Homing Spread: Community Home-based Multi-copy Routing in Mobile Social Networks

Jie Wu[†], Mingjun Xiao*[†‡], and Liusheng Huang[‡]
[†]Department of Computer Information and Sciences, Temple University, Philadelphia, PA 19122, USA
[‡]School of Computer Science and Technology / Suzhou Institute for Advanced Study,
University of Science and Technology of China, Hefei, P. R. China
*Correspondence to: xiaomj@ustc.edu.cn

***Abstract*—A mobile social network (MSN) is a special delay tolerant network (DTN) composed of mobile nodes with social characteristics. Mobile nodes in MSNs generally visit community homes frequently, while other locations are visited less frequently. We propose a novel zero-knowledge MSN routing algorithm, *homing spread* (HS). The community homes have a higher priority to spread messages into the network. Theoretical analysis shows that the proposed algorithm can spread a given number of message copies in an optimal way when the inter-meeting times between any two nodes and between a node and a community home follow exponential distributions. We also calculate the expected delivery delay of HS. In addition, extensive simulations are conducted. Results show that community homes are important factors in efficient 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 home, mobile social networks (MSNs), routing.**

I. Introduction

As more users use portable devices to contact each other, mobile social networks (MSNs) attract more attention. As a special type of delay tolerant network (DTN), MSNs experience intermittent connectivity, and even long-lasting disconnections, due to the mobility of the nodes. There are generally no stable end-to-end delivery paths in an MSN. Therefore, delivering messages becomes 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 be simply divided into two categories.

One category is *knowledge-based routing algorithms*, which mainly includes probability-based algorithms (e.g., [1], [2]–[4]) and social-aware algorithms (e.g., [5]–[7]). These algorithms record the historical contact information or social characteristics of nodes, and then this knowledge is used to guide message deliveries. However, these algorithms take time and storage space to collect knowledge on historical contacts.

Another category consists of *zero-knowledge routing algorithms*, which do not require any prior knowledge regarding the contact information among nodes. The typical algorithms include Epidemic [8] and Spray&Wait [9]. Epidemic is based on the flooding strategy, which incurs a significant number of message copies. In this paper, we consider a type of Epidemic in which the number of copies is limited. 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. This algorithm assumes that all nodes just randomly walk in a given area and nodes visit all locations in a uniformly random way. However, the mobility of nodes in MSNs generally follows some social characteristics, making Spray&Wait less efficient, as we will show later in this paper.

We consider an MSN in which nodes visit some locations, called *community homes* or simply *homes*, frequently, while the other locations are visited less frequently. Many mobility models [10]–[14] capture this characteristic of skewed location visiting preferences from several real MSN traces. Besides, we assume that each home supports a *virtual* throwbox [15], a mechanism that can store a message at a local storage device, or at another node currently at the same home. A *message holder* is either a mobile node or a home that has message copies. The objective is to send a message from a mobile source to a destination quickly, using a given number of copies.

We propose a zero-knowledge multi-copy routing algorithm, *homing spread* (HS). HS consists of three phases. In the first phase, the source spreads copies quickly to homes. In the second phase, homes that have received copies 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. HS 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 show that HS is optimal when the inter-meeting times between any two nodes and between a node and a community home follow exponential distributions.
2) We construct a continuous Markov chain to calculate the expected delivery delay of HS and derive an upper bound. Moreover, we calculate the required number of copies needed to bound the expected delivery delay to a given threshold.
3) We conduct extensive simulations on a synthetic MSN trace to evaluate HS. The results show that HS significantly outperforms several existing zero-knowledge multi-copy routing algorithms, including Epidemic with a given number of copies, and Spray&Wait.

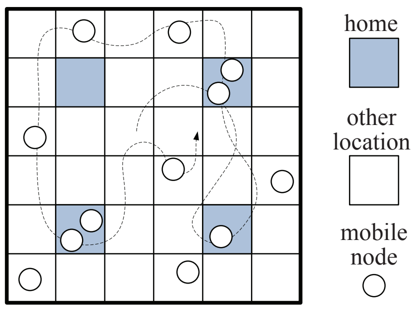


Fig. 1. The network model.

The remainder of the paper is organized as follows. We introduce the network model and problem in Section II. HS and its performance analysis are presented in Sections III and IV, respectively. In Section V, we evaluate the performance of HS through extensive simulations. After reviewing the related work in Section VI, we conclude the paper in Section VII. The proofs of the major theorems are presented in the Appendix.

