Up-and-Down Routing Through Nested Core-Periphery Hierarchy in Mobile Opportunistic Social Networks

Huanyang Zheng and Jie Wu, *Fellow, IEEE*

***Abstract*—Routings in Mobile Opportunistic Social Networks (MOSNs) are challenging problems due to their dynamic network structures. This paper finds that MOSNs exhibit a Nested Core-Periphery Hierarchy (NCPH). In NCPH, few active nodes with large weighted degrees are the network core, while the network peripheries are composed of inactive nodes with small weighted degrees. 'Nested' indicates that the NCPH is preserved when periphery nodes are iteratively removed. A distributed labeling scheme is proposed to determine the NCPH with a granularity control. Based on the NCPH, this paper proposes an up-and-down routing protocol for MOSNs, which includes an upload phase and a download phase. A message can be uploaded from the source to the network core, through iteratively forwarding the message to a relay that has a higher position in the NCPH. Then, space-efficient Bloom-filter-based routing hints are introduced to download messages from the network core to the destination. Compared to traditional hierarchical routings, the up-and-down routing protocol achieves a better balance among the data delivery delay, ratio, and cost. Extensive experiments demonstrate that the proposed routing protocol achieves a competitive performance on the data delivery delay and ratio, with a small cost on the prior information maintenance and a low forwarding cost.**

***Index Terms*—Mobile opportunistic social networks, nested core-periphery, Bloom filter, routing hints.**

I. Introduction

In recent years, we have witnessed the emergence of a new kind of network known as the *Mobile Opportunistic Social Network* (MOSN), where people contact each other by chance using mobile wireless devices. A classic example of an MOSN is a scenario in which people walk around with smartphones and communicate with each other via Bluetooth or WiFi while within each other's transmission range [1], [2], [3]. Routings in MOSNs are characterized by intermittent network connectivity, where instantaneous end-to-end paths may not exist. Therefore, the prior knowledge on the network state information (e.g., the contact history) is extremely important for improving the MOSN routing performance (e.g., the data delivery delay, ratio, and cost). However, the prior state information collection and maintenance is usually difficult and costly due to the dynamic nature of MOSNs.

The network structural information is useful for optimizing the routings in MOSNs. Leskovec et al. [4] found that social networks exhibit the *Nested Core-Periphery Hierarchy* (NCPH). We show that MOSNs also exhibit the NCPH, since wireless devices are actually carried by people. This property is shown in Fig. 1: majority inactive nodes have small degrees and low contact frequencies, and they become the network peripheries on the left; minority active nodes have large degrees and high contact frequencies, and they are the network core on the right. 'Nested' indicates that the core-periphery hierarchy is preserved, when periphery nodes (and their associated links) are removed. Specifically, after iterative removal, the last remaining node is called the *root node*. To utilize these MOSN structural properties, we propose an up-and-down routing protocol, which has an *upload phase* and a *download phase*, as described in the following paragraphs.

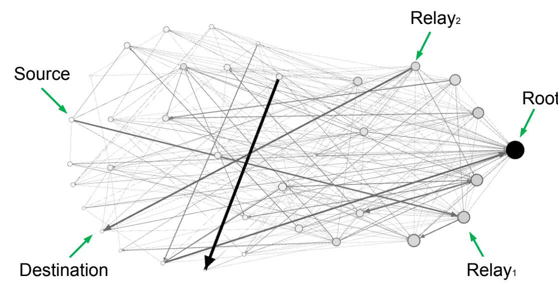


Fig. 1. The contact network in a primary school [5]. Nodes represent people (only nodes labeled "10xx" are selected), where the node with a higher degree is larger and darker. Weighted edges represent contact frequencies, where the edge with a higher weight is thicker and darker. Contacts are directional and are represented by arrows.

In the upload phase, a single message copy is used to deliver the message from the source node to the network core, by utilizing the NCPH, without resorting to other prior information. The NCPH enables a message to be uploaded from any node to the root node, through iteratively forwarding this message to a relay that has a higher position in the NCPH than the message holder. The NCPH can be determined by an iterative labeling process based on weighted node degrees, discounting neighbors that have already been labeled. The NCPH is *not* equivalent to the hierarchy in BubbleRap [6], which ranks nodes by degrees (or weighted degrees). The upload phase uses only one copy, but achieves a competitive delivery delay with a low forwarding cost.

The download phase delivers the message from the network core to the destination. Extra network information is introduced to provide routing guidance. At this stage, if each node maintains an accurate routing table as the extra information, then the delivery delay and ratio are improved at the expense

H. Zheng and J. Wu are with the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, 19122, USA email: {huanyang.zheng, jiewu}@temple.edu

Manuscript received XXX, XX, 2016; revised XXX, XX, 2016.

of oversized routing tables. To balance the storage demand, we compress the desired network information into Bloom-filter-based hints, which reduce the hint storage demand at the expense of hint accuracy. To deal with the inaccurate hints, the download phase uses multiple copies to ensure the delivery. Moreover, Bloom-filter-based hints enable multiple paths for the message download, and thus using multiple copies can effectively reduce the download delay. Meanwhile, *the routing hint update is directional*, which follows the NCPH. Therefore, the routing hints have a relatively small maintenance cost. These tradeoffs enable our routing protocol to achieve a good data delivery delay and ratio, with a low prior information collection cost and a low forwarding cost.

Our key contributions are fourfold:

- We systematically explore the MOSNs, verifying the existences of the NCPH. An up-and-down routing protocol is proposed by utilizing the NCPH. It has a good data delivery delay and ratio, with a low prior information collection cost and a low forwarding cost.
- We study the message upload in different network hierarchies. We show that the NCPH has a subtle difference from the degree or weighted-degree network hierarchies. A distributed labeling scheme is proposed to determine the NCPH with a granularity control.
- We introduce Bloom-filter-based hints as the download routing guidance to achieve a good data delivery delay and ratio, with a low prior information collection cost and a low forwarding cost.
- Extensive real trace-driven experiments are conducted to evaluate the proposed solutions. Multiple real traces are analyzed. The experimental results are shown from different perspectives to provide insightful conclusions.

The remainder of this paper is organized as follows: In Section II, the related work is surveyed; in Section III, the problem is formulated; in Section IV, the NCPH is described; in Section V, the Bloom-filter-based routing hint is described; in Section VI, the up-and-down routing protocol is presented; in Section VII, we evaluate the proposed scheme; and finally, in Section VIII, we conclude the paper.

II. Related Work

Routings in MOSNs have a tradeoff between the routing performance and the prior knowledge on the network state information. The first generation of MOSN routing algorithms ignores the importance of the prior information. For example, message holders in Spray and Wait [7] keep sharing message copies with new encountered nodes that have no copies (spray phase), until only one copy is left waiting for the destination (wait phase). Since no prior information is utilized, the performance of Spray and Wait is much worse than that of the state-of-the-art algorithms [6], [8], [9], in terms of a larger delivery delay, a lower delivery ratio, and a higher forwarding cost. Another example is that Epidemic routing [10], or flooding routing, replicates messages in every node encounter. A node without a message copy can always obtain one message copy, if it encounters a node with a message copy. Epidemic routing can archive the lowest message delivery delay and the highest message delivery ratio, at the expense of unacceptable forwarding costs. Clearly, the prior information is needed to balance the tradeoff among the data delivery delay, data delivery ratio, and the forwarding cost.

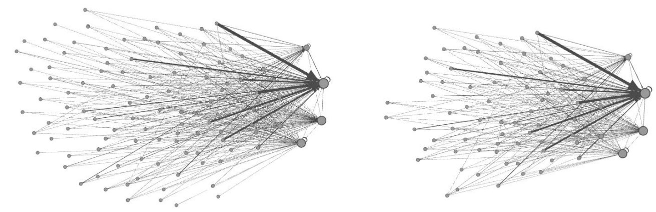
After a short period of time, researchers become aware of the importance of the prior information [11]. Therefore, routings with prior information such as contact history [8], social centrality [12], or clustering [13] are developed. For example, PROPHET [14] uses the contact history to make message forwarding decisions. Encounter probabilities are maintained by each node, and the message is always forwarded to a relay that has a higher encounter probability with the destination. Nodes that are encountered frequently have an increased encounter probability, whereas older contacts are degraded over time. Liu et al. [15], [16] designed routing protocols through cyclic mobile space. Delegation Forwarding [8] is another routing protocol that is based on the contact history, where nodes maintain historical records on their contacts. Each node will remember the relay that has the highest contact frequency with the destination in the past encounters. Messages are forwarded, only if the newly encountered node has a higher contact frequency with the destination than the message holder has ever seen. The number of message copies in Delegation Forwarding is expected to be square root with respect to the number of nodes in the network. The social centrality has also been used as the prior information in some routing protocols. For example, Gao et al. [12] exploited transient social contact patterns for the message forwarding. Contact frequencies are considered to be changing smoothly over time (people have more contacts during the daytime than the night). Only the node contact frequency at the current time is useful for the message forwarding. SimBet [17] uses social similarities and betweenness centralities to forward the message. Messages are always forwarded to relays that have high social similarities and betweenness centralities with respect to the destination. Clustering information is also used in some routing protocols. Tao et al. [13] designed an adaptive clustering algorithm for the routing. Nodes are clustered adaptively with respect to their contact frequencies. Messages are delivered through inter-cluster contacts and intra-cluster contacts. However, it is expensive to collect and maintain this prior information, due to the dynamic nature of the MOSN [18], [19].

