.

In the above schemes, two points need to be emphasized. The first point is that the three phases might not follow a strict order. There might be overlaps among them in probability. For example, it is possible that the delivery of a message copy enters the second phase while the delivery of another copy might still be in the first phase. The third phase also might occur before the second phase. The second point is that, when a message holder first visits a home, it will dump all copies into the home, and then it immediately enters the second phase to receive copies from the home.

4 HS: Homogeneous MSNs

In this section, we propose a homing scheme and a spreading scheme for the message spreading in

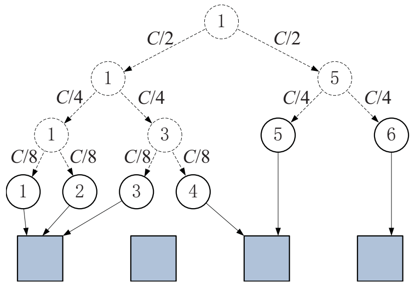


Fig. 4. The binary homing scheme.

Algorithm 1 The Homing Spread (HS)

1: **for** each mobile node i **do**
2: **if** node i encounters another node j **then**
3: **if** node j is the destination **then**
4: node i sends the message to j;
5: **if** nodes i and j have c_i and c_j message copies **then**
6: node i holds $\lceil c_i/2\rceil+\lfloor c_j/2\rfloor$ copies through exchange with node j;
7: **if** node i visits a home l **then**
8: node i sends all its copies to l;
9: **if** $l \in H_{\oplus}$ or i is the destination **then**
10: l sends a copy to node i.

the homogeneous MSNs. Based on these, we present the detailed HS algorithm. We focus on one message delivery only, but the results can be applied to multiple messages as long as each node, including home, has sufficient cache space and the link has enough bandwidth. In addition, we also show that, when the inter-meeting time between any two nodes and between a node and a home follows independent and identical exponential distributions, HS is optimal in terms of minimizing the expected delivery delay in homogeneous MSNs.

4.1 The Homing Phase

In the homing phase, the source tries to send the message to the homes first. If the source encounters other nodes before it reaches a home, it will give some of its copies to the encountered node, and will let the node jointly send the copies to homes. The more nodes that the message copies are sent to before reaching homes, the smaller the delay of the next two phases will be. Thus, the source needs to spread the copies to as many other nodes as possible before they reach the homes. To this end, we adopt the following homing scheme:

Definition 3: (*Binary Homing Scheme*): Each message holder sends all of its copies to the first (visited) home. If the message holder encounters another node before it visits a home, it binary splits the copies between them.

Fig. 4 shows an example of the binary homing scheme. Message copies are binary split until they reach the homes.

4.2 The Spreading Phase

In the spreading phase, the homes which have more than one copy spread their extra copies to other homes and nodes. Let $H_{\oplus}$, $H_{\ominus}$ and $H_{\odot}$ ($H = H_{\oplus} + H_{\ominus} + H_{\odot}$) denote the homes with more than one copy, the homes with only one copy, and the homes without copies, respectively. Then, we adopt the following spreading scheme.

Definition 4: (*1-Spreading Scheme*): Each home $l_i \in H_{\oplus}$ spreads a copy to each node in the same home until only one copy remains, so that $l_i \in H_{\ominus}$ after the spreading. If such a node with one copy later visits another home $l_j \in H_{\odot}$, the node sends the copy to that home, so that $l_j \in H_{\ominus}$ after the visit.

Using the *1*-spreading scheme, as shown in Fig. 5, each home will have at most one copy. Then, after the spreading phase, there would be C message holders, including h homes and $C-h$ nodes outside the homes, or C homes if $C \leq h$. Each of them only has one copy.

4.3 The Fetching Phase

In the fetching phase, the destination just fetches the message once it encounters a message holder. This message holder might be in the homing phase or the spreading phase. The worst case is that all message copies have finished the spreading phase before the destination gets the message.

4.4 The HS Algorithm

We present the HS algorithm, as shown in Algorithm 1. Algorithm 1 is a distributed algorithm, in which each node only needs to exchange the copies with the encountered node or home. Note that we do not distinguish the three phases when nodes exchange the copies. This is because the message exchange in this algorithm is compatible with each phase. In fact, if the node encounters the destination, which falls into the third phase, the node will send the message to the destination in Steps 3-4. If two nodes in the first phase encounter each other, they will send half of their copies to the other one in Steps 5-6, where $\lceil c_i/2\rceil$ and $\lfloor c_j/2\rfloor$ are the ceiling of $c_i/2$ and the floor of $c_j/2$, respectively. If two nodes in the second phase encounter each other, the message exchange scheme in Steps 5-6 is still correct. When a node visits a home, no matter which phase it falls under, it is compatible for the node to send all of its copies to the home and to receive a copy from the home if it has extra copies, as shown in Steps 7-10. Note that in Algorithm 1, the part for node j is the same as the one for node i (by exchanging i and j).

4.5 Optimality of HS

In this paper, we are mainly concerned with the average delay performance of HS. For simplicity, we assume that the inter-meeting time between any two nodes and between a node and a community home follows exponential distributions with parameters λ and Λ ($\Lambda \gg \lambda$, more specifically $\Lambda > C\lambda$), respectively. Such an assumption is widely adopted to analyze the average performance of a routing algorithm (e.g., [14]).

First, we consider the homing phase, in which the binary homing scheme is adopted. Note that this scheme binary splits message copies between encountered nodes before these copies reach homes, just like the binary spraying in Spray&Wait [14]. It has been proven to be the fastest way to spread message copies among mobile nodes. Thus, we can directly get the following lemma:

Lemma 1: The binary homing scheme can spread the C message copies to the maximum number of nodes before they reach the homes.

In addition, in terms of the homing phase, we have another lemma:

Lemma 2: If all nodes have the same number of homes, i.e., $|H_1| = |H_2| = \cdots = |H_n| = h'$, the expected delay of each message copy reaching a home is always $\frac{1}{h'\Lambda}$, no matter which splitting scheme is adopted.

Proof: We first consider the case where the source in the homing phase reaches a home without meeting any other nodes. Since the inter-meeting time between each node and a given home follows the exponential distribution with the parameters Λ, the expected delay for a visit to this home is $\frac{1}{\Lambda}$. Thus, the expected delay of the source visiting one of its h' homes is $\frac{1}{h'\Lambda}$.

Now, we consider the case where the source meets another node before it reaches a home. Note that the meeting will not change the expected delay of the message copies in the source reaching a home. Thus, we only focus on the copies that are spread to the encountered node in this meeting. Assume that the source meets the node at time t_1, and the encountered node reaches a home at time t. The corresponding probability density is $\Lambda e^{-h'\Lambda t_1} e^{-h'\Lambda(t-t_1)} = \Lambda e^{-h'\Lambda t}$. Then, the expected delay of the copies in the encountered node reaching a home is $\int_0^\infty \Lambda t e^{-h'\Lambda t} dt = \frac{1}{h'\Lambda}$. That is to say, the meeting also does not change the expected delay of the copies in the encountered node reaching a home. Thus, the lemma holds. □

Second, we consider the spreading phase, in which the *1*-spreading scheme is adopted. Note that the inter-meeting time between a node and a home follows independent and identical exponential distributions. Moreover, each node has a much larger probability of visiting a home than meeting another node. A home can spread the copies to other nodes more quickly than a node can. Thus, the *1*-spreading scheme can spread the message copies to other nodes most quickly. That is:

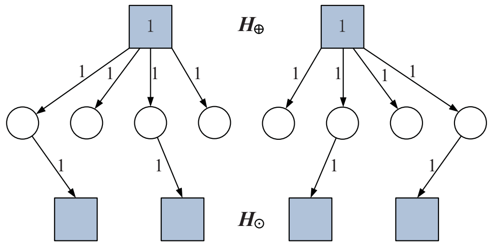


Fig. 5. The *1*-spreading scheme.