II. Network Model & Problem

We consider a typical MSN that is composed of a number of mobile nodes and some locations. Each node visits a few frequent locations, called community homes or homes, while the other locations are visited less frequently. Each node has multiple homes. Consider students (i.e., mobile nodes in an MSN) in a university environment; the community homes are the dormitories, cafeteria, classrooms, and laboratories. At a more refined level (or a heterogeneous setting), we can perhaps add Chinatown for Chinese students, and so on. We focus first on a homogeneous setting and later discuss our results in a heterogeneous setting.

More specifically, 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 homes $H=\{n+1,\cdots,n+h\}$ and $m-h$ other locations $L=\{n+h+1,\cdots,n+m\}$, as shown in Fig. 1. Moreover, each node visits either a home with a relatively high probability or another location with the remaining probability. The visited home and other location are randomly selected from H and L, respectively. A visit to a home is known as *homing*, but when a message holder meets with another node at a different location, it is known as *roaming*.

Given a fixed number of message copies C, 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?

III. Homing Spread (HS)

In this section, we provide the details of HS with a given number of message copies. Since each node has a relatively high probability of visiting homes, we treat these homes in a special way. Once a home receives a copy, HS maintains the corresponding home as a static message holder. We focus on one message 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. HS includes three phases: *homing*, *spreading*, and *fetching*.

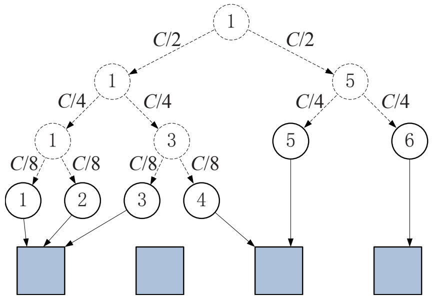


Fig. 2. The binary homing scheme.

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 to the home. When roaming occurs (i.e., a message holder meets another node at another location), copies are equally split between the two nodes and both become message holders.
2) In the spreading phase, homes with multiple copies spread them to other homes and mobile nodes. The home gives one copy to each node located at the same home, subject to the availability of the copies. However, the last copy is kept at the home through a virtual throwbox. Each new message holder with one copy starts its homing phase.
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.

Note that local storage is not essentially for a virtual throwbox if there is a node at the same home. We will show that when the inter-meeting times between any two nodes and between a node and a community home follow exponential distributions, HS is optimal in terms of minimizing the *expected delivery delay* to the destination. Unless otherwise specified in the rest of the discussion, "other nodes" refers to mobile nodes outside of the homes.

A. 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 1: (*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.

Algorithm 1 The Homing Spread (HS) algorithm

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 r_i and r_j message copies **then**
6:       node i holds $\lceil r_i/2 \rceil + \lfloor r_j/2 \rfloor$ copies through exchange with node j;
7:   **if** node i visits a home h **then**
8:     node i sends all its copies to h;
9:     **if** $h \in H_+$ or i is the destination **then**
10:       h sends a copy to node i.

Fig. 2 shows an example of the binary homing scheme. Message copies are binary split until they reach the homes.

B. 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_+, H_1 and H_0 ($H = H_+ + H_1 + H_0$) 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 2: (*1-Spreading Scheme*): Each home $h_i \in H_+$ spreads a copy to each node in the same home until only one copy remains, so that $h_i \in H_1$ after the spreading. If such a node with one copy later visits another home $h_j \in H_0$, the node sends the copy to that home, so that $h_j \in H_1$ after the visit.

Using the *1*-spreading scheme, as shown in Fig. 3, each home has at most one copy. If $C > h$, there are $C - h$ nodes outside the homes that have a copy.

C. The Fetching Phase

After the spreading phase, there would be C message holders, including h homes and $C{-}h$ other nodes, or C homes if $C \leq h$. Then, in the fetching phase, the destination fetches this message once it encounters a message holder.

D. 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. 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).

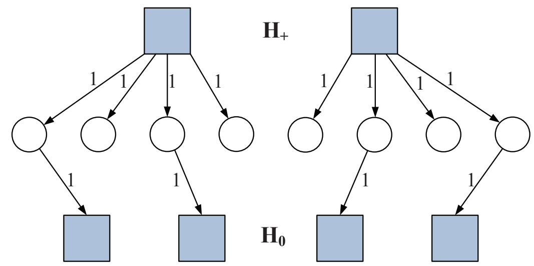


Fig. 3. The *1*-spreading scheme.

E. Optimality of HS

When we analyze the delay performance of HS, we assume that the inter-meeting times between any two nodes and between a node and a community home follow exponential distributions with parameters λ and Λ ($\Lambda \gg \lambda$), respectively. Such an assumption is widely adopted (e.g., [16]). Moreover, we define a concept of network *state*, which is used to describe the distribution of message copies in the whole network.