Researchers have not focused on the network structural information until BubbleRap [6], which is a hierarchical approach. BubbleRap uses ranks (node degrees or weighted degrees) to layer nodes, where the messages are iteratively forwarded to nodes with higher ranks, until a local community that contains the destination is encountered. However, it remains unknown whether the messages will become trapped in some local high-rank nodes or not. In contrast, our NCPH is trap-free, and thus, it outperforms the hierarchy in BubbleRap in terms of routings. Furthermore, BubbleRap tacitly assumes that high-rank nodes are connected to a relay in the same community as the destination, which is not always true. Alternatively, we deal with this problem through Bloom-filter-based routing hints. Li et al. [20] also proposed a routing protocol, based on the network structural information. A topology control problem is studied in a predictable MOSN, where the time-evolving network topology is known a priori or can be predicted. Such a time-evolving MOSN is modeled as a directed space-time graph, which includes both spacial and temporal information. A connected topology is constructed to support the routing between any two nodes with a minimized cost. In contrast, the NCPH in our paper is a natural structure of the MOSN, i.e., it is not constructed. Peng et al. [21] studied the structure of road networks to support the routing in vehicular ad hoc networks. The practical attributes of road networks are sampled in main cities of Europe and the USA. A new graph metric, called characteristic central length, is proposed to estimate the average shortest-path length of a large-scale road network. Real road networks from Europe and the USA are shown to have certain routing patterns in terms of the characteristic central length. However, this paper focuses on MOSNs that are composed of moving people with wireless devices, and therefore different structural patterns are explored. Wei et al. [22], [23] explored social structures for the routing in MOSNs. High-quality social structures are constructed by considering both frequency and duration of contacts. To improve the performance of social-based message delivery, the community evolution problem is also addressed in [22]. Distributed clustering algorithms are developed, such that the overlapping communities and bridge nodes (i.e., connecting nodes between communities) can be dynamically detected for the message delivery. However, our NCPH is a better social structure than the approach in [22].

III. Problem Formulation

In MOSNs, people contact each other by chance using mobile wireless devices. We model MOSNs as directed weighted networks. The contact frequency from nodes i to j is recorded as the link weight [24], which is denoted as λ_{ij}. λ_{ij} is formally defined as the number of contacts from nodes i to j within a unit time. A larger λ_{ij} indicates a lower average link delay from nodes i to j. We assume the links to be symmetric, i.e., $\lambda_{ij} = \lambda_{ji}$. The weighted degree is the sum of the weights associated with every outgoing link incident to the corresponding node. Let w_i denote the weighted degree of the node i, i.e., $w_i = \sum_j \lambda_{ij}$. A larger weighted degree means that the node is more active. For MOSNs, the node activity (weighted degree) is more important than the node connectivity (degree), since messages are delivered via contacts.

This paper studies a hierarchical routing protocol that includes both unicast and multicast schemes for MOSNs. The objective is to achieve a balanced performance on the data delivery delay, ratio, and cost, through utilizing the natural MOSN hierarchy of the NCPH. We do not consider the routing that is constrained by the data delivery delay, ratio, or cost. However, in our future work, these constraints can be considered, through controlling the the number of hierarchical levels for a message to go up and down in the MOSN. Section IV describes the construction and the property of the NCPH, which is the foundation. Section V presents the space-efficient Bloom-filter-based routing hint, which records the indirect contact frequencies of a node and its descendants in the NCPH (i.e., the estimated data delivery delay from a node to its descendants in the NCPH). Then, Section VI describes the up-and-down routing protocol, both unicast and multicast versions, which includes the upload phase and the download phase. In the upload phase, the message is uploaded from the source to the root node (the node with the highest position in the NCPH), through iteratively forwarding this message to a relay that has a higher position in the NCPH than the message holder. In the download phase, the message is downloaded from the root node to the destination (or destinations in multicast), based on the Bloom-filter-based routing hint.



(a) Complete trace. (b) 30% removed.

Fig. 2. The NCPH in the MIT trace.

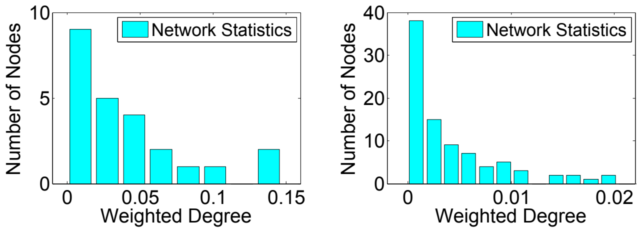
IV. Nested Core-Periphery Hierarchy in MOSNs

A. MOSN Structures

The classic scenario of an MOSN is that people walk around with smartphones and communicate with each other via Bluetooth or WiFi when they are within each other's transmission range. Since wireless devices are actually carried by people, MOSNs inherit the NCPH from social networks [4], in terms of weighted degrees. An active person that has a large degree in the social network should have a larger weighted degree in the corresponding MOSN. The structures of MOSNs, while similar, are significantly more dense than the structures of social networks. Connections in MOSNs are dense, since the transmission range of Bluetooth or WiFi is relatively large in this scenario. An example is the Infocom06 trace [25], where several hundreds of people are in few rooms during the academic conference of Infocom06.

To verify the existence of the NCPH, we look into the MIT [26] trace, as shown in Fig. 2. Its settings remain the same as those in Fig. 1. The MIT trace consists of one hundred Nokia 6600 smart phones with context applications developed by the University of Helsinki. The information collected includes call logs, Bluetooth devices in proximity, cell tower IDs, application usage, and phone status (such as charging or idle), which comes primarily from the Context application. The data represents approximately 500,000 hours of data on the users' location, communication, and device usage behavior. In Fig. 2, nodes represent people, where a node with a higher weighted degree is larger and darker. Weighted edges represent contact frequencies, where the edge with a higher weight is thicker and darker. Then, Fig. 2(a) shows the complete trace, which has a core-periphery hierarchy. Fig. 2(b) shows the partial trace, where 30% of nodes with low-weighted degrees are removed. Fig. 2(b) also exhibits a core-periphery hierarchy, i.e., the NCPH exists in the MIT trace.

We also study the weighted degree distribution of MOSNs. Experiments in the ST_Andrews [27] and MIT [26] traces are



(a) ST_Andrews trace. (b) MIT trace.

Fig. 3. The weighted degree distribution in MOSNs.

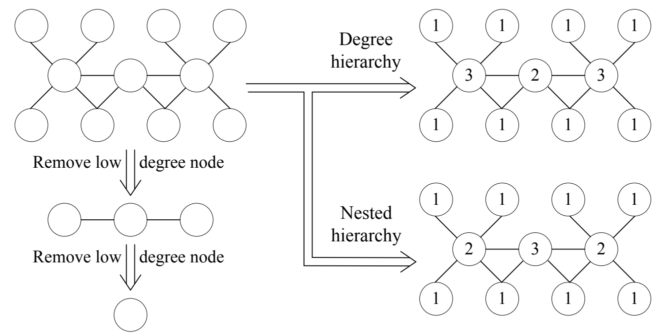


Fig. 4. An illustration of the subtle difference between the NCPH and the degree (or weighted degree) hierarchy. A node labeled by a larger number has a higher position in the hierarchy.

conducted, and the results are shown in Fig. 3. The unit of the weighted degree is the number of contacts per minute. The ST_Andrews trace includes a mobile sensor network, comprising T-mote invent devices carried by human users and Linux-based base stations that bridge the sensors to the wired network. For the ST_Andrews trace, devices with IDs 16, 26 and 27 are removed, since no data was collected. In Fig. 3, the node activity distributions have heavy tails. Few network core nodes in MOSNs hold a large fraction of total contact frequencies. More experiments are conducted based on [27], [26], [25], [28], where all nodes are classified into internal and external nodes. Only internal nodes are analyzed, since contacts of external nodes are not collected. The results are shown in Table I. In the Cambridge and the ST Andrews traces, the top 20% nodes hold more than half of the total contacts.

B. The Construction and The Property of The NCPH

This subsection describes the construction and the property of the NCPH. Note that the NCPH is totally different than the degree or weighted degree hierarchies used in BubbleRap [6]. For a clear explanation, we have the following definitions:

Definition 1: The NCPH defines the hierarchy of nodes in a network, through iteratively removing nodes with the lowest degrees (or weighted degrees) among their neighbors. The nodes removed earlier have lower hierarchies, while the nodes removed later have higher hierarchies.

Definition 2: The degree hierarchy defines the hierarchy of nodes in a network, according to the degree of the node. The nodes with smaller degrees have lower hierarchies, while the nodes with larger degrees have higher hierarchies. The weighted degree hierarchy is defined in a similar manner.

To better show the subtle difference between these two hierarchies, an example is provided in Fig. 4 (link weights to be identical for simplicity). The original network is shown on the top left corner of Fig. 4, while these two hierarchies are shown on the right side. Numbers within nodes represent their hierarchies. The key observation and insight are described as follows. (1) The degree (or weighted degree) hierarchy has local maximums, i.e., the two nodes with the highest hierarchies are not connected. If we use the BubbleRap [6] and iteratively forward the message to an encountered neighbor with a larger degree (or weighted degree) than the message holder, then local maximums are likely to be encountered. The message may not be uploaded to the node with the largest degree (or weighted degree). (2) MOSNs have the NCPH. If we iteratively forward the message to an encountered neighbor that has a higher position in the nested hierarchy, then this message is likely to be forwarded to the node that has a highest position in the nested hierarchy (i.e., root node). Since MOSNs inherit the NCPH from social networks, there are *many fewer local maximums*. Therefore, the NCPH facilitates the message upload in hierarchical routing protocols.

We construct the NCPH through a distributed labeling scheme that works through synchronous node iterations. All nodes operate in synchronous rounds. In MOSNs, a round can be implemented as a time period, which is sufficiently long for each node to exchange the information with all of its neighbors. This scheme is shown in the following:

Algorithm 1 Distributed labeling scheme

1: All nodes are initially unlabeled and operate in synchronous rounds. We define the *effective weighted degree* of a node as the summation of its link weights to all of its unlabeled neighbors.

2: A node labels itself, only if it has the lowest effective weighted degree among all its unlabeled neighbors. Its label is set to be the largest label among all its labeled neighbors plus one. However, if it does not have a labeled neighbor, then its label is set to be one.

A larger label means that the corresponding node has a higher position in the NCPH. This distributed labeling scheme is equivalent to the process of iteratively removing the nodes with the lowest weighted degrees among their neighbors. A node that is labeled earlier has a lower hierarchy, while a node that is labeled later has a higher hierarchy. Considering the structural dynamics of real MOSNs, Algorithm 1 is executed periodically to adjust the node labels, depending on how fast the node contact patterns change. Newly arrived nodes will have the lowest label in the NCPH, since they are new network peripheries. Once all nodes are labeled, we have:

Definition 3: When all nodes in an MOSN are labeled, a node is called a root node, if it has the largest label among all its neighbors.