Lemma 3: The *1*-spreading scheme can spread message copies from a home to the maximum number of nodes at the fastest speed when $\Lambda > C\lambda$.

Proof: We compare the speeds of a home and a mobile node spreading k ($1 < k \leq C$) message copies to k of $n-1$ nodes.

First, we derive the delay for a home to spread the k message copies. Since the inter-meeting time between a home and a node follows the exponential distribution with parameter Λ, the expected delay for this node visiting the home is $\frac{1}{\Lambda}$. Note that, any one of the $n-1$ nodes might receive the first copy. Thus, the expected delay of the first node to receive a copy is $\frac{1}{(n-1)\Lambda}$. Moreover, the expected delay of the i-th ($1 \leq i \leq k$) node to receive a copy, denoted by D_h^i, satisfies:

$$D_h^i = \sum_{j=1}^{i} \frac{1}{n-j} \cdot \frac{1}{\Lambda} \tag{1}$$

Second, we analyze the delay for a mobile node to spread the k message copies. According to Lemma 1, the binary spreading scheme is the fastest spreading manner, when the inter-meeting time between nodes follows independent and identical exponential distributions. By using this spreading scheme, the source of the k message copies will send half of its copies to its first encountered node, which might be any one of the $n-1$ nodes. The corresponding expected delay is $\frac{1}{(n-1)\lambda}$. After each of them meets another node, respectively, the two encountered nodes become the second and third nodes to receive copies. The corresponding expected delays are $\frac{1}{(n-1)\lambda} + \frac{1}{(n-2)\lambda}$ and $\frac{1}{(n-1)\lambda} + \frac{1}{(n-3)\lambda}$, respectively. For generality, the expected delay of the i-th ($1 \leq i \leq k$) node to receive a copy, denoted by D_n^i, satisfies:

$$D_n^i = \sum_{j=0}^{\lfloor \log_2^i \rfloor} \frac{1}{n - \lfloor \frac{i}{2^j} \rfloor} \cdot \frac{1}{\lambda} \tag{2}$$

It is easy to verify $D_n^i > D_h^i$ when $1 \leq i \leq 4$. Now, we consider the case of $4 < i \leq k \leq C$. Note that $\frac{in}{(n-i)\log_2^i} < \frac{Cn}{(n-C)\log_2^C} < \frac{Cn}{(n-n/2)\log_2^C} < C < \frac{\Lambda}{\lambda}$. Then, we can get $D_h^i \leq \frac{i}{(n-i)\Lambda} < \frac{\log_2^i}{n\lambda} \leq D_n^i$. Thus, we have $D_n^i > D_h^i$ for each i ($1 \leq i \leq k$). The lemma holds. □

Algorithm 2 The Extended Homing Spread

```
for each mobile node i do
  if node i encounters another node j then
    if node j is the destination then
      node i sends the message to j;
    if nodes i and j have c_i and c_j message copies then
      node i holds ⌈(1 − α_ij)c_i⌉ + ⌊α_ji c_j⌋ copies through exchange with node j;
  if node i visits a home l then
    node i sends all its copies to l;
    if l ∈ H_⊕ or i is the destination then
      l sends a copy to node i.
```

Based on Lemmas 1-3, we get that HS is optimal.

Theorem 4: (*Optimality of HS*): HS can achieve the minimum expected delay when the inter-meeting time between any two nodes and between each node and each home follows the independent and identical exponential distributions with parameters λ and Λ ($\Lambda > C\lambda$), respectively.

Proof: We consider the binary homing scheme in the first phase and the *1*-spreading scheme in the second phase. According to Lemma 1, the binary homing scheme can spread the message copies to the maximum number of nodes before they reach the homes. Meanwhile, this scheme will let the maximum number of homes receive these copies. Moreover, according to Lemma 2, this scheme will not increase the delay of each copy reaching a home. As a result, it can maximize the probability of the destination meeting a message holder in the first phase, and can contribute to the spreading phase most. According to Lemma 3, we can get that the *1*-spreading scheme can spread message copies from homes to the maximum number of nodes at the fastest speed. That is to say, this scheme can maximize the probability of the destination meeting a message holder in the second phase. Thus, HS is optimal. □

5 HS: Heterogeneous MSNs

In this section, we extend the HS algorithm from the homogeneous setting to the heterogeneous setting, where each node might have a different home set, but all of them will form the overlapped home set H. As a zero-knowledge routing algorithm, the source in HS does not know which homes the destination is related to. Without loss of generality, the source treats every home as a potential home of destination. Then, the objective is still to spread the message copies to each home. If there are extra copies, it will spread them to other mobile nodes.

5.1 The Extended HS

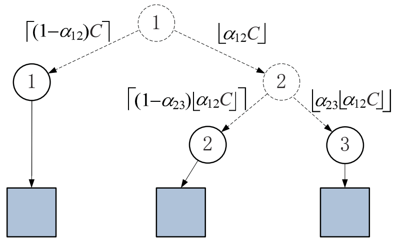


Fig. 6. The proportional homing scheme.

First, we consider the homing phase. Since the nodes in the heterogeneous setting have different home sets, the expected delays for them to visit a home will be different. In general, the more homes a node has, the more quickly the node will send its copies to a home. Thus, when two nodes that have copies meet, the node with more homes should hold more copies, so as to minimize the average delay for these copies to be delivered to the homes. On the other hand, in order to minimize the delays of the next two phases, we also need to let these copies spread to as many homes as possible. In terms of this objective, each pair of encountered nodes should equally split their copies. Thus, there is a tradeoff in the splitting of copies. To this end, we adopt the following homing scheme in HS.

Definition 5: (*Proportional Homing Scheme*): Each node with message copies sends its copies to the first (visited) home. If a node i that has c copies encounters another node j before it visits a home, node i will split these copies between them by sending out $\lfloor \alpha_{ij} c \rfloor$ copies and holding the remaining copies by itself, where $\alpha_{ij} = \frac{|H_j|}{|H_i|+|H_j|}$.

In the proportional homing scheme, α is a ratio of copy-splitting between encountered nodes. Each pair of nodes generally has a different ratio of copy-splitting. Determining the optimal α for each pair of nodes will lead to an exponential computation overhead. Here, we simply let the message copies be split in proportion to the numbers of homes of encountered nodes, since the number of homes of a node represents the message-spreading capability of this node. Fig. 6 shows an example of the proportional homing scheme. Message copies are proportionally split until they reach the homes. When $\alpha = 0.5$, the proportional homing scheme becomes the binary homing scheme.

Second, we consider the spreading phase. As a zero-knowledge routing algorithm, each home in HS does not know the home set of visiting nodes. As a result, it cannot distinguish the visiting nodes, as to know which one can spread messages faster than the others. Thus, we still adopt the *1*-spreading scheme in the spreading phase, in which the homes let each visiting node spread its copies without distinction.

Based on the proportional homing scheme and the *1*-spreading scheme, we present the extended HS algorithm, as shown in Algorithm 2. Compared to

Algorithm 1, the extended HS uses the proportional homing scheme in Step 6. Note that, when we set $\alpha = 0.5$, the proportional homing scheme will degrade to be the binary homing scheme. Thus, Algorithm 1 can be seen as a special case of Algorithm 2.