Definition 3: (*State* of Network s): s is a vector with $h{+}n$ components, i.e., $s = \langle s_1, s_2, \cdots, s_h, s_{h+1}, \cdots, s_{h+n} \rangle$, 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 $i > h$).

Based on Definition 3, there is an optimal state, denoted by s_o, which can make the destination fetch a message copy the quickest. According to the assumptions about the network model, the destination has a higher probability of visiting a home than that of meeting another node. Thus, the optimal state is that each home holds a message copy and other $C{-}h$ nodes each hold a copy if $C > h$, or arbitrary C homes each hold a copy if $C{<}h$. For example, $s_o = \langle 1, 1, \cdots, 1, 0, 0 \cdots \rangle$.

Now, we consider the homing phase. Since the binary homing scheme is the same as the binary spraying scheme in Spray&Wait [16], we directly have 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 terms of the spreading phase, which differs from Spray&Wait, we also can derive its optimality. Note that each node has a higher 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 the quickest. Moreover, more nodes might receive these copies, which ensures that each home in H_0 will receive a copy the quickest. Thus, we have:

Lemma 2: The *1*-spreading scheme can spread message copies from a home in H_+ to the maximum number of nodes with the fastest speed.

Based on Lemmas 1 and 2, we get that HS is optimal.

Theorem 3: (*Optimality of HS*): HS can achieve the minimum expected delay when the inter-meeting times between any two nodes and between a node and a community home follow independent and identical exponential distributions, respectively.

IV. Performance Analysis

In this section, we formally analyze the expected delivery delay of HS. First, we use the continuous Markov chain to compute the expected delivery delay. Since it is hard to derive the close formula, we derive a upper bound, whereby we determine the number of message copies.

A. Computing the Expected Delivery Delay

We adopt a continuous Markov chain to compute the expected delivery delay of HS. First, we determine all possible states in the *state transition graph*. According to Definition 3, a state $s=\{s_1,\cdots,s_h,\cdots,s_{h+n}\}$ satisfies:

$$\sum_{i=1}^{h+n} s_i = C. \tag{1}$$

Let S denote the state space. Then, S is the solution space of Eq. 1. The start state is $s_t=\langle 0,\cdots,0,C,0,\cdots,0\rangle$, where C is the $(h+1)$-th component that corresponds to the source.

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 will encounter each other. Then, checking the values of s_i,s_j,s'_i,s'_j, we can determine whether they follow the binary splitting rule. If their values do not follow the rule, 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 one of i,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 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.

Algorithm 2 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**: $\int_0^\infty tf_{s_t,s_e}(t)dt$;

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 4: 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:

$$f_{s,s_e}(t)=\sum_{s'\in N_s}\int_0^t\rho_{s,s'}(x)f_{s',s_e}(t-x)dx. \tag{2}$$

This theorem shows that if the cumulative probability density functions for the state transitions from the next states of s to s_e have been calculated, then the cumulative probability density function of the state s also can 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 $\int_0^\infty tf_{s_t,s_e}(t)dt$.

Based on the above method, we present Algorithm 2 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 would 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

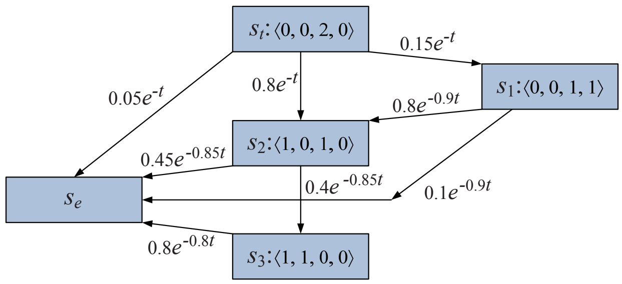


Fig. 4. An example of the state transition graph ($h = 2$, $n = 5$, $C = 2$, $\Lambda = 0.4$, and $\lambda = 0.05$).

overhead of Algorithm 2 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 2 for simplicity. In fact, these cumulative probability density functions can be realized in the real implementation, since they are composed with exponential functions that can be described by pairwise coefficients and exponents.

B. An Example to Compute the Expected Delivery Delay

Consider a simple example, in which $h = 2$, $n = 5$, $C = 2$, $\Lambda = 0.4$, and $\lambda = 0.05$. We can use Algorithm 2 to calculate the expected delivery delay. First, the state graph is constructed, as shown in Fig. 4, in which invalid states are deleted, the equivalent states are combined into one state, and the $5 - 7$ components of each state are ignored due to their zero values. Besides the end state s_e, the first two components of each state 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_2 is an intermediate state where a home and a node each hold a copy, respectively.

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. Thus, $\rho_{s_t,s_2}(t) = 2\Lambda e^{-2\Lambda t - 4\lambda t} = 0.8e^{-t}$.