As previously mentioned, MOSNs are likely to have very few root nodes. For further verification, we also look into the real MOSN traces [27], [26], [25], [28] in Table I. The last column of Table I shows the number of root nodes in these

TABLE I
MOSN STATISTICS

CRAWDAD Tace	Total Contacts	Trace Duration	Internal Nodes	External Nodes	Contacts of Top 20% Nodes	Number of Roots
Intel	2,766	6 days	9	128	30.72%	1 node
Cambridge	6,732	7 days	12	223	51.27%	1 node
Infocom06	28,126	4 days	41	264	29.83%	1 node
Sigcomm09	285,880	5 days	76	11,938	43.64%	1 node
ST_Andrews	112,265	79 days	27	N/A	55.14%	1 node

traces. One remarkable observation is that these traces have only one root node. Consequently, we have:

Theorem 1: In an MOSN with only one root node, the message can be uploaded from any node to the root, if the message holders iteratively forward the message to the relays that have larger labels than they do themselves.

Theorem 1 is intuitive, which can be proven by a simple contradiction. We also propose strategies to deal with multiple local maximums (i.e., multiple root nodes) in Section VI. The NCPH in MOSNs is useful for optimizing the routings, especially for the upload phase. All nodes can find the root through iteratively forwarding the message to the relays that have larger labels than they do. However, a potential problem is that the distributed labeling scheme for the NCPH may converge slowly. The next subsection will study the converge speed of the distributed labeling scheme.

C. The Bound of The NCPH

The NCPH is bound as stated in the following:

Theorem 2: Suppose that the MOSN has N nodes and the weighted degree distribution follows power-law in a nested manner. Then, Algorithm 1 is expected to terminate within $\Theta(\ln N)$ rounds of synchronous node iterations. The maximal node label is also $\Theta(\ln N)$.

Proof: The key idea is to show that a constant percentage of unlabeled nodes are expected to label themselves in each round of Algorithm 1. We start with the upper bound. Since the weighted degree distribution follows power-law, we have:

$$P_w = \frac{\alpha-1}{w_{\min}} \cdot \Big(\frac{w}{w_{\min}}\Big)^{-\alpha} \qquad (1)$$

In Eq. 1, P_w denotes the fraction of nodes with a weighted degree of w. α is a constant parameter ranging from 2 to 3 [29]. $w_{\min}$ is also a constant parameter from where the power-law distribution holds. Eq. 1 indicates that majority nodes have small weighted degrees, while minority nodes have large weighted degrees. This is previously verified in Table I.

Initially, all the nodes are unlabeled. Let us consider the probability that an arbitrary node (say v) labels itself. Suppose v's weighted degree is w_v. Based on Eq. 1, the probability that a neighbor of v has a higher weighted degree than v is:

$$\int_{w_v}^{\infty} \frac{\alpha-1}{w_{\min}} \cdot \Big(\frac{w}{w_{\min}}\Big)^{-\alpha} \mathrm{d}w = \Big(\frac{w_v}{w_{\min}}\Big)^{1-\alpha} \qquad (2)$$

According to Algorithm 1, the node v labels itself when all of its w_v neighbors have larger effective degrees than v. Since all nodes are initially unlabeled, the probability that v labels itself is $(w_v/w_{\min})^{(1-\alpha)\times w_v}$. The expected percentage of nodes that will label themselves in the first round of Algorithm 1 is:

$$\begin{aligned}
&\int_{w_{\min}}^{\infty} \Big(\frac{w_v}{w_{\min}}\Big)^{(1-\alpha)\times w_v} \cdot \frac{\alpha-1}{w_{\min}} \cdot \Big(\frac{w_v}{w_{\min}}\Big)^{-\alpha} \mathrm{d}w_v \\
&> \int_{w_{\min}}^{2w_{\min}} \Big(\frac{w_v}{w_{\min}}\Big)^{(1-\alpha)\times w_v} \cdot \frac{\alpha-1}{w_{\min}} \cdot \Big(\frac{w_v}{w_{\min}}\Big)^{-\alpha} \mathrm{d}w_v \\
&> \int_{w_{\min}}^{2w_{\min}} \Big(\frac{w_v}{w_{\min}}\Big)^{2(1-\alpha)w_{\min}} \cdot \frac{\alpha-1}{w_{\min}} \cdot \Big(\frac{w_v}{w_{\min}}\Big)^{-\alpha} \mathrm{d}w_v \\
&= \frac{1}{2w_{\min}+1}\Big[1 - \frac{1}{2^{(\alpha-1)(2w_{\min}+1)}}\Big] = c \qquad (3)
\end{aligned}$$

Let c denote the result in Eq. 3. Note that α is a constant that ranges from 2 to 3 [29]. $w_{\min}$ is also a constant parameter. Therefore, c is a positive constant, meaning that more than a constant percentage of unlabeled nodes will label themselves in the first round of Algorithm 1. Then, in the second round, these labeled nodes are peeled off from the remaining network. This is because their connections are not counted in the effective degrees of unlabeled nodes. With the nested property, we consider that the effective weighted degree distribution for the unlabeled nodes remains the same. Through the same arguments in Eqs. 2 and 3, more than a constant percentage of unlabeled nodes will label themselves in the second round of Algorithm 1. By induction, we conclude that *more than* a constant percentage of unlabeled nodes will label themselves in each round of Algorithm 1. Since $|V| \times c^{-\log_c |V|} = 1$, Algorithm 1 is expected to terminate within $-\log_c |V|$ rounds. c is a positive constant that is smaller than one, and thus $-\log_c |V|$ belongs to $O(\ln |V|)$. Hence, the number of rounds for Algorithm 1 to terminate is expected to be $O(\ln |V|)$.

The proof of the lower bound is similar. We further show that *less than* a constant percentage of unlabeled nodes will label themselves in each round of Algorithm 1. Similar to Eq. 3, we have the following inequation:

$$\begin{aligned}
&\int_{w_{\min}}^{\infty} \Big(\frac{w_v}{w_{\min}}\Big)^{(1-\alpha)\times w_v} \cdot \frac{\alpha-1}{w_{\min}} \cdot \Big(\frac{w_v}{w_{\min}}\Big)^{-\alpha} \mathrm{d}w_v \\
&= \frac{\alpha-1}{w_{\min}} \int_{w_{\min}}^{\infty} \Big(\frac{w_v}{w_{\min}}\Big)^{(1-\alpha)\times w_v - \alpha} \mathrm{d}w_v \\
&< \frac{\alpha-1}{w_{\min}} \int_{w_{\min}}^{\infty} \Big(\frac{w_v}{w_{\min}}\Big)^{(1-\alpha)\times w_v} \mathrm{d}w_v \\
&= \frac{\alpha-1}{w_{\min}} \int_{w_{\min}}^{\infty} e^{(1-\alpha)\times w_v \times \ln(w_v/w_{\min})} \mathrm{d}w_v \\
&< \frac{\alpha-1}{w_{\min}} \Big[\int_{w_{\min}}^{ew_{\min}} \Big(\frac{w_v}{w_{\min}}\Big)^{(1-\alpha)w_{\min}} \mathrm{d}w_v + \int_{ew_{\min}}^{\infty} e^{(1-\alpha)w_v} \mathrm{d}w_v\Big] \\
&= \frac{\alpha-1}{w_{\min}} \Big[\frac{e^{(1-\alpha)w_{\min}+1} - 1}{(1-\alpha)w_{\min}+1} w_{\min} - \frac{e^{(1-\alpha)ew_{\min}}}{1-\alpha}\Big] = c^* \qquad (4)
\end{aligned}$$

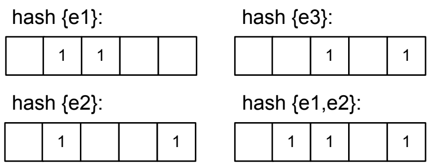


Fig. 5. A Bloom filter with $m = 5$ and $k = 2$. If two elements of e_1 and e_2 are inserted, a query for e_3 returns "in the set", which is a false positive.

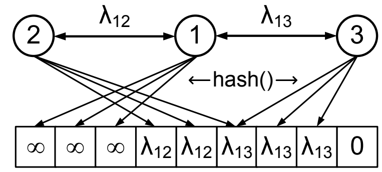


Fig. 6. The Bloom-filter-based routing hint of node 1.

Let c^* denote the result in Eq. 4. Clearly, c^* is a constant. Through the same argument, the number of rounds for Algorithm 1 to terminate is expected to be $\Omega(\ln N)$.

Combining the upper and lower bounds, we conclude that Algorithm 1 is expected to terminate within $\Theta(\ln N)$ rounds of synchronous node iterations. In each round, the maximal node label increases by one. Consequently, the expected maximal node label in NCPH is also $\Theta(\ln N)$. ■

D. The Granularity Control

Although Theorem 2 states that Algorithm 1 can converge within $\Theta(\ln N)$ rounds of synchronous node iterations under certain weighted degree distributions, sometimes we may need an even quicker convergence for a large-scale MOSN. Therefore, this subsection studies a granularity control for the distributed labeling scheme, in which the number of hierarchy levels are controlled [30]. For example, Algorithm 1 may return 20 different hierarchical levels for an MOSN with 100 nodes, but we only want to have 10 different hierarchical levels to facilitate the routing. In such an event, the granularity control is needed. We use the parameter, Δ, to control the granularity in the distributed labeling scheme:

Algorithm 2 Distributed labeling scheme (granularity control)

1: All nodes are initially unlabeled and operate on synchronous rounds. At each round, each unlabeled node designates a 1-hop neighbor, or itself, as its "max" status.

2: At the i^{th} round, a node labels itself, when (1) it has the lowest effective weighted degree among all its unlabeled neighbors, or (2) its effective weighted degree is smaller than $i \times \Delta$ and it is not in a "max" status. Its label is set to be the largest label among its labeled neighbors plus one. However, if it does not have a labeled neighbor, then its label is set to be one.

In MOSNs, a round can be implemented as a time period, which is sufficiently long for each node to exchange the information with all of its neighbors. The insight of Algorithm 2 is that, nodes in "max" status could ensure the establishment of the NCPH. Since some nodes in "max" status are unlabeled, not all the nodes would immediately label themselves within one round. This granularity control can tune the convergence speed. Note that a larger Δ brings a quicker convergence, and thus the maximal peer label is smaller.