5.2 Discussion

In heterogeneous MSNs, the extended HS algorithm can still achieve a good result, especially when each node has the same number of homes that are randomly selected from H. That is:

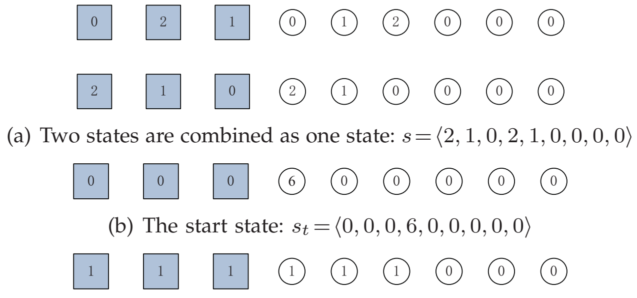
Theorem 5: When each node has the same number of homes that are randomly selected from H (i.e., $|H_1| = |H_2| = \cdots = |H_n| = h'$), and the inter-meeting time between any two nodes and between each node and its homes follows the independent and identical exponential distributions with parameters λ and Λ ($\Lambda > C\lambda$), the extended HS algorithm is still the best zero-knowledge routing algorithm.

Proof: First, when each node has the same number of homes, the proportional homing scheme in the extended HS algorithm becomes the binary homing scheme. According to Lemma 1, the proportional homing scheme in this case can still spread the C message copies to the maximum number of nodes before they reach the homes, which can maximize the probability that the destination meets a message holder in the homing phase, and also can maximize the number of homes receiving copies. Second, when each node has the same number of homes, the expected delay for each message copy reaching a home is $\frac{1}{h'\Lambda}$, since the inter-meeting time between each node and its homes follows the independent and identical exponential distributions with parameter Λ. That is to say, Lemma 2 holds in this case. Third, Lemma 3 is also right, since each home is randomly selected from H. Like Theorem 4, we get that HS is the best zero-knowledge routing algorithm in the heterogeneous MSNs, where each node has the same number of homes. □

When nodes' numbers of homes differ, the expected delay for each message copy to reach a home in the homing phase might be different. That is to say, Lemma 2 does not hold. As a result, the extended HS algorithm will be not optimal in this case. Nevertheless, this algorithm still can achieve a good performance. This is because the proportional homing scheme in this algorithm takes the message-spreading ability of homes and mobile nodes into account at the same time, and makes a simple balance between them.

6 Performance Analysis

In this section, we formally analyze the expected delivery delay of HS. First, we adopt the continuous Markov chain to compute the expected delivery delay. Since it is hard to derive the closed formula, we derive an upper bound, whereby we determine the



(a) Two states are combined as one state: $s = \langle 2, 1, 0, 2, 1, 0, 0, 0, 0\rangle$

(b) The start state: $s_t = \langle 0, 0, 0, 6, 0, 0, 0, 0, 0\rangle$

(c) The optimal state: $s_o = \langle 1, 1, 1, 1, 1, 1, 0, 0, 0\rangle$

Fig. 7. An example of network state, in which the numbers in the squares and the circles are the numbers of copies held by the homes and nodes, respectively ($h = 3$, $n = 6$, $C = 6$).

number of message copies. For generality, we focus on the extended HS for heterogeneous MSNs in the following.

6.1 Computing the Expected Delivery Delay

We construct a state transition graph and use a continuous Markov chain to compute the expected delivery delay of HS.

First, we define a concept of network *state*, which is used to describe the distribution of message copies in the whole network.

Definition 6: (*State* of Network s): s is a vector with $h{+}n$ components, i.e., $s = \langle s_1, s_2, \cdots, s_h, s_{h+1}, \cdots, s_{h+n}\rangle$ ($s_1 \geq \cdots \geq s_h$; $s_{h+1} \geq \cdots \geq s_{h+n}$), in which the i-th component s_i represents the number of message copies held by the i-th home (if $i \leq h$) or node $i-h$ (if $i > h$).

Here, for simplicity, we let $s_1 \geq s_2 \geq \cdots \geq s_h$ and $s_{h+1} \geq \cdots \geq s_{h+n}$. If $s_i < s_j$ ($1 \leq i < j \leq h$ or $h+1 \leq i < j \leq n$), we exchange s_i and s_j, and treat the states before and after the exchange as the same state, so as to decrease the number of total states. Then, based on Definition 6, there are two special states. One is the *start state*, denoted by $s_t = \langle 0, \cdots, 0, s_{h+1} = C, 0, \cdots, 0\rangle$. Another is the state that all message copies have finished the homing phase and the spreading phase, but none of them are received by the destination. In this state, the probability of the destination fetching a message copy is the largest. Thus, we call it the *optimal state*, denoted by $s_o = \langle 1, 1, \cdots, 1, 0, 0 \cdots\rangle$. Fig. 7 shows three states of a simple MSN, where $h = 3$, $n = 6$, and $C = 6$.

Now, we determine all possible states in the *state transition graph*. According to Definition 6, a state $s = \{s_1, \cdots, s_h, \cdots, s_{h+n}\}$ satisfies:

$$\begin{cases} \sum_{i=1}^{h+n} s_i = C \\ s_1 \geq s_2 \geq \cdots \geq s_h \\ s_{h+1} \geq \cdots \geq s_{h+n} \end{cases} \quad (3)$$

Let S denote the state space. Then, S is the solution space of Eq. 3.

Algorithm 3 Compute the expected delivery delay

1: Construct the state transition graph G:
2: Determine the state set S;
3: Determine $\rho_{s,s'}(t)$ for each pairwise $s, s' \in S$;
4: Set $f_{s,s_e}(t) = 0$ ($\forall s \in S$);
5: Delete all states ($\neq s_t$) whose in-degree is 0;
6: Let array $d_{out}(s)$ =out-degree of s ($\forall s \in S$);
7: **while** $S \neq \emptyset$ **do**
8: **for each** $s' \in S$ that $d_{out}(s') = 0$ **do**
9: $S = S - \{s'\}$;
10: **for each** $s \in S$ that $\rho_{s,s'}(t) \neq 0$ **do**
11: **if** s' is s_e **then**
12: $f_{s,s_e}(t) = \rho_{s,s'}(t)$;
13: **else**
14: $f_{s,s_e}(t) = f_{s,s_e}(t) + \int_0^t \rho_{s,s'}(x) f_{s',s_e}(t-x)dx$;
15: $d_{out}(s) = d_{out}(s) - 1$;
16: **Output**: $\int_0^\infty t f_{s_t,s_e}(t)dt$;

Second, we determine the state transition functions. For two arbitrary states $s, s' \in S$, we use $\rho_{s,s'}(t)$ to denote the *probability density function* about the time t that it takes for the state transition from s to s'. The transition probability is zero if more than two components of s, s' are different. If there are exactly two different components between s and s', we can check whether there is a state transition that follows the HS algorithm, and then the corresponding probability density function can be calculated. Assume that the i-th and j-th components are different. If $i, j > h$, this means that nodes i and j encounter each other. Then, checking the values of s_i, s_j, s'_i, s'_j, we can determine whether they follow the binary/proportional homing scheme of HS. If their values do not follow the scheme, there is still not a state transition between them. Otherwise, the corresponding probability density function is the probability density that nodes i and j will encounter each other, while other nodes and homes will not encounter to exchange their message copies. In the same way, we can determine the probability density for the case where either i or j is a home.

Finally, we add the *end state* into the graph, denoted by s_e, which is related to the third phase. In fact, each state in the first phase and the second phase can be directly transited to be the end state when a message holder encounters the destination. Thus, each state has a direct edge to the end state s_e. The corresponding probability density function is the probability density that one of the message holders encounters the destination, while other nodes and homes will not encounter to exchange their copies.