After the state graph construction, Algorithm 2 uses the backward derivation from state s_e to compute the cumulative probability density functions. First, the cumulative probability density function of s_3 is determined, i.e., $f_{s_3,s_e}(t) = \rho_{s_3,s_e}(t) = 0.8e^{-0.8t}$. Next, $f_{s_2,s_e}(t)$ is determined, i.e., $f_{s_2,s_e}(t) = \rho_{s_2,s_e}(t) + \int_0^t \rho_{s_2,s_3}(x) f_{s_3,s_e}(t - x)dx$, and so on. Finally, $f_{s_t,s_e}(t)$ is derived. Then, we can deduce 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.

C. The Upper Bound of the Expected Delivery Delay

Although we can calculate the expected delivery delay through Algorithm 2, it is hard to derive a close formula. Here, we derive an upper bound of the expected delivery delay.

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_0 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:

Theorem 5: The average delays of the first two phases $D^{(1)}$, $D^{(2)}$, and the delay of the fetching phase $D^{(3)}$ satisfy:

$$D^{(1)} = \frac{1}{h\Lambda}; \quad D^{(2)} \leq \frac{2}{\Lambda}; \tag{3}$$

$$D^{(3)} = \begin{cases} \frac{1}{C\Lambda}, & C \leq h; \\ \frac{1}{h\Lambda + (C-h)\lambda}, & C > h. \end{cases} \tag{4}$$

Note that the message delivery in HS might complete at each phase; in the worst case, it would complete 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 have:

Corollary 6: The expected delivery delay of the HS algorithm, denoted by D, satisfies:

$$D \leq \begin{cases} \frac{1}{h\Lambda} + \frac{2}{\Lambda} + \frac{1}{C\Lambda}, & C \leq h; \\ \frac{1}{h\Lambda} + \frac{2}{\Lambda} + \frac{1}{h\Lambda + (C-h)\lambda}, & C > h. \end{cases} \tag{5}$$

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

$$C = \begin{cases} \frac{1}{\Theta\Lambda - 2 - 1/h}, & \Theta \geq \frac{2}{h\Lambda} + \frac{2}{\Lambda}; \\ \frac{\Lambda}{\lambda} \cdot \left(\frac{1}{\Theta\Lambda - 2 - 1/h} - h\right) + h, & \Theta < \frac{2}{h\Lambda} + \frac{2}{\Lambda}. \end{cases} \tag{6}$$

Then, according to Theorem 6, we can ensure that $D \leq \Theta$.

D. Discussion

Here, we discuss the performance of HS in the heterogeneous setting, where each node only visits a part of the homes frequently. In this setting, HS still works depending on the subset overlap of homes.

For simplicity, we assume that each node only has h' ($0 < h' < h$) homes, which are random but uniformly selected from the home set H. The homes of each node might be different, but all of them will form the overlapped home set H. Obviously, this heterogeneous visit model will lead to a degradation in the performance of HS. Let $D^{\langle 1 \rangle}$, $D^{\langle 2 \rangle}$, and $D^{\langle 3 \rangle}$ denote the delay of three phases of HS in this setting. Then, a similar analysis shows that they satisfy:

$$D^{\langle 1 \rangle} = \frac{1}{h'\Lambda}; \quad D^{\langle 2 \rangle} \leq \frac{1}{\Lambda} + \frac{1}{\lambda};$$

$$D^{\langle 3 \rangle} \leq \begin{cases} \frac{h}{h'C\Lambda + (h-h')C\lambda}, & C \leq h; \\ \frac{1}{h'\Lambda + (C-h')\lambda}, & C > h. \end{cases}$$

Despite a performance degradation, HS is still an effective zero-knowledge method for spreading the copies throughout the network, even if each node does not record their homes. In addition, the accurate expected delivery delay of HS in this setting can still be derived through Algorithm 2. Our future work will focus on a special heterogenous setting, where nodes are aware of which homes they belong to. In addition, the home set of the destination can be piggybacked in the message.

TABLE I
EVALUATION SETTINGS.

parameter name	default	range
deployment area	20×20	-
number of nodes n	200	100-400
number of homes h	5	0-15
homing probability in each second Λ	0.04	0.04-0.16
number of messages	10,000	-
allowed message copies C	10	2-20

V. PERFORMANCE EVALUATION

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

A. 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 several existing zero-knowledge routing algorithms. More specifically, besides Homing Spread, we implement the Spray&Wait [9] algorithm and the Epidemic [8] 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. In contrast, the message holder in Epidemic just sends one message copy to each encountered node.

In addition, we also implement an Epidemic algorithm where 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.