V. Bloom-Filter-Based Routing Hints

The previous section describes the NCPH. Then, this section presents the space-efficient Bloom-filter-based routing hint, which records the indirect contact frequencies of a node and its descendants in the NCPH (i.e., the estimated data delivery delay from a node to its descendants in the NCPH).

A. Classic Bloom Filters

The classic Bloom filter [31], [32], [33], [34], [35] is a space-efficient probabilistic data structure. It is used, through the query operation, to check whether an element is a member of a set or not. The answer of "in the set" may be incorrect, the probability of which is called false positive rate. The answer of "not in the set" is guaranteed to be correct. Assume there are a total of n elements (denoted as e_1 to e_n) inserted into the Bloom filter, where each element belongs to $\{1, 2, ..., N\}$ ($n \ll N$). A classic Bloom filter employs a bit array of size m ($n < m \ll N$) to store these elements. There exist k different independent hash functions (denoted as h_1 to h_k), each of which maps the input element to one of the m array positions. Initially, all bits in the array are 0. To add an element e_i, the bits of array positions $h_1(e_i), ..., h_k(e_i)$ are set to be 1. To query a specified element e_j, k bits in the array positions of $h_1(e_j), ..., h_k(e_j)$ are checked. If any bits are 0, then "not in the set" is returned. Otherwise, it returns "in the set". Since the k bit positions of e_j may be covered by some other elements, e_j may be incorrectly verified as existing, resulting in a false positive. An example is illustrated in Fig. 5. The false positive rate [31] can be calculated as:

$$\left[1 - (1 - \frac{1}{m})^{nk}\right]^k \approx (1 - e^{-kn/m})^k \tag{5}$$

According to [31], the false positive rate can be minimized by setting $k = \frac{m}{n} \ln 2$. Bloom filter shows a tradeoff between the time complexity of a query and the space complexity of the storage. If we use a simple array to store these n elements, the time complexity of a query is $O(1)$, and the space complexity of the storage is $O(N)$. For linked lists, the time and space complexities are $O(n)$ and $O(n)$, respectively. For Bloom filters, the time and space complexities are $O(k)$ and $O(m)$, respectively. In general, we have $k \ll n < m \ll N$.

B. Construction of Bloom-Filter-Based Routing Hints

The objective of the Bloom-filter-based routing hint is to record the indirect contact frequencies of a node and its descendants in the NCPH (i.e., the estimated data delivery delay from a node to its descendants in the NCPH). In traditional IP networks, each node can simply record the best forwarder for a specified destination (e.g., Bellman-Ford algorithm) as the routing hints (or routing tables). However, the node storage capacity is limited in MOSNs, where tracking all the connectivity information is impossible. Therefore, Bloom-filter-based hints are introduced to reduce the hint storage

Algorithm 3 $Initialization_i$

Input: IDs and contact frequencies of the neighbors;
Output: Initialization completion;
Initialize a weighted array T of size m;
Initialize k independent hash functions $h_1, ..., h_k$;
for each neighbor j of node i **do**
  for each hash function h from h_1 to h_k **do**
    Set $T[h(j)] = \lambda_{ij}$;
Set the corresponding positions of node i to be ∞;

Algorithm 4 $Query_i$

Input: A specified node ID j;
Output: The contact frequency λ_{ij};
Initialize a weighted array R of size k;
for $s = 1$ *to* k **do**
  Set $R[s] = T[h_s(j)]$;
if R contains an element of 0 **then**
  return 0;
else
  return the most "popular" element in R;

Algorithm 5 $Update_i$

Input: Routing hints from all the neighbors of node i;
Output: Update completion;
for each neighbor j **do**
  if $label(j) < label(i)$ **then**
    for each possible node ID e **do**
      if $1/Query_j(e) + 1/\lambda_{ij} < 1/Query_i(e)$ **then**
        for each hash function h from h_1 to h_k **do**
          Set $T[h(e)] = 1/(1/Query_j(e) + 1/\lambda_{ij})$;

demand at the expense of hint accuracy. In other words, Bloom-filter-based hints take a smaller storage requirement, but they may be inaccurate in terms of false positives. Each node in MOSN keeps a Bloom-filter-based routing hint to track its descendants in the NCPH. However, the sizes of the routing hints in different nodes may not be identical.

Let N denote the total number of nodes in the MOSN, with node ID ranging from 1 to N. The proposed routing hint is an extension of the Bloom filter, where the bit table is replaced by a weighted table to record the contact frequency. An example is shown in Fig. 6, in terms of the Bloom-filter-based routing hint for node 1. Initially, each node stores the contact frequencies of its neighbors from lowest node ID to highest node ID. For example, node 2 is found as a neighbor of node 1, the corresponding table position of which is set to be λ_{12}. The same thing happens for recording node 3. However, the 6th table position has been taken by node 2, the value of which is then refreshed. Finally, the routing hint of node 1 stores itself, the corresponding table positions of which are set to be ∞. The routing hint also supports query operations, which returns the frequency of a queried node. A query operation looks into all the k array positions of the queried node, and returns the most "popular" value (the value that appears most frequently), if these positions are all non-zero values. For example, if node 2 is queried (for the routing

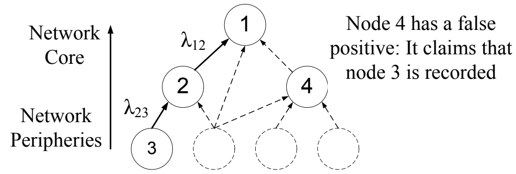


Fig. 7. An example of the routing hint updates, where the arrow represents the information flow of the routing hints. Node 3 sends its routing hint to node 2. After updating, node 2 sends its routing hint to node 1.

hint of node 1), λ_{12} is returned, since it appears twice and λ_{13} only appears once. If there is a table position of value 0 among the k positions, 0 is returned. In the event that several values are equally most "popular," the first met value will be returned. For the Bloom-filter-based routing hint of node i with the weighted table T, the initialization and query operation ($Initialization_i$ and $Query_i$) are described in Algorithms 3 and 4, respectively. The size of the routing hint, parameters m and k, is described in subsection V.D.

The time complexity of the hint initialization is the same as the Bloom filter, i.e., $O(nk)$. However, the time complexity of the query operation is changed to be $O(k \log k)$, since it needs to sort the values of the k array positions to find the most "popular" one. Moreover, the proposed routing hint supports the deletion operation, which deletes the most "popular" elements. The deletion operation provides a method for updating the connectivity information in MOSNs.

C. Maintenance of Bloom-Filter-Based Routing Hints

Inspired by the NCPH, the routing hint of a node is only sent to adjacent neighbors that have larger labels than that node (i.e., neighbors that have higher positions in the NCPH than that node). In other words, the routing hint update is directional, which is a stark contrast from the Bellman-Ford algorithm in RIP. Directional updates result in a relatively small cost on the routing hint maintenance. Moreover, the structure of the directional routing hint update is a directed acyclic graph (DAG) rather than a tree, as shown in Fig. 7. A node sends its routing hint to multiple higher-level neighbors (i.e., its parents in the NCPA). The NCPH enables multiple routing paths in the download phase to reach the destination.

Heterogeneous routing hint size is used to save the storage space, where the major periphery nodes record less information, and the minor core nodes record more information. Based on the received routing hints, the node updates its routing hint to record the estimated contact frequency (or *the expected shortest path delay*) with its descendants in the NCPH. The reciprocal of the contact frequency represents expected data delivery delay along the shortest path. Note that the information of all nodes are expected to be recorded by roots, since the other nodes are their descendants and the Bloom-filter-based routing hints do not have false negatives (an existing descendant will not be missed). An example is shown in Fig. 7. After receiving node 2's routing hint that contains the information on node 3 and λ_{23}, node 1 updates its routing hint to set up routing information on node 3. Since the expected delay from node 1 to 3 is $(1/\lambda_{12}+1/\lambda_{23})$, node 1 sets

λ_{13} to be $1/(1/\lambda_{12}+1/\lambda_{23})$. The algorithm for the routing hint update of node i (denoted as $Update_i$) is shown in Algorithm 5. The routing hints are updated during node encounters.

D. Sizes of Bloom-Filter-Based Routing Hints

In Fig. 7, it can be seen that nodes with higher positions in the NCPH record more information in terms of their routing hints, since they have more descendants. Therefore, the sizes of the routing hints for those nodes should be larger, i.e., m and k should be larger. This subsection discusses the values of m and k. We start with the expected number of inserted elements (i.e., the expected n) for each node:

Theorem 3: The expected number of inserted elements in the routing hint of a node with degree d (denoted as n_d), satisfies $n_d \leq d(\alpha-1)^{d-2}$, if (1) the corresponding network has a power-law degree distribution, and (2) the label of a node is proportional to its node degree.

The proof is shown in [36]. α is a constant parameter. It ranges from 2 to 3 for real-world networks [37]. α can be estimated by the network degree distribution, and the related tools are published in [38]. Based on Theorem 3, we have:

Theorem 4: For the node with degree d, the expected false positive rate of its routing hint would be no larger than ε ($\varepsilon \leq 1$), if $m \geq -\frac{\ln\varepsilon}{\ln^2 2} \times d(\alpha-1)^{d-2}$.

Proof: According to [31], $k = \frac{m}{n}\ln 2$ minimizes the false positive rate. Therefore, Eq. 5 can be rewritten as

$$\Big[1-(1-\frac{1}{m})^{nk}\Big]^k \approx (1-e^{-kn/m})^k = \frac{1}{2}^{\frac{m}{n}\ln 2} \leq \varepsilon \quad (6)$$

As a result, we have $\frac{m}{n}\ln 2 \ln\frac{1}{2} \leq \ln\varepsilon$ or $m \geq -\frac{\ln\varepsilon}{\ln^2 2} \times n$ Since Theorem 3 shows that $n_d \leq d(\alpha-1)^{d-2}$, we can derive $m \geq -\frac{\ln\varepsilon}{\ln^2 2} \times d(\alpha-1)^{d-2}$, and the proof completes. ■

Nodes with different degrees keep different sizes of routing hints. The network core nodes are generally more active and more powerful, and thus they should maintain larger sizes of routing hints. This is the "hotspot" effect, which is inevitable in MOSN routings. This is because the network core nodes are bottlenecks for high-performance routing algorithms. Unfortunately, Theorems 3 and 4 show that, as an upper bound, the node storage consumption may scale exponentially with respect to the node degree. However, the worse case is that roots need to record the information of all the other nodes, i.e., the storage consumption scales linearly with respect to the number of nodes in MOSNs. As our future work, the scalability issue can be solved through the clustering method, in which the MOSN is divided into small clusters. As a result, roots in a cluster only need to record the number of nodes in the cluster. Roots also act as cluster heads to guide routings.