Based on the above method, we construct the state transition graph $G\langle S, \{\rho_{s,s'}(t) | s, s' \in S\}\rangle$. Moreover, according to the binary/proportional homing scheme in the first phase and the *1*-spreading scheme in the second phase, the state transition is irreversible, which will not lead to a loop. That is, the state transition graph G is a directed acyclic graph.

After constructing the state transition graph, we can calculate the expected delivery delay of the message, which is equal to the expected delay for the transition from the start state to the end state. To this end, we derive the cumulative probability density function for the state transition from the start state to the end state, denoted by $f_{s_t,s_e}(t)$. Regarding the cumulative probability density function, we have the following theorem.

Theorem 6: Consider an arbitrary state s and its next states $N_s = \{s' | \rho_{s,s'}(t) > 0, s' \in S\}$. Then, the cumulative probability density functions for the state transitions from these states to s_e satisfy:

$$f_{s,s_e}(t) = \sum_{s' \in N_s} \int_0^t \rho_{s,s'}(x) f_{s',s_e}(t-x)dx. \quad (4)$$

Proof: For each next state s' ($\in N_s$) of state s, the cumulative probability density function for the state transition from s to s_e via s' is a convolution $\int_0^t \rho_{s,s'}(x) f_{s',s_e}(t-x)dx$, where $\rho_{s,s'}(x)$ is the probability density for the state transition from s to s' at time x, and $f_{s',s_e}(t-x)$ is the probability for the state transition from s' to s_e at time $t-x$. Then, we can get the total cumulative probability density function for the state transition from s to s_e, i.e., $f_{s,s_e}(t) = \sum_{s' \in N_s} \int_0^t \rho_{s,s'}(x) f_{s',s_e}(t-x)dx$. □

This theorem shows that if the cumulative probability density function for the state transition from each next state of s to s_e has been calculated, then the cumulative probability density function of the state s can also be derived. Then, we can adopt a backward derivation method to get the cumulative probability density functions of all states, since the state transition graph G is a directed acyclic graph. Based on this backward derivation, we can eventually get $f_{s_t,s_e}(t)$. Then, the expected delay for the message delivery from the source to the destination is $\int_0^\infty t f_{s_t,s_e}(t)dt$.

Based on the above method, we present Algorithm 3 to calculate the expected delivery delay from the source to the destination. Steps 1-3 construct the state transition graph. Step 5 deletes the invalid states. In step 6, an array is used to record the out-degrees of each state in the graph. A state s' with a zero out-degree means that the cumulative probability density function $f_{s',s_e}(t)$ has been determined. Then, it will be deleted from the graph in Step 9. Accordingly, the cumulative probability density functions for the state transition via this state are updated in Steps 10-15. By repeating this process, all of the cumulative probability density functions can be determined. Then, the algorithm outputs the results in Step 16. The overhead of Algorithm 3 is dominated by Steps 11-14, which will be executed within $O(|S|^2)$. Note that we directly use the cumulative probability density functions in

Algorithm 3 for simplicity. In fact, these cumulative probability density functions can be realized in the real implementation, since they are composed of exponential functions that can be described by pairwise coefficients and exponents.

6.2 A Simple Example

Here, we present an example to calculate the expected delivery delay of HS by using Algorithm 3. Consider a simple homogeneous MSN, in which $h = 2$, $n = 5$, $C = 2$, $\Lambda = 0.4$, and $\lambda = 0.05$. Then, the state transition graph is constructed as follows:

According to Eq. 3, we first derive all network states s_t, s_1, s_2, s_o, s_e, as shown in Fig. 8. In each state (except the end state s_e), the first two components are the message copies of homes, and the remaining components are the copies of nodes. State s_t is the start state where the source holds two copies. State s_1 is an intermediate state where two nodes each hold a copy. State s_2 is another intermediate state where a home and a node hold a copy, respectively. State s_o is the optimal state where two homes each hold a copy.

The probability density function for each state transition is also determined. For instance, the state transition from s_t to s_2 means that the source visits a home before it encounters any other nodes. The corresponding probability density function is $\rho_{s_t,s_2}(t) = 2\Lambda e^{-2\Lambda t - 4\lambda t} = 0.8e^{-t}$. The state transition from s_t to s_1 indicates that the source encounters another node (any node except the source and the destination) before it visits a home. The corresponding probability density function is $\rho_{s_t,s_1}(t) = 3\lambda e^{-2\Lambda t - 4\lambda t} = 0.15e^{-t}$. The state transition from s_2 to s_o means that a node with a copy visits the home in $H_{\odot}$ before it meets the destination, and before the destination visits the home in $H_{\ominus}$. The corresponding probability density function is $\rho_{s_2,s_o}(t) = \Lambda e^{-2\Lambda t - \lambda t} = 0.4e^{-0.85t}$. In the same way, all state transition functions are derived, as shown in Fig. 8.

After the state graph construction, Algorithm 3 uses the backward derivation from state s_e to compute the cumulative probability density functions. First, the cumulative probability density function of s_o is determined, i.e., $f_{s_o,s_e}(t) = \rho_{s_o,s_e}(t) = 0.8e^{-0.8t}$. Next, $f_{s_2,s_e}(t)$ is determined, i.e., $f_{s_2,s_e}(t) = \rho_{s_2,s_e}(t) + \int_0^t \rho_{s_2,s_o}(x) f_{s_o,s_e}(t-x)dx$, and so on. Finally, $f_{s_t,s_e}(t)$ is derived. Then, we can get that the expected delivery delay is 2.81.

It is worth noting that we also calculate the expected delivery delay for the case where $h = 0$, which corresponds to Spray&Wait. The corresponding expected delivery delay is 12.25. That is, compared to Spray&Wait, our algorithm reduces the expected delivery delay by 77.1% for this example.

6.3 The Upper Bound of Expected Delivery Delay

Although we can calculate the expected delivery delay through Algorithm 3, it is hard to derive a close formula. Here, we derive an upper bound of the expected delivery delay, by which we can derive the number of required message copies C to ensure a given expected delivery delay performance.

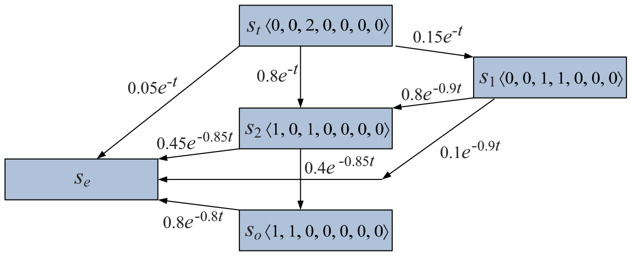


Fig. 8. An example of the state transition graph ($h = 2$, $n = 5$, $C = 2$, $\Lambda = 0.4$, and $\lambda = 0.05$).