B. Simulation Settings and Metrics

Our simulations are conducted on synthetic traces that are generated by a Time-Variant Community Model (TVCM) [14]. 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 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 is 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 $h = 0, 5, 15$, and 20 homes. Mobile nodes perform random waypoint trips inside and outside homes. 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 when they visit the same small square. Each home is equipped with a virtual throwbox [15] for virtual storage. In each evaluation, we randomly generate $10,000$ messages, whose sources and destinations are assigned uniformly random among the n nodes. All of the evaluation variables are shown in Table I.

The widely adopted metrics are evaluated in our simulations, including the average delivery delay and average delivery ratio. The average delivery delay is the average time of all message deliveries. The average delivery ratio is the ratio of successful deliveries to all message deliveries.

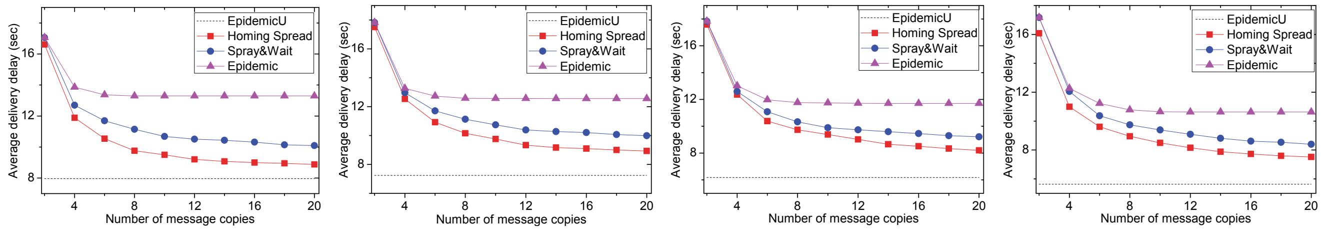
C. Evaluation Results

We conduct three groups of simulations to evaluate the performance of the algorithms on the average delivery delay. In the first group of simulations, we change the number of nodes, while keeping other variables fixed. Then, we vary the number of homes in the second group. Finally, we modify the homing probability of each node in the third group. In all of the simulations, we record the average delivery delays of Homing Spread, Spray&Wait, and Epidemic, when given a different number of copies, as shown in Figs. 5-7. 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.

More specifically, the results in Figs. 5-7 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, and the average delivery delay of Epidemic reduces slightly, as shown in Figs. 6 and 7, respectively. When the number of homes is zero, Homing Spread is degraded to Spray&Wait, as shown in Fig. 6(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, as shown in Fig. 6(d), Fig. 7(c), and Fig. 7(d).

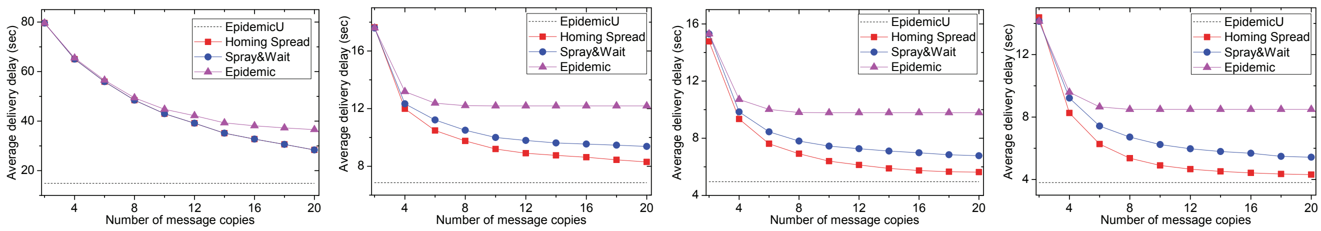
Next, we also conduct three groups of simulations to evaluate the performance of the above algorithms on the delivery ratio. We vary the number of homes, the homing probability, and the number of copies, while fixing other variables, respectively. In each simulation, we calculate the average delivery 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. 8-10.