VI. Up-and-Down Routing Protocol

This section describes the up-and-down routing protocol (for both unicast and multicast), which includes upload and download phases. The unicast upload phase is a single-copy routing, based on the NCPH in MOSNs. The unicast download phase is a multi-copy routing, which is guided by the Bloom-filter-based routing hints with copy controls. The multicast up-and-down routing is extended. Finally, we explain the up-and-down routing from another perspective.

Algorithm 6 Unicast Upload Phase for Node i

Input: Routing hints of node i and its neighbors;

for each neighbor j **do**
  if $Query_j(destiantion) > 0$ **then**
    Upload phase terminates, download phase starts;
for the inter-meeting neighbor j **do**
  if $label(i) < label(j)$ **then**
    Forward the message to node j;

A. Unicast Upload Phase

Theorem 1 states that the source can upload its message to the unique root node, through iteratively forwarding the message to the encountered relays that have higher positions (i.e., nodes with larger labels) in the NCPH than the message holder. However, it is not always necessary to upload the message to the root, because shortcuts may exist. Therefore, the upload termination condition is that the message is forwarded to a node, where the routing information on the destination (i.e., $Query(destination)$>0) is available in the routing hints of that node and its neighbors. Note that the routing hint of the root should include the information on all the nodes within the MOSN. The upload phase uses only one message copy, since it achieves competitive delivery delay with a low forwarding cost. Although multiple copies can improve the upload performance, they also bring additional costs. We consider that the cost outweighs the performance improvement for multiple copies in the upload phase. As a result, the upload phase uses only one copy, as described in Algorithm 6.

A potential problem for the upload phase is that, multiple root nodes may exist in large-scale MOSNs. We have several methods to solve the existence of multiple roots:

- The first method is to use the broadcast and registration for the root nodes to discover each other and then set up virtual connections before the up-and-down routing. Paths are pre-determined to connect different roots.
- The second method is the logical link removal. If we disable some existing connections, some root nodes can be removed. They are no longer root nodes.
- The third method is the probabilistic upload, where the message can jump out of a root with a carefully-designed probability. This idea is based on [36].

Clearly, the above methods bring additional costs to connect multiple roots in MOSNs. For large-scale MOSNs, we recommend the first method, in which a special node or fixed infrastructure is used as the registration server. The location of the registration server is broadcasted among the network. Roots can connect each other through this registration server. As a result, the additional costs can be bounded to a broadcast (discounting the routing costs among roots). Since the multiple-roots problem is potential, but not observed in current real traces, this problem is not discussed further. Note that the NCPH naturally exists in MOSNs. The NCPH has many fewer local maximums than the degree (or weighted degree) hierarchies in BubbleRap [6]. Since local maximums are few, the additional costs brought by multiple roots should scale well with respect to the network size.

Algorithm 7 Unicast Download Phase for Node i

Input: Routing hints of node i and its neighbors;
for the inter-meeting neighbor j **do**
  $c_1 = Query_i(destiantion)$;
  $c_2 = Query_j(destiantion)$;
  if $label(i) > label(j)$ in the NCPH **then**
    Hand over $\lceil c_2 \times c/(c_1 + c_2) \rceil$ copies to node j;
    Update c to be the number of the remaining copies;

B. Unicast Download Phase

Whenever the destination information is available in the routing hint, the upload phase switches to the download phase. Since roots record the information of all the other nodes, the upload phase is guaranteed to terminate. Meanwhile, at least one correct path to the destination exists. In the upload phase, we only use one message copy to save the data forwarding cost. However, in the download phase, we use multiple copies. There are two reasons for using multiple copies:

- The first reason is that the Bloom-filter-based routing hints have false positives, i.e., a node may incorrectly claim the destination as its descendant. The routing hint storage is reduced at the expense of the routing hint accuracy, which causes a lower delivery ratio. Therefore, multiple copies are used to mitigate negative routing effects caused by the inaccurate routing hints.
- The second reason is that using multiple copies can reduce the download delay. Although the routing hint within a relay only records the shortest path to the destination, the opportunistic nature can still accelerate the download phase. When the message holder opportunistically meets a neighbor with a secondary path, partial copies will be passed to that neighbor to reduce download delay.

The node that terminates the upload phase would replicate the message for the download phase. In the download phase, the message copies are only passed to *relays that have smaller labels than the message holder*. Moreover, the number of message copies allocated to the two nodes are proportional to their estimated contact frequencies (i.e., reciprocal of the expected delay) with the destination. This information is recorded in their Bloom-filter-based routing hints and can be accessed by the query operation. If node i currently holds c copies, and node j contacts node i, then the number of copies handed over to node j can be calculated as:

$$\left\lceil \frac{Query_j(destination)}{Query_i(destination) + Query_j(destination)} \times c \right\rceil \quad (7)$$

Clearly, the node with a smaller expected shortest path delay is allocated with more message copies. The download phase is formally described in Algorithm 7. Due to false positives, a message may be sent to a node, which does not record the destination in its bloom-filter-based hint. In this event, the message will not be discarded. As a result, the next subsection will discuss the copy control in the unicast download phase, i.e., the number of copies (parameter c) needed to resist the false positives in the Bloom-filter-based routing hints.

C. Copy Control in Unicast Download Phase

This subsection discusses the copy control (in terms of the parameter c) in the unicast download phase, assuming that the average false positive rate of the routing hint is ε. This assumption posits that a node has an average probability of ε, to incorrectly claim the destination as its descendant. Since false positives may lead to incorrect forwarding decisions, multiple copies are used in the unicast download phase to mitigate this problem. Let us focus on a node that is a predecessor of the destination in the NCPH. Suppose this node currently holds c message copies. The average node degree in the network is $\langle d \rangle$. Since the average false positive rate of the routing hint is ε, $\varepsilon\langle d \rangle$ neighbors of the message holder would incorrectly claim the destination as its descendant. The worst case is that only one neighbor is a true predecessor of the destination. Therefore, the number of copies that can be correctly handed over is:

$$\frac{1}{1 + \varepsilon\langle d \rangle} \times c \quad (8)$$

Eq. 8 validates when $Query(destination)$ is uniformly distributed. If the current message holder needs to take at least h hops to reach the destination, then the expected number of copies that can reach the destination is:

$$\frac{1}{(1 + \varepsilon\langle d \rangle)^h} \times c \quad (9)$$

If the value in Eq. 9 is smaller than one, it means that less than one message copy would eventually reach the destination, leading to a low delivery ratio. Therefore, the average false positive rate of the routing hint should be sufficiently small, as to provide an effective routing that can deliver the message copy to the destination. When $\varepsilon\langle d \rangle$ is sufficiently small, Eq. 9 can be approximated by:

$$\frac{c}{(1 + \varepsilon\langle d \rangle)^h} \approx \frac{c}{e^{\varepsilon\langle d \rangle h}} \quad (10)$$

This is because $1+\varepsilon\langle d \rangle \approx e^{\varepsilon\langle d \rangle}$, when $\varepsilon\langle d \rangle \to 0$. If c satisfies $c \geq e^{\varepsilon\langle d \rangle h}$, then the following theorem can be obtained as the copy control in the unicast download phase:

Theorem 5: When the number of copies used in the unicast download phase is more than $e^{\varepsilon\langle d \rangle h}$, at least one copy is expected to be delivered to the destination.

Theorem 5 shows that a certain number of copies (i.e., message forwarding cost) is needed to obtain a high delivery ratio. An important insight is that the resultant number of copies grows exponentially with respect to the false positive rate, which should be small enough. Note that the false positive rate, ε, can in turn be controlled by Theorem 4. Theorems 4 and 5 represent the tradeoff between the message forwarding cost and the routing hint storage consumption, in terms of the data delivery ratio. A larger storage consumption results in a smaller false positive rate, and thus, a higher data delivery ratio. Using Bloom-filter-based routing hints as the routing guidance, the unicast download routing phase can achieve a good data delivery ratio, with a low prior information collection cost and a low forwarding cost. Another notable point is that h can also be controlled through the distributed

Algorithm 8 Multicast Upload Phase for Node i

Input: Routing hints of node i and its neighbors;

for each $destination$ in the multicast **do**
  if $Query_i(destination) = 0$ **then**
    Upload phase continues;
    for the inter-meeting neighbor j **do**
      if $label(i) < label(j)$ **then**
        Forward the message to node j;
  Upload phase terminates, download phase starts;

Algorithm 9 Multicast Download Phase for Node i

Input: Routing hints of node i and its neighbors;

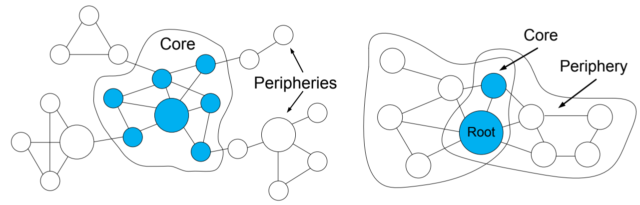
for the inter-meeting neighbor j **do**
  Initialize $c_1 = 0$ and $c_2 = 0$.
  for each $destination$ in the multicast **do**
    $c_1 = c_1 + Query_i(destiantion)$;
    $c_2 = c_2 + Query_j(destiantion)$;
  if $label(i) > label(j)$ in the NCPH **then**
    Hand over $\lceil c_2 \times c/(c_1 + c_2) \rceil$ copies to node j;
    Update c to be the number of the remaining copies;

labeling scheme with the granularity control (i.e., Algorithm 2). h is bounded by the number of hierarchy levels in the NCPH. The parameter, h, represents the tradeoff between the message forwarding cost and the convergence speed of the distributed labeling scheme. A slower convergence brings a higher message forwarding cost.