First, we define the average delay of the homing phase as the average value of delays for each copy reaching the first home in the homing phase, denoted by $D^{(1)}$. Moreover, we define the average delay of the spreading phase as the average value of delays for each home in $H_{\odot}$ to receive a copy, denoted by $D^{(2)}$. The delay for the destination to fetch a copy from a message holder is defined as the delay of the fetching phase, and is denoted by $D^{(3)}$. Then, we have:

Lemma 7: Assume that the inter-meeting time between each node and each home follows the exponential distribution with the parameters Λ, and the average number of homes of each node is $\bar{h}$. Then, the average delays of the first two phases $D^{(1)}$, $D^{(2)}$, and the delay of the fetching phase $D^{(3)}$ satisfy:

$$D^{(1)} = \frac{1}{\bar{h}\Lambda}; \tag{5}$$

$$D^{(2)} \leq \frac{3h}{2\bar{h}\Lambda}; \tag{6}$$

$$D^{(3)} = \begin{cases} \frac{h}{\bar{h}C\Lambda + (h-\bar{h})C\lambda}, & C \leq h \\ \frac{1}{\bar{h}\Lambda + (C-\bar{h})\lambda}, & C > h \end{cases}. \tag{7}$$

Proof: First, we calculate the average delay $D^{(1)}$ for the homing phase. If the source does not meet other nodes in this phase, $D^{(1)}$ is the expected delay for the source to visit a home. Since the inter-meeting time between each node and each home follows the exponential distribution with the parameters Λ, and the average number of homes of each node is $\bar{h}$, $D^{(1)}$ is equal to $\frac{1}{\bar{h}\Lambda}$ in this case. Now, we consider the case where the source meets other nodes in the homing phase. Without loss of generality, we assume that a message holder that has c copies meets another node before it reaches a home. The message holder and the encountered node have h_1 and h_2 homes, respectively. Then, according to the proportional homing scheme, they will get $\frac{h_1 c}{h_1+h_2}$ and $\frac{h_2 c}{h_1+h_2}$ copies, respectively. The corresponding average delay is $\frac{1}{c} \cdot (\frac{h_1 c}{h_1+h_2} \cdot \frac{1}{h_1\Lambda} + \frac{h_2 c}{h_1+h_2} \cdot \frac{1}{h_2\Lambda}) = \frac{2}{(h_1+h_2)\Lambda}$, which is the average delay for the two nodes to visit a home. Thus, when the copies are split among the encountered nodes according to the proportional homing scheme, $D^{(1)}$ is still equal to the average delay for the nodes to visit a home, i.e., $D^{(1)}=\frac{1}{\bar{h}\Lambda}$.

Second, we derive the upper bound of $D^{(2)}$ for the spreading phase. We consider the worst case. That is, a home that has received $C-1$ message copies from the source spreads its extra copies to other homes according to the *1*-spreading scheme. In fact, this average delay includes two parts. The first part is the average delay for the home to spread its extra copies to its visiting nodes, denoted by $D_1^{(2)}$. In Lemma 3, we have derived the expected delays for a home in a homogeneous MSN to spread k message copies to $n-1$ nodes. Here, we only need to consider a home spreading its extra copies to $C-2$ of $\frac{\bar{h}n}{h}$ nodes, since the average number of homes of each node is $\bar{h}$. By using the same analysis in Lemma 3, we can get:

$$D_1^{(2)} \leq \frac{1}{C-2} \cdot \sum_{j=1}^{C-2} \frac{C-j-1}{\frac{\bar{h}n}{h}-j+1} \cdot \frac{1}{\Lambda} < \frac{hC}{\bar{h}n\Lambda} < \frac{h}{2\bar{h}\Lambda} \tag{8}$$

The second part is the average delay for each home in $H_{\odot}$ to receive copies from those nodes that hold message copies. More specifically, it is the average delay for $\frac{\bar{h}}{h}\cdot(C-1)$ nodes to send their copies to the homes in $H_{\odot}$. In fact, the delay for the first home in $H_{\odot}$ to receive a copy is the expected delay for a node visiting the home divided by $\frac{\bar{h}}{h}\cdot(C-1)$, i.e., $\frac{h}{\bar{h}}\cdot\frac{1}{(C-1)\Lambda}$. The delay for the i-th home in $H_{\odot}$ to receive a message copy is $\sum_{j=1}^{i}\frac{h}{\bar{h}}\cdot\frac{1}{(C-j)\Lambda}$ $(1\leq i<min\{h,C\})$. Let $k=min\{h,C\}$. Then, we have:

$$D_2^{(2)} \leq \frac{1}{k-1} \sum_{i=1}^{k-1} \frac{h}{\bar{h}} \cdot \frac{k-i}{C-i} \cdot \frac{1}{\Lambda} \leq \frac{h}{\bar{h}\Lambda} \tag{9}$$

By combining the average delay of the two parts, we have $D^{(2)}=D_1^{(2)}+D_2^{(2)}\leq\frac{3h}{2\bar{h}\Lambda}$.

Finally, we compute $D^{(3)}$. In the fetching phase, the destination will fetch the message from C homes if $C\leq h$, among which $\frac{\bar{h}}{h}C$ homes are the homes of the destination, on average. The corresponding expected delay is $\frac{h}{\bar{h}C\Lambda+(h-\bar{h})C\lambda}$. If $C>h$, the destination will fetch the message from one of its $\bar{h}$ homes, the other $h-\bar{h}$ homes, or the $C-h$ nodes that hold the copies. Then, the corresponding expected delay is $\frac{1}{\bar{h}\Lambda+(C-\bar{h})\lambda}$. By combining the results of the two cases, we can get that the theorem is correct. □

Note that the message delivery in HS might complete at each phase; in the worst case, it completes at the third phase. Thus, the sum of $D^{(1)}$, $D^{(2)}$, and $D^{(3)}$ is an upper bound for the expected delivery delay of HS. That is, we directly have the following theorem:

Theorem 8: The expected delivery delay of the HS algorithm, denoted by D, satisfies:

$$D \leq \begin{cases} \frac{1}{\bar{h}\Lambda}+\frac{3h}{2\bar{h}\Lambda}+\frac{h}{\bar{h}C\Lambda+(h-\bar{h})C\lambda}, & C\leq h \\ \frac{1}{\bar{h}\Lambda}+\frac{3h}{2\bar{h}\Lambda}+\frac{1}{\bar{h}\Lambda+(C-\bar{h})\lambda}, & C>h \end{cases} \tag{10}$$

Proof: This is a straightforward result of Lemma 7. □

Now we can, in turn, determine the number of message copies C. Given an arbitrary threshold Θ $(\geq\frac{1}{\bar{h}\Lambda}+\frac{3h}{2\bar{h}\Lambda})$ about the expected delivery delay of HS, we let C satisfy the following equation.

$$C=\begin{cases} \frac{h}{\bar{h}\Lambda+(h-\bar{h})\lambda}\cdot\frac{2\bar{h}\Lambda}{2\bar{h}\Lambda\Theta-2-3h}, & \Theta\geq\frac{1}{\bar{h}\Lambda}+\frac{3h}{2\bar{h}\Lambda}+\frac{1}{\bar{h}\Lambda+(h-\bar{h})\lambda} \\ \frac{\Lambda}{\lambda}\cdot(\frac{2\bar{h}}{2\bar{h}\Lambda\Theta-2-3h}-\bar{h})+\bar{h}, & \Theta<\frac{1}{\bar{h}\Lambda}+\frac{3h}{2\bar{h}\Lambda}+\frac{1}{\bar{h}\Lambda+(h-\bar{h})\lambda} \end{cases} \tag{11}$$

Then, according to Theorem 8, we can ensure that $D\leq\Theta$.

Here, we point out that the upper bound on the expected delivery delay of the (extended) HS algorithm in Eq. 10 is not a tight one. Despite this, the bound is enough, since our objective is to estimate the required number of message copies to ensure a given expected delivery delay performance for this algorithm. A little estimation error is negligible.

7 Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performance of HS. The algorithms in the comparison, evaluation methods, settings, and results are presented as follows.