The results in Figs. 8-10 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



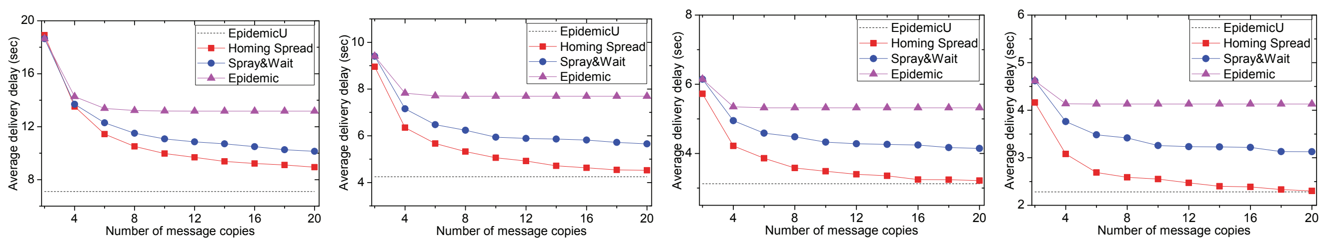
(a) Number of nodes: $n = 100$ (b) Number of nodes: $n = 200$ (c) Number of nodes: $n = 300$ (d) Number of nodes: $n = 400$

Fig. 5. 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. 6. Performance comparisons of average delivery delay vs. number of message copies ($n{=}200$, $\Lambda{=}0.04$).



(a) Homing probability: $\Lambda{=}0.04$ (b) Homing probability: $\Lambda{=}0.08$ (c) Homing probability: $\Lambda{=}0.12$ (d) Homing probability: $\Lambda{=}0.16$

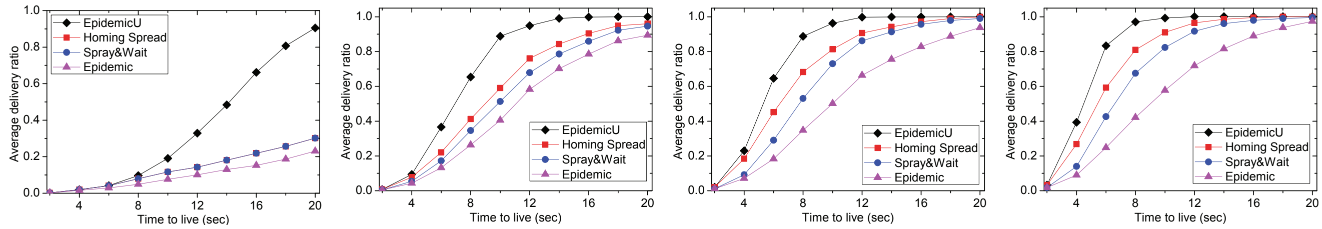
Fig. 7. Performance comparisons of average delivery delay vs. number of message copies ($n{=}200$, $h{=}5$).

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. 8 and 9. In contrast, the average delivery ratio of Spray&Wait reduces moderately. However, the average delivery ratio of Epidemic reduces by only a little. This is because Epidemic 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. 9(c) and 9(d). When the number of homes is zero, Homing Spread is degraded to Spray&Wait, as shown in Fig. 8(a). In addition, Fig. 10 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 of the number of homes in Fig. 10(c)), their average delivery ratios increase slightly. In contrast, Epidemic is barely affected by the number of copies. This is still due to the fact 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.

VI. Related Work

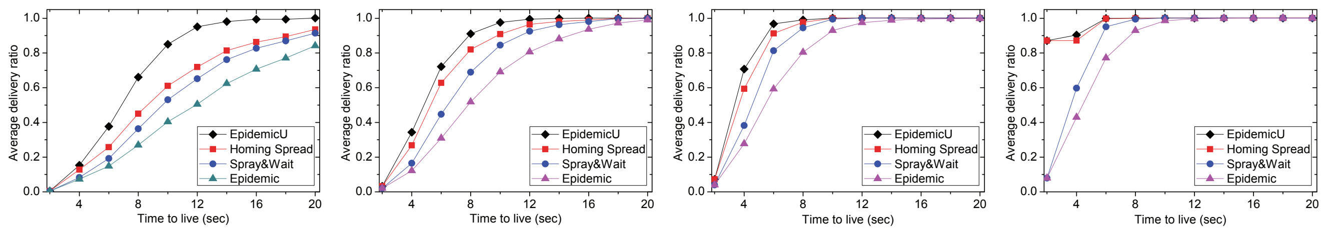
Many routing algorithms have been proposed for MSNs. Most of them are probability-based algorithms (e.g., [1]–[4], [17], [18]) or social-aware algorithms (e.g., [5]–[7]). These algorithms assume that the contact probability between nodes changes very slowly along with time. Then, the historical contact 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 [8] and Spray&Wait [9], 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 [15], [19]. Moreover, these throwboxes are randomly placed, and usually they are physical storage devices. In addition, some other works also use the stationary relays to improve the routing performance, such as [20]. 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