D. Multicast Up-and-Down Routing Protocol

Previous subsections describe the unicast up-and-down routing protocol: the message is uploaded from the source to the root node in the NCPH though a single-copy routing, and then, the message is downloaded from the root node to the destination through a multi-copy routing with a copy control. This subsection discusses the multicast up-and-down routing protocol, where the source needs to send the message to multiple destinations.

The multicast upload phase is identical to the unicast upload phase, where the message is uploaded to the root through iteratively being forwarded to a relay that has a higher hierarchical level in the NCPH than the message holder. The upload phase uses only one message copy for both unicast and multicast. However, the unicast and multicast upload phases have different termination conditions, since multiple destinations need to be found in the multicast upload phase. The multicast upload phase terminates (i.e., it switches to the multicast download phase), only if all the destinations are recorded as descendants in the Bloom-filter-based routing hint of the message holder. Note that the root node records all the other nodes as descendants in its Bloom-filter-based routing hint. Therefore, the multicast upload phase will eventually switch to the multicast download phase. The multicast download phase is similar to the multicast upload phase. In the multicast download phase, the number of message copies allocated to the two encountered nodes are proportional to their total estimated contact frequencies to all the destinations (rather than a single destination in the unicast download phase). Clearly, the node, which has larger total estimated contact frequencies to all the destinations, should be allocated with more message copies. Formal description for the multicast upload and download phases are presented in Algorithms 8 and 9, respectively. Note that the multicast up-and-down routing protocol outperforms the repeated unicast up-and-down routing protocol, since duplicated upload phases in the repeated unicast are saved in multicast. Therefore, the up-and-down routing protocol can be applied to both the unicast and multicast scenarios.



(a) General core-periphery network. (b) The NCPH and communities.

Fig. 8. Another perspective of the up-and-down routing.

E. Up-and-Down Routing: Another Perspective

This subsection explains the up-and-down routing from another perspective. More specifically, we describe the NCPH in the MOSNs through another manner. The latest study by Yang and Leskovec [39] pointed out a paradigm for uncovering the modular network structures, based on the decomposition of a network into any combination of overlapping, nonoverlapping, and hierarchically organized communities. They reported that regions of the network where communities overlap have a higher density of edges than the nonoverlapping regions. Their study considers networks exhibiting a core-periphery hierarchy, where nodes that belong to multiple communities form the core of the network. Hence, the NCPH can result from the mechanism of the community formation [40], [41].

Additionally, we explain the NCPH in MOSNs from the view of overlapping communities. The classic scenario of an MOSN is that people walk around with smartphones and communicate with each other via Bluetooth or WiFi while within one another's transmission range. Since people have communities, nodes in MOSNs should also be organized in communities. However, a notable point is that not all the core-periphery hierarchies can be utilized by the upload routing. Fig. 8(a) shows a core-periphery hierarchy, where the network core and network peripheries are not organized in a nested manner. The upload routing in that hierarchy may encounter the local maximum problem. Meanwhile, Fig. 8(b) illustrates the NCPH in terms of overlapping communities. The nested property enables the message upload.

Moreover, the Bloom-filter-based routing hints abstract the NCPH in the MOSN as a DAG. The routing hint update is directional from the nonoverlapping regions of the MOSN to the overlapping regions. From the community's perspective, people, who belong to multiple communities, should "remember" the structural information for the routing. This is because they serve as bridges connecting different communities. This phenomenon is consistent with our real-life experience, i.e., "with great power comes great responsibility".

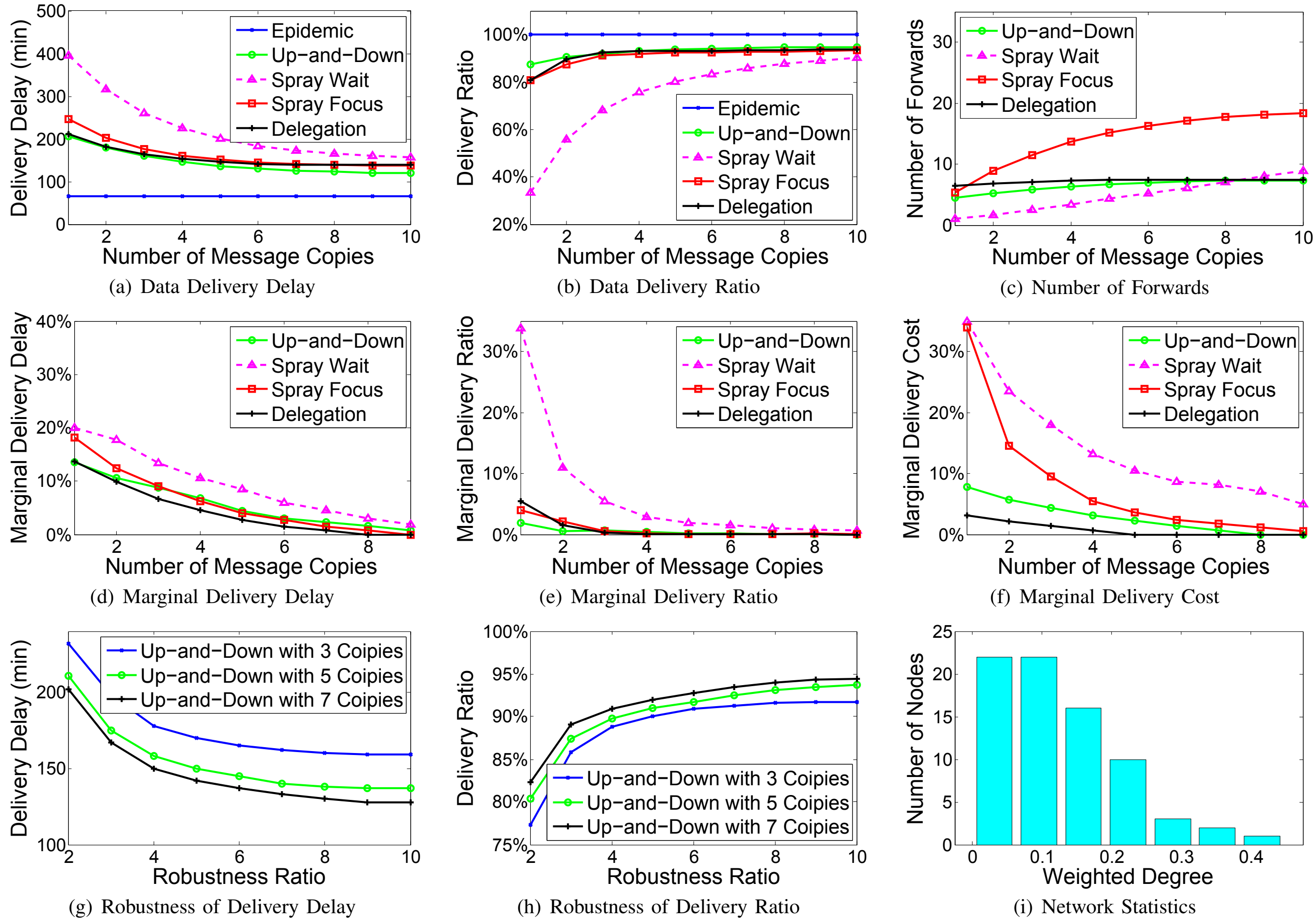


Fig. 9. Experimental results for unicasts in the Sigcomm trace.

TABLE II
STORAGE CONSUMPTION IN THE SIGCOMM TRACE.

Robustness ratio	2	3	4	5	6	7	8	9	10
Average false positive rate	38%	28%	17%	10%	6%	3%	2%	1%	0.7%
Storage saving percentage	81%	72%	62%	53%	44%	31%	21%	10%	0%

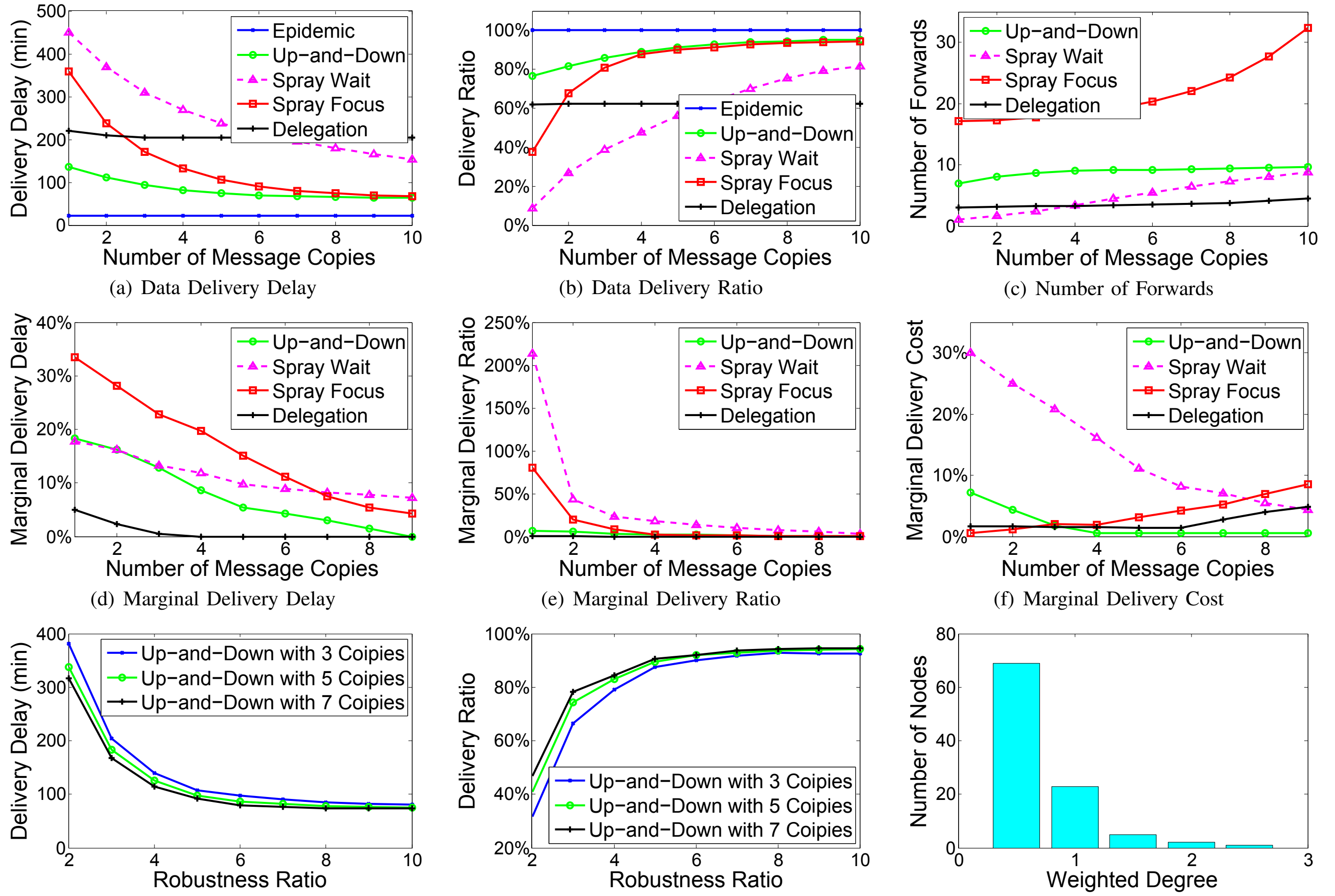
VII. EVALUATION

Extensive experiments are conducted to evaluate the up-and-down routing protocol. Evaluation results are shown from different perspectives to provide insightful conclusions.