7.1 Algorithms in Comparison

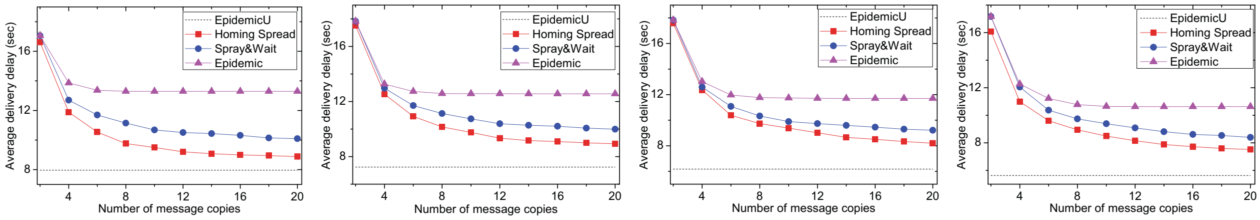
In this paper, we only focus on zero-knowledge multi-copy routing algorithms for MSNs. To make a fair performance comparison, we only compare the Homing Spread algorithm with the existing zero-knowledge routing algorithms: the Spray&Wait [14] algorithm and the Epidemic [13] algorithm with a given number of copies.

Both Spray&Wait and Epidemic deliver messages through replication. The message holder in Spray&Wait adopts the binary scheme to split the copies among itself and the encountered receivers. Note that there is no global view that can be used to control the number of message copies for the Epidemic [13] algorithm with a given number of copies. Thus, for simplicity, we just let the source in Epidemic spread message copies to each encountered node.

In addition, we also implement an Epidemic algorithm in which there is no limit to the number of copies, denoted by EpidemicU, since it can get the optimal expected delivery delay among all routing algorithms.

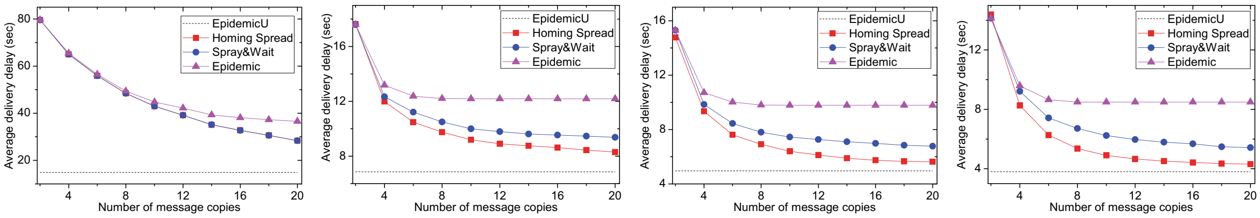
7.2 Simulation Settings and Metrics

Our simulations are conducted on synthetic traces that are generated by a Time-Variant Community Model (TVCM) [18]. This is because the commonly used real traces (such as Cambridge Haggle Trace and UMassDieselNet Trace) do not provide the needed community information. In contrast, the TVCM model



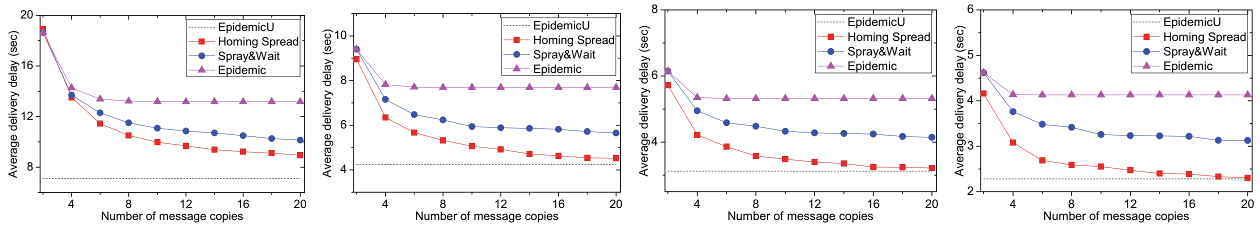
(a) Number of nodes: $n = 100$ (b) Number of nodes: $n = 200$ (c) Number of nodes: $n = 300$ (d) Number of nodes: $n = 400$

Fig. 9. Performance comparisons of average delivery delay vs. number of message copies ($h=5$, $\Lambda=0.04$).



(a) Number of homes: $h = 0$ (b) Number of homes: $h = 5$ (c) Number of homes: $h = 10$ (d) Number of homes: $h = 15$

Fig. 10. Performance comparisons of average delivery delay vs. number of message copies ($n=200$, $\Lambda=0.04$).



(a) Homing probability:Λ=0.04 (b) Homing probability:Λ=0.08 (c) Homing probability:Λ=0.12 (d) Homing probability:Λ=0.16

Fig. 11. Performance comparisons of average delivery delay vs. number of message copies ($n=200$, $h=5$).

TABLE 1
Evaluation Settings.

parameter name	range
deployment area	20×20
number of nodes n	100-400
number of homes h	0-15
homing probability per second Λ	0.04-0.16
number of messages	10,000
allowed message copies C	2-20

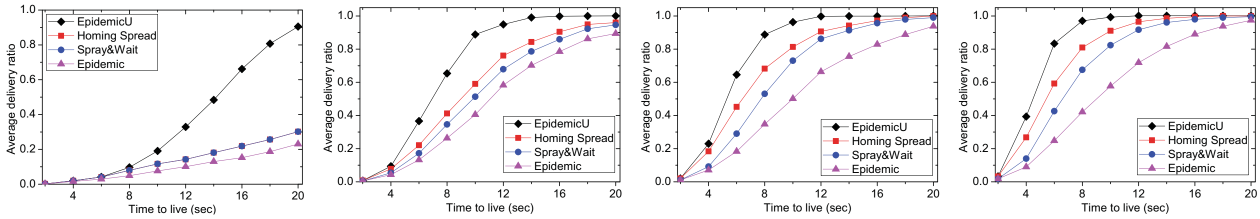
is a widely-adopted model derived from real MSNs. Moreover, we can modify the model parameters as needed, so that it can reproduce various empirical mobility properties, which are beneficial to the performance evaluation of our algorithm.

In the simulations, we deploy $n=100, 200, 300$, and 400 nodes in a grid, a square area composed of 20×20 small squares, each of which represents a location. Among the locations, there are $h = 0-15$ homes. Mobile nodes perform random waypoint trips inside and outside homes following the TVCM model [18]. The unit of time is seconds. In each second, the homing probability of each node, which is equal to Λ, is selected from $0.04-0.16$, while ensuring that the total homing probability does not exceed 1. Nodes can communicate with each other only when they visit the same small square. Each home is equipped with a throwbox [19]. In each evaluation, we randomly generate $10,000$ messages, whose sources and destinations are assigned randomly among the n nodes. Each message is assigned with a TTL (Time-To-Live), beyond which the corresponding message copies will be discarded. All of the evaluation variables are shown in Table 1.

The widely-adopted metrics are evaluated in our simulations, including the average delivery delay and average delivery ratio. The average delivery delay is the delivery time for the first message copy to reach its destination. The average delivery ratio is the ratio of successful deliveries to all message deliveries.