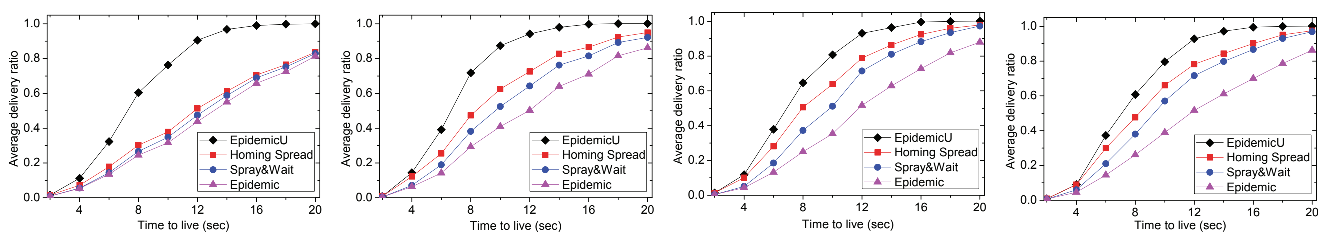
(a) Number of homes: $h=0$ (b) Number of homes: $h=5$ (c) Number of homes: $h=10$ (d) Number of homes: $h=15$

Fig. 8. Performance comparisons of average delivery ratio vs. time-to-live ($n\!=\!200$, $\Lambda\!=\!0.04$, $C\!=\!10$).



(a) Homing probability: $\Lambda\!=\!0.04$ (b) Homing probability: $\Lambda\!=\!0.08$ (c) Homing probability: $\Lambda\!=\!0.12$ (d) Homing probability: $\Lambda\!=\!0.16$

Fig. 9. 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. 10. Performance comparisons of average delivery ratio vs. time-to-live ($n\!=\!200$, $h\!=\!5$, $\Lambda\!=\!0.04$).

the social characteristic of MSNs into consideration.

VII. Conclusion

In this paper, we study a special type of mobile social network, where the routing space includes some frequently visited homes, and we 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 spread messages quickly. Theoretical analysis shows that HS can spread a given number of message copies in an optimal way when the inter-meeting times between any two nodes and between a node and a home follow exponential distributions. Simulation results also 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. Our future work will focus on an in-depth study of a heterogenous setting, where nodes have different, but overlapping subsets of homes.

Acknowledgment

This research was supported in part by NSF grants ECCS 1231461, ECCS 1128209, CNS 1138963, CNS 1065444, and CCF 1028167; 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 NSF of China (Grant No. 60803009, 61003044), the NSF of Jiangsu Province in China (Grant No. BK2009150); and the Fundamental Research Funds for the Central Universities.

References

[1] A. Balasubramanian, B. N. Levine, and A. Venkataramani, "Dtn routing as a resource allocation problem," in *ACM SIGCOMM*, 2007.
[2] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "Maxprop: routing for vehicle-based disruption-tolerant networks," in *IEEE INFOCOM*, 2006.
[3] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility," in *IEEE PerCom*, 2007.
[4] X. Tie, A. Venkataramani, and A. Balasubramanian, "R3: robust replication routing in wireless networks with diverse connectivity characteristics," in *ACM SIGCOMM*, 2011.
[5] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in *ACM MobiHoc*, 2008.
[6] E. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *ACM MobiHoc*, 2007.
[7] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: A social network perspective," in *ACM MobiHoc*, 2009.
[8] A. Vahdate and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Duke University, Tech. Rep. CS-2000-06, June 2000.
[9] 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.
[10] T. Henderson, D. Kotz, and I. Abyzov, "The changing usage of a mature campus-wide wireless network," in *ACM MobiCom*, 2004.
[11] L. Jeremie, F. Timur, and C. Vania, "Evaluating mobility pattern space routing for dtns," in *IEEE INFOCOM*, 2006.

[12] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Performance analysis of mobility-assisted routing," in *ACM MobiHoc*, 2006.
[13] 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.
[14] W. Hsu, T. Spyropoulos, K. Psounis, and A. Helmy, "Modeling time-variant user mobility in wireless mobile networks," in *IEEE INFOCOM*, 2007.
[15] M. Ibrahim, P. Nain, and I. Carreras, "Analysis of relay protocols for throwbox-equipped dtns," in *WiOPT*, 2009.
[16] T. Spyropoulos, K. Psounis, and C. Raghavendra, "Spray and wait: An efficient routing scheme for intermittently connected mobile networks," in *ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN)*, 2005.
[17] X. Chen, J. Shen, T. Groves, and J. Wu, "Probability delegation forwarding in delay tolerant networks," in *IEEE ICCCN*, 2009.
[18] Y. Wang, W.-S. Yang, and J. Wu, "Analysis of a hypercube-based social feature multi-path routing in delay tolerant networks," *accepted to appear in IEEE Transactions on Parallel and Distributed Systems*.
[19] B. Gu, X. Hong, P. Wang, and R. Borie, "Latency analysis for thrown box based message dissemination," in *IEEE Globecom*, 2010.
[20] 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.