A. Traces and Settings

Two traces are used. One is a real trace, the Sigcomm trace [28]. It was collected during the Sigcomm 2009 conference in Barcelona, Spain. Around 100 smartphones were distributed to a set of participants during the first two days of the conference. Each participant was instructed to keep the device with them and powered on at all times, and to use the MobiClique application as the MOSN during the conference. The statistics of the Sigcomm trace are summarized in Table I. We have $\alpha = 2.5$ in Theorem 3 for the Sigcomm trace. The other trace used in our experiments is a synthetic trace, which is generated by the preferential attachment model [29]. This trace includes 100 nodes, while the average node degree is 10. α is set to be 2.1. In the synthetic trace, the weight of each link follows a uniform distribution from 0 to 0.1. To guarantee the network connectivity, one node in each small component is randomly selected to connect to the largest degree node.

In our experiments, the unit of the edge weight is determined as the number of contacts per minute. We use 500 minutes as the data delivery deadline. If the destination is not achieved before the deadline, then the data delivery is viewed as failed, and the deadline is regarded as the delivery delay. For the up-and-down routing protocol, we use the routing hints with sizes of $m = 10 \times d(\alpha - 1)^{d-2}$ based on Theorem 3. Here, the constant of 10 is the *robustness ratio*, which is defined as the ratio of the routing hint size to $d(\alpha - 1)^{d-2}$. A higher robustness ratio means the routing hints are more accurate, and also indicates higher storage demands, and therefore are supposed to bring better performance. The influence of the robustness ratio on the routing performance is observed and analyzed, such as the robustness of the delivery delay and ratio. We also study the storage saving percentages of the Bloom-filter-based routing hints, compared to the case where we use arrays to store the routing information. The relationships among the robustness ratio, the storage saving percentage, and the average false positive rate are studied.



(a) Data Delivery Delay (b) Data Delivery Ratio (c) Number of Forwards

(d) Marginal Delivery Delay (e) Marginal Delivery Ratio (f) Marginal Delivery Cost

(g) Robustness of Delivery Delay (h) Robustness of Delivery Ratio (i) Network Statistics

Fig. 10. Experimental results for unicasts in the synthetic trace.

TABLE III
STORAGE CONSUMPTION IN THE SYNTHETIC TRACE.

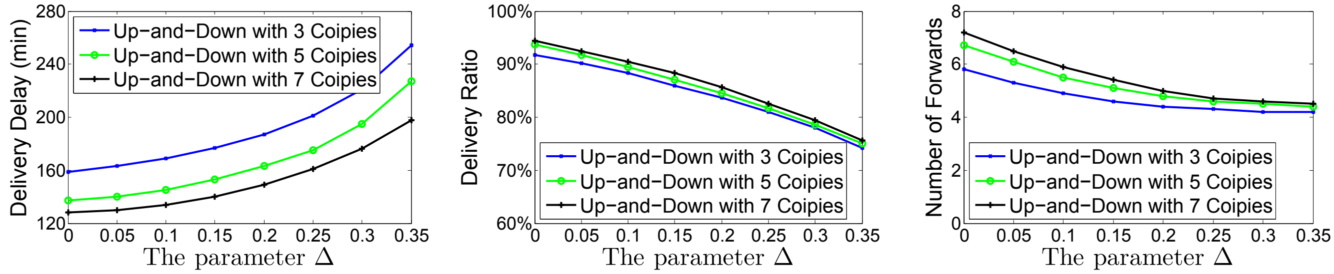
Robustness ratio	2	3	4	5	6	7	8	9	10
Average false positive rate	39%	24%	15%	9%	6%	4%	2%	1%	0.8%
Storage saving percentage	83%	74%	65%	57%	48%	39%	30%	22%	13%

B. Algorithms and Metrics for Comparison

We assign the following algorithms for comparison:

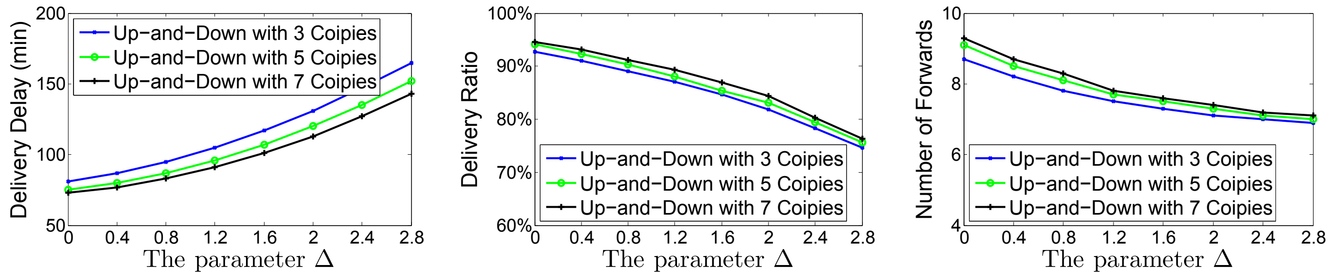
- Epidemic [10], where the nodes continuously replicate and transmit messages to newly discovered nodes that do not already possess a copy. Epidemic has the minimum data delivery delay and the highest forwarding cost among all the routing algorithms.
- Spray and Wait [7] has a spray phase and a wait phase. During the spray phase, the message holders hand over half of their copies with encountered nodes that have no copies. When a message holder only has one copy left, it enters the wait phase to meet the destination.
- Spray and Focus [42], where the wait phase in Spray and Wait is replaced by a single-copy routing. The message holder would forward the copy to the encountered relay, if the relay has a higher contact frequency with the destination than the message holder.
- Delegation Forwarding [8], which is modified to use a limited number of copies. Instead of holding a copy for each message transmitter, the numbers of copies allocated to the message holder and relay are proportional to their contact frequencies with the destination.
- Single-Data Multicast [43] exploits node centralities and social community structures to deliver the message. The message is forwarded to high-centrality nodes, which is active in the MOSN, and gateway nodes, which belong to multiple communities.
- Quota-Based Multicast [44] delivers the message based on quota distributions. Nodes that have more contacts with the destinations have more quotas. A probabilistic model is used to distribute the quota.

Epidemic, Spray and Wait, Spray and Focus, and Delegation Forwarding can be used for unicasts. Epidemic, Single-Data Multicast, and Quota-Based Multicast can be used for multicasts. The BubbleRap [6] is not used for comparison, since it requires an additional community information besides the contact history. It is not a fair comparison. Three classical metrics are used to measure the quality of the up-and-down routing, including the data delivery delay, ratio, and cost. The cost refers to the number of forwards in the data delivery.



(a) Data Delivery Delay. (b) Data Delivery Ratio. (c) Number of Forwards.

Fig. 11. Granularity control in the Sigcomm trace.



(a) Data Delivery Delay. (b) Data Delivery Ratio. (c) Number of Forwards.

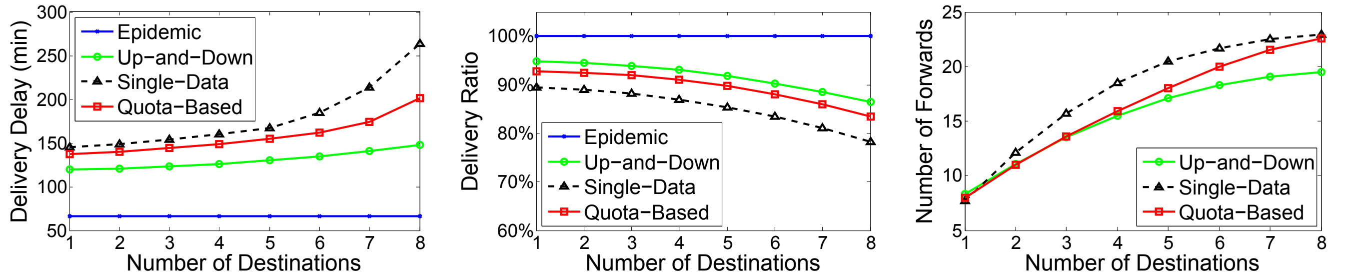
Fig. 12. Granularity control in the synthetic trace.

For further analyses, three additional metrics are proposed: the marginal delivery delay, the marginal delivery ratio, and the marginal delivery cost. These metrics measure the percentages of reduced delivery delay, increased delivery ratio, and cost brought *by using one more copy* in the download routing phase. These metrics can determine the appropriate number of copies to be used in the routing protocols.