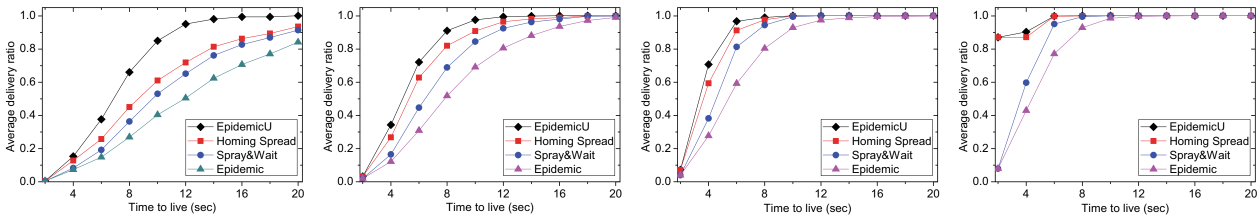
7.3 Evaluation in Homogeneous Settings

We conduct three groups of simulations to evaluate the performance of average delivery delay of the algorithms under the homogeneous setting. In the first group of simulations, we change the number of nodes from 100 to 400, while setting $h = 5$, $\Lambda = 0.04$, and $C = 10$. Then, we vary the number of homes from 0 to 15, while setting $n = 200$, $\Lambda = 0.04$, and $C = 10$, in the second group of simulations. Finally, we modify the homing probability of each node in the third group of simulations. In all of the simulations, we record the average delivery delays of Homing Spread,



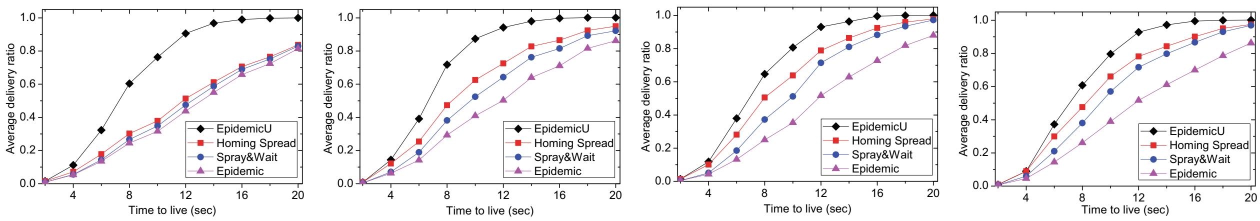
(a) Number of homes: $h = 0$ (b) Number of homes: $h = 5$ (c) Number of homes: $h = 10$ (d) Number of homes: $h = 15$

Fig. 12. Performance comparisons of average delivery ratio vs. time-to-live ($n = 200$, $\Lambda = 0.04$, $C = 10$).



(a) Homing probability:Λ=0.04 (b) Homing probability:Λ=0.08 (c) Homing probability:Λ=0.12 (d) Homing probability:Λ=0.16

Fig. 13. Performance comparisons of average delivery ratio vs. time-to-live ($n = 200$, $h = 5$, $C = 10$).



(a) Message copies: $C = 5$ (b) Message copies: $C = 10$ (c) Message copies: $C = 15$ (d) Message copies: $C = 20$

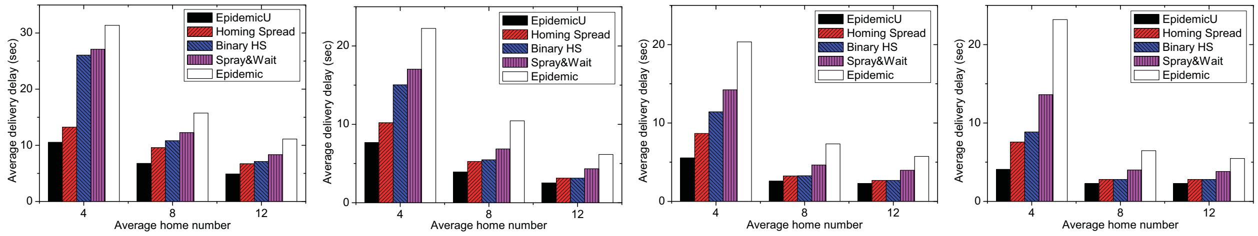
Fig. 14. Performance comparisons of average delivery ratio vs. time-to-live ($n = 200$, $h = 5$, $\Lambda = 0.04$).

Spray&Wait, and Epidemic, when given a different number of copies, as shown in Figs. 9-11. Moreover, as the minimum average delivery delay that can be achieved by all possible routing algorithms, we record the average delivery delay of EpidemicU and plot it as a lower bound in these figures.

More specifically, the results in Figs. 9-11 show that the average delivery delays of the three algorithms reduce when there is an increase in the number of copies. In contrast, Epidemic, in which only the source spreads the copies in the network, has the worst delivery delay. Spray&Wait, in which multiple nodes and homes help to spread the copies in the network, has a medium performance. Homing Spread, which mainly lets homes, assisted by nodes, spread the copies in the network, has the best performance among the three algorithms. The results also prove that homes play an important role in the message spreading process. When the number of homes increases, or the homing probability increases, the average delivery delay of Homing Spread reduces significantly, while the average delivery delay of Spray&Wait decreases moderately. At the same time, the average delivery delay of Epidemic reduces slightly, as shown in Figs. 10 and 11, respectively. When the number of homes is zero, Homing Spread is degraded to Spray&Wait, as shown in Fig. 10(a), where the curves of the two algorithms overlap. Moreover, when the number of homes or the homing probability is sufficiently large (e.g., $h = 15$ or $\Lambda = 0.12, 0.16$), Homing Spread can achieve nearly the same performance on average delivery delay as EpidemicU, i.e., the best result of all possible algorithms, as shown in Fig. 10(d), Fig. 16(c), and Fig. 16(d).

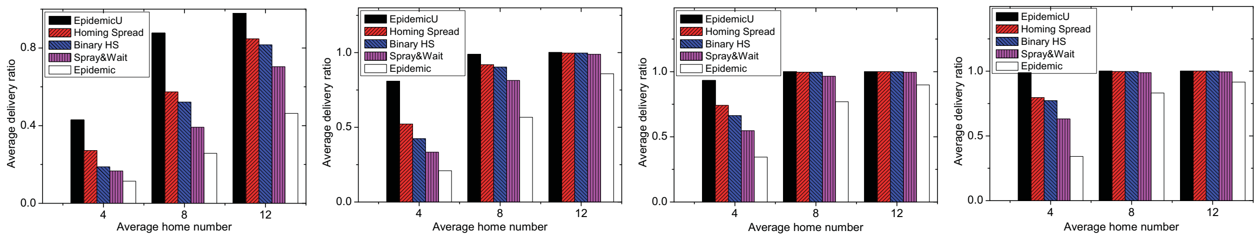
Next, we also conduct three groups of simulations to evaluate the performance of the above algorithms on the delivery ratio. We vary the number of homes, the homing probability, and the number of copies, while fixing other variables, respectively. In each simulation, we calculate the average delivery ratios of the four algorithms when given different values of time-to-live for each message, beyond which the message will be discarded, as shown in Figs.12-14.

The results in Figs.12-14 show that Homing Spread can successfully deliver the messages more quickly, and can achieve an average delivery ratio that is much higher than those of Epidemic and Spray&Wait. The results also show that homes greatly affect the performance of message deliveries. When the number of homes or the homing probability increases, the average delivery ratio of Homing Spread reduces significantly, as shown in Figs. 10 and 11. In contrast, the average delivery ratio of Spray&Wait reduces moderately. However, the average delivery ratio of Epidemic reduces by a little. This is because Epidemic



(a) Homing probability:Λ=0.04 (b) Homing probability:Λ=0.08 (c) Homing probability:Λ=0.12 (d) Homing probability:Λ=0.16

Fig. 15. Performance comparisons of average delivery delay vs. average home number ($n=200$, $C=10$).