Appendix

A. Proof of Theorem 3

Consider the message spreading scheme for the homing phase. A spreading scheme can minimize the expected delay of HS if it has the following three characteristics at the same time: 1) it can let the source send the message copies to the homes with the minimum average delay by this scheme; 2) it can let the source spread the copies to most homes; and 3) it also can let the source spread the copies to most other nodes during the spreading process. Here, the second and third characteristics will ensure that, when two nodes meet to exchange their copies, this scheme can always transit the state to a state that can achieve the smaller expected delay than others. According to Lemma 1, the binary homing scheme satisfies the second and third characteristics. In the proof of Theorem 5, we show that any spreading scheme will lead to the same average delay for the homing phase. Thus, the binary homing scheme can minimize the expected delay of HS. Moreover, the *1*-spreading scheme can spread copies to the maximum number of nodes with the fastest speed to minimize the expected delay of HS according to Lemma 2. Therefore, HS can achieve the minimum expected delay.

B. Proof of Theorem 4

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

C. Proof of Theorem 5

First, we compute $D^{(1)}$. Consider the case where $h=1$. If the source node reaches the home at time t, and it does not meet any other nodes before time t, then the corresponding probability density function is $\Lambda e^{-(\Lambda+\lambda)t}$. If the source reaches the home at time t, it encounters another node at time t_1 $(0\leq t_1\leq t)$. The source gives α $(0\leq\alpha\leq C)$ copies to this node, and this node takes the α copies to the home after time increment t_2 $(t_2\geq 0)$. Then, the corresponding probability density function is $\Lambda^2\lambda e^{-\Lambda t-\lambda t_1-\Lambda t_2}$ and the average delay is $\alpha t+(1-\alpha)(t_1+t_2)$. Then, the average delay of the homing phase for this case is given by:

$$
\begin{aligned}
d=&\int_0^\infty \Lambda t e^{-(\Lambda+\lambda)t}dt\\
&+\int_0^\infty\int_0^\infty\int_0^t \Lambda^2\lambda(\alpha t+(1-\alpha)(t_1+t_2))e^{-\Lambda t-\lambda t_1-\Lambda t_2}dt_1dt_2dt\\
=&\frac{1}{\Lambda}.
\end{aligned}
$$

Moreover, if the source node encounters multiple other nodes, the average delay is still equal to $\frac{1}{\Lambda}$ when we recursively apply the above formula. Therefore, the message spread would not change the average delay of the homing phase. Now, we consider the case where $h>1$. Since nodes' visit to homes are independent and identical, the average delay of the homing phase is equal to d divided by h. That is, $D^{(1)}=\frac{d}{h}=\frac{1}{h\Lambda}$.

Second, we derive the upper bound of $D^{(2)}$. The average delay of the *1*-spreading scheme includes two parts. The first part is the average delay for each home in H_+ to spread its extra copies to the network, denoted by $D_1^{(2)}$. Consider a home that has k extra copies. Then, the corresponding average delay is the average time for the home to encounter the first k mobile nodes, which is no more than the average time for the home to encounter each node. Thus, we have $D_1^{(2)}\leq\frac{1}{\Lambda}$. The second part is the average delay for each home, without copies, to receive copies that are spread by the homes in H_+, i.e., the average delay for $C-|H_+|$ nodes to send their copies to the homes in H_0 that have no copies. In fact, the delay for the first home in H_0 to receive a copy is the expected delay for a node visiting the home divided by $C-|H_+|$, i.e., $\frac{1}{(C-|H_+|)\Lambda}$. The delay for the second home in H_0 to receive a message copy is $\frac{1}{(C-|H_+|-1)\Lambda}$, and so on. Let $k=min\{|H_0|-1,C-|H_+|\}$, we have:

$$D_2^{(2)}=\frac{1}{k}\sum_{i=0}^{k-1}\frac{1}{(R-|H_+|-i)\Lambda}\leq\frac{1}{\Lambda}$$

By combining the average delay of the two parts, we have $D^{(2)}\leq\frac{2}{\Lambda}$.

Finally, we compute $D^{(3)}$. In the fetching phase, the destination will fetch the message from C homes if $C\leq h$. The corresponding expected delay is $\frac{1}{C\Lambda}$. If $C>h$, the destination will fetch the message from one of the h homes or the $C-h$ nodes that hold the copies. The probability density function for this delivery delay is $(h\Lambda+(C-h)\lambda)e^{-h\Lambda-(C-h)\lambda}$. Then, the corresponding expected delay is $\frac{1}{h\Lambda+(C-h)\lambda}$. By combining the results of the two cases, we can deduce the theorem.