(a) Homing probability:Λ=0.04 (b) Homing probability:Λ=0.08 (c) Homing probability:Λ=0.12 (d) Homing probability:Λ=0.16

Fig. 16. Performance comparisons of average delivery ratio vs. average home number (n=200, C=10, TTL=10).

only depends on the source to spread copies in the network. The increased number of homes, and the homing probability, cannot contribute to this message spreading scheme. Moreover, when the homing probability is large enough (e.g., $\Lambda = 0.12, 0.16$), Homing Spread can achieve nearly the same performance on average delivery ratio as EpidemicU, as shown in Figs. 13(c) and 13(d). When the number of homes is zero, Homing Spread is degraded to Spray&Wait, as shown in Fig. 12(a). In addition, Fig. 14 shows that when the number of copies increases, the average delivery ratios of Homing Spread and Spray&Wait will increase significantly. However, when the number of copies goes beyond a moderate value (e.g., 3 times the number of homes in Fig 14(c)), their average delivery ratios increase slightly. In contrast, Epidemic is barely affected by the number of copies. This is still due to the reason that only the source in this algorithm spreads the copies. If there is no time to encounter other nodes, the source just keeps the extra copies to itself, which is not beneficial to the improvement of the delivery ratio.

7.4 Evaluation in Heterogeneous Settings

We also evaluate the performance of the (extended) Homing Spread algorithm in the heterogeneous setting by comparing it with Spray&Wait and Epidemic. In this setting, we first determine an average home number $\bar{h}$ for all nodes. Then, we let each node randomly select an integer from $[\bar{h}-2, \bar{h}+2]$ as its number of homes. Here, the number of homes is determined according to our experience from the study on the real MSN trace. Other parameters can achieve a similar result. In addition, in order to demonstrate that Homing Spread using the proportional homing scheme in the heterogeneous setting outperforms that using the binary homing scheme, we also realize a Homing Spread algorithm that adopts the binary homing scheme in the heterogeneous setting, denoted by *Binary HS*, in the simulations. Moreover, we evaluate the average delivery delay and the average delivery ratio by conducting all simulations like that in the homogeneous setting. Due to space limitations, we only provide two groups of evaluation results here.

First, we change the parameter $\bar{h}$ from 4 to 12, set $n = 200$, $\Lambda = 0.04 - 0.16$, $C = 10$, and then, record the average delay of all message deliveries. Second, we evaluate the average delivery ratio by setting the Time-To-Live of each message $TTL = 10$, beyond which the message copy will be discarded. The results are shown in Figs. 15 and 16, from which we can get that Homing Spread has a smaller average delivery delay and a larger average delivery ratio than other algorithms, including Binary HS. When the average home number increases, the average delivery delay of Homing Spread will decrease, and the average delivery ratio will increase; these come close to the best results.

8 Related Work

Many routing algorithms have been proposed for MSNs. Most of them are probability-based algorithms (e.g., [2]–[6]) or social-aware algorithms (e.g., [7]–[12]). These algorithms assume that the contact probability between nodes changes very slowly along with time. Then, the historical contact records between nodes are collected and used to guide the message delivery. Compared to existing works, HS does not require any historical information. Among the existing MSN routing algorithms, only two typical algorithms, Epidemic [13] and Spray&Wait [14], are similar to

HS, which is zero-knowledge-based. However, neither distinguishes homes from other locations, as all locations are considered to be the same.

HS assumes that each home has a virtual throwbox. In contrast, the existing works on throwboxes mainly focus on the capacity and delivery delay of the Epidemic algorithm when adding throwboxes into MSNs [19], [21]. Moreover, these throwboxes are randomly placed, and they are usually physical storage devices. In addition, some other works also use the stationary relays to improve the routing performance, such as [22]. The network model and routing scheme are different from this paper. To the best of our knowledge, this is the first zero-knowledge MSN routing algorithm that takes the social characteristic of MSNs into consideration.

9 Conclusion

In this paper, we study a special type of mobile social network, where the routing space includes some frequently visited homes, and propose a zero-knowledge multi-copy routing algorithm called Homing Spread (HS). HS utilizes the home feature and sets a higher priority for homes to help spread messages quickly. Theoretical analysis and simulation results show that homes play an important role in the message spreading process. By using the notion of home, HS achieves a better performance than existing zero-knowledge MSN routing algorithms.

References

[1] J. Wu, M. Xiao, and L. Huang, "Homing spread: Community home-based multi-copy routing in mobile social networks," in *IEEE INFOCOM*, 2013.

[2] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *ACM SIGCOMM*, 2007.

[3] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: routing for vehicle-based disruption-tolerant networks," in *IEEE INFOCOM*, 2006.

[4] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility," in *IEEE PerCom*, 2007.

[5] X. Tie, A. Venkataramani, and A. Balasubramanian, "R3: robust replication routing in wireless networks with diverse connectivity characteristics," in *ACM SIGCOMM*, 2011.

[6] X. Chen, J. Shen, T. Groves, and J. Wu, "Probability delegation forwarding in delay tolerant networks," in *IEEE ICCCN*, 2009.

[7] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *ACM MobiHoc*, 2008.

[8] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *ACM MobiHoc*, 2007.

[9] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," in *ACM MobiHoc*, 2009.

[10] M. Xiao, J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *to appear in IEEE Transactions on Computers*, p. Digital Object Indentifier: 10.1109/TC.2013.55, 2013.

[11] T. Ning, Z. Yang, H. Wu, and Z. Han, "Self-interest-driven incentives for ad dissemination in autonomous mobile social networks," in *IEEE INFOCOM*, 2013.

[12] L. Guo, C. Zhang, H. Yue, and Y. Fang, "A privacy-preserving social-assisted mobile content dissemination scheme in dtns," in *IEEE INFOCOM*, 2013.

[13] A. Vahdate and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University, Tech. Rep. CS-2000-06, June 2000.

[14] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Efficient routing in intermittently connected mobile networks: The multiple-copy case," *IEEE/ACM Transactions on Networking*, vol. 16, no. 1, pp. 77–90, 2008.

[15] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *ACM MobiCom*, 2004.

[16] L. Jeremie, F. Timur, and C. Vania, "Evaluating mobility pattern space routing for dtns," in *IEEE INFOCOM*, 2006.

[17] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *ACM MobiHoc*, 2006.

[18] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *IEEE INFOCOM*, 2007.

[19] M. Ibrahim, P. Nain, and I. Carreras, "Analysis of relay protocols for throwbox-equipped dtns," in *WiOPT*, 2009.

[20] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, "Relays, base stations, and meshes: Enhancing mobile networks with infrastructure," in *ACM MobiCom*, 2008.

[21] B. Gu, X. Hong, P. Wang, and R. Borie, "Latency analysis for thrown box based message dissemination," in *IEEE Globecom*, 2010.

[22] S. Shahbazi, S. Karunasekera, and A. Harwood, "Improving performance in delay/disruption tolerant networks through passive relay points," *Wireless Networks*, vol. 18, no. 1, pp. 9–31, 2012.

Mingjun Xiao is an associate professor in the School of Computer Science and Technology at the University of Science and Technology of China (USTC). He received his Ph.D. degree from USTC in 2004. In 2012, he was a visiting scholar at Temple University, under the supervision of Dr. Jie Wu. He has served as a reviewer for many journal papers. His main research interests include delay tolerant networks and mobile social networks.

Jie Wu is the chair and a Laura H. Carnell Professor in the Department of Computer and Information Sciences at Temple University. Prior to joining Temple University, he was a program director at the National Science Foundation and Distinguished Professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Computers, IEEE Transactions on Service Computing, and Journal of Parallel and Distributed Computing. Dr. Wu was general co-chair/chair for IEEE MASS 2006 and IEEE IPDPS 2008 and program co-chair for IEEE INFOCOM 2011. Currently, he is serving as general chair for IEEE ICDCS 2013 and ACM MobiHoc 2014, and as program chair for CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

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