C. Evaluation Results for Unicasts

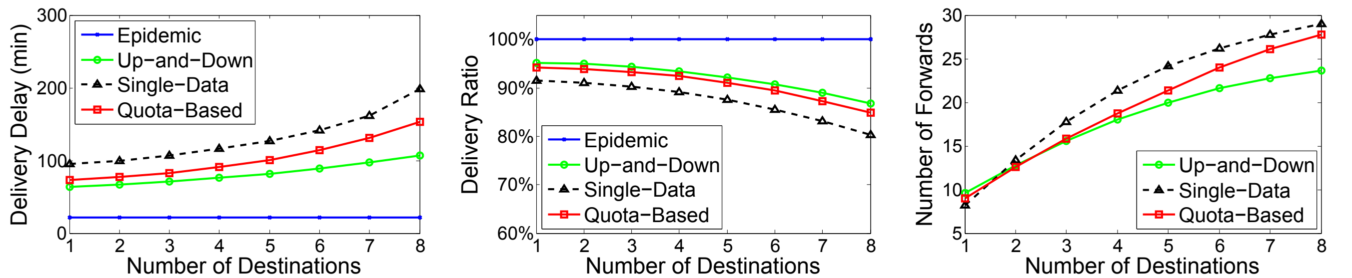
To evaluate the unicast up-and-down routing protocol, two nodes in the MOSN trace are uniform-randomly selected as the source and the destination. Algorithm 1 is used to determine the NCPH in the trace. The evaluation results are averaged over 1,000 times for the smoothness. Unicast results in the Sigcomm trace are shown in Fig. 9. With the exception of Epidemic, up-and-down routing outperforms the other routing schemes in terms of both data delivery delay and ratio. In terms of delivery cost (i.e., number of forwards), the up-and-down protocol is also good, and is only larger than the Spray and Wait. The reason for the low cost of the Spray and Wait is that it has a low delivery ratio, since all the copies are waiting for the destination rather than being forwarded to better relays. Figs. 9(d) to 9(f) show the marginal delivery delay, ratio, and cost. The number of copies follows the law of diminishing returns, i.e., decreased marginal delivery delay, ratio, and cost. We found that 4 copies are enough for the Sigcomm trace. More message copies will not bring a significantly smaller delivery delay or a significantly larger delivery ratio. Figs. 9(g) and 9(h) show the impact of the robustness ratio on the data delivery, in terms of the data delivery delay and ratio. When the robustness ratio is less than 2, the routing scheme almost fails due to the inaccurate routing hints. Meanwhile, a robustness ratio of 10 is good enough to support the up-and-down routing scheme. Therefore, Theorem 3 gives out an effective estimation. The weighted degree distribution of the Sigcomm trace is shown in Fig. 9(i). Table II presents the tradeoff between the average false positive rate and the storage consumption in the Sigcomm trace. A larger robustness ratio indicates a larger storage consumption, leading to a smaller false positive rate and a better routing performance (a smaller delivery delay and a larger delivery ratio). When the robustness ratio is 6, the average false positive rate is less than 10%, and almost half of the storage consumption is saved. Figs. 9(g) and 9(h) show that a robustness ratio of 6 obtains a good routing performance on the data delivery delay and ratio.

Unicast results in the synthetic trace are presented in Fig. 10. Since routings in this trace usually need more than two hops, the delegation forwarding performs poorly (the neighbors of the source may not recognize the destination). In this situation, the proposed up-and-down routing protocol performs much better than the traditional schemes, in terms of the delivery delay and ratio in Figs. 10(a) to 10(b). The cost of the proposed protocol is a little higher than the other schemes as a tradeoff to obtain much better delivery delay and ratio. In terms of the marginal delivery delay and ratio, routings in the Sigcomm trace and the synthetic trace have similar results. However, the delivery costs no longer have diminishing returns for all algorithms. We also obtain similar results on the robustness ratio, as shown in Figs. 10(g) and 10(h). Table III presents the tradeoff between the average false positive rate and the storage consumption in the synthetic trace. The result is similar to that in the Sigcomm trace. When the robustness ratio is 6, a competitive data delivery delay and ratio can be obtained at the expense of a small storage consumption. Finally, Fig. 10(i) shows that the weighted degree distribution in the synthetic trace also follows power-law.



(a) Data Delivery Delay. (b) Data Delivery Ratio. (c) Number of Forwards.

Fig. 13. Experimental results for multicasts in the Sigcomm trace.



(a) Data Delivery Delay. (b) Data Delivery Ratio. (c) Number of Forwards.

Fig. 14. Experimental results for multicasts in the synthetic trace.

D. Experiments on The Granularity Control

This subsection experimentally studies the impact of the granularity control for the distributed labeling scheme that computes the NCPH. Algorithm 2 is used to determine the NCPH in the trace. The other settings are the same as those in the previous subsection. The number of different hierarchical levels in the NCPH is controlled by the parameter Δ in Algorithm 2. Note that a larger Δ brings a quicker convergence, and thus the maximal peer label is smaller, i.e., fewer different hierarchical levels. Results for the granularity control in the Sigcomm trace is shown in Fig. 11. In Figs. 11(a) and 11(b), it can be seen that a larger Δ brings a larger data delivery delay and a smaller data delivery ratio. This is because a large Δ reduces the number of different hierarchical levels in the NCPH. It destroys the NCPH and degrades the routing performance. However, a large Δ can reduce the number of forwards, as shown in Fig. 11(c). This is because the root gets closer to the source and the destination, when the number of different hierarchical levels in the NCPH becomes fewer. Results for the granularity control in the synthetic trace is shown in Fig. 12, which is similar to Fig. 11. For the up-and-down routing protocol, the performance gap brought by using different message copies is smaller in the synthetic trace than in the Sigcomm trace. Experimental results for multicasts in the Sigcomm and synthetic traces show that the granularity control can reduce the number of forwards in the up-and-down routing protocol, at the expense of a larger data delivery delay and a smaller data delivery ratio.

E. Evaluation Results for Multicasts

To evaluate the multicast up-and-down routing protocol, several nodes in the MOSN trace are uniform-randomly selected as the source and the destinations. Algorithm 1 is used to determine the NCPH in the trace. The evaluation results are averaged over 1,000 times for smoothness. Multicast results in the Sigcomm and synthetic traces are shown in Figs. 13 and 14, respectively. They have similar results, in which the up-and-down routing protocol outperforms Single-Data Multicast and Quota-Based Multicast. This is because the up-and-down routing protocol has a better abstraction of the MOSN structure (i.e., the NCPH) than Single-Data Multicast and Quota-Based Multicast. Another notable point is that more destinations bring a larger data delivery delay and a smaller data delivery ratio for all the multicast protocols. This is because the routings for more destinations are harder to cooperate in the multicast, leading to performance degradations. Figs. 13(c) and 14(c) show that the number of forwards sub-linearly scales with respect to the number of destinations. This is because the up-and-down routing protocol avoids duplicated upload phases for different destinations. Experiments validate that, through utilizing the natural MOSN structure of the NCPH, the up-and-down routing protocol achieves a competitive performance on the data delivery delay and ratio, with a small cost on the prior information maintenance and a low forwarding cost.

VIII. Conclusion

This paper proposes an up-and-down routing protocol, which has an upload phase and a download phase. A message can be uploaded from the source to the network core, through iteratively forwarding the message to a relay that has a higher position in the NCPH. Bloom-filter-based routing hints are proposed to provide guidance for the message download. A subtle balance among the data delivery delay, ratio, and cost is achieved by the up-and-down routing protocol. Experimental results show that good data delivery delay and ratio are achieved, with a small cost on the prior information maintenance and a low forwarding cost.

ACKNOWLEDGEMENT

This research was supported in part by NSF grants CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, and ECCS 1231461.

REFERENCES

[1] A.-K. Pietilänen and C. Diot, "Dissemination in opportunistic social networks: the role of temporal communities," in *ACM MobiHoc*, 2012.

[2] N. Jabeur, S. Zeadally, and B. Sayed, "Mobile social networking applications," *Communications of the ACM*, 2013.

[3] B. Han, P. Hui, V. A. Kumar, M. V. Marathe, J. Shao, and A. Srinivasan, "Mobile data offloading through opportunistic communications and social participation," *IEEE Transactions on Mobile Computing*, 2012.

[4] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *ACM WWW*, 2010.

[5] SocioPatterns, "Contact networks in a primary school," 2011.

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

[7] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: an efficient routing scheme for intermittently connected mobile networks," in *WDTN*, 2005.

[8] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, "Delegation forwarding," in *ACM MobiHoc*, 2008.

[9] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: positive and negative social effects," *IEEE Communications Surveys & Tutorials*, 2013.

[10] Q. Wang and Q. Wang, "Restricted epidemic routing in multi-community delay tolerant networks," *IEEE Transactions on Mobile Computing*, 2015.

[11] K. Lee, J. Jeong, Y. Yi, H. Won, I. Rhee, and S. Chong, "Max contribution: An online approximation of optimal resource allocation in delay tolerant networks," *IEEE Transactions on Mobile Computing*, 2015.

[12] W. Gao, G. Cao, T. La Porta, and J. Han, "On exploiting transient social contact patterns for data forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, 2013.

[13] Y. Tao and X.-f. Wang, "Adaptive clustering hierarchy routing for delay tolerant network," *Journal of Central South University*, 2012.

[14] R. Chen, F. Bao, M. Chang, and J.-H. Cho, "Dynamic trust management for delay tolerant networks and its application to secure routing," *IEEE Transactions on Parallel and Distributed Systems*, 2014.

[15] C. Liu and J. Wu, "Routing in a cyclic mobispace," in *ACM MobiHoc*, 2008.

[16] ——, "Practical routing in a cyclic mobispace," *IEEE/ACM Transactions on Networking*, 2011.

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

[18] D. Niyato, P. Wang, H.-P. Tan, W. Saad, and D. I. Kim, "Cooperation in delay-tolerant networks with wireless energy transfer: Performance analysis and optimization," *IEEE Transactions on Vehicular Technology*, 2015.

[19] L. Wang, H. Tang, and M. Cierny, "Device-to-device link admission policy based on social interaction information," *IEEE Transactions on Vehicular Technology*, 2015.

[20] F. Li, S. Chen, M. Huang, Z. Yin, C. Zhang, and Y. Wang, "Reliable topology design in time-evolving delay-tolerant networks with unreliable links," *IEEE Transactions on Mobile Computing*, 2015.

[21] W. Peng, G. Dong, K. Yang, and J. Su, "A random road network model and its effects on topological characteristics of mobile delay-tolerant networks," *IEEE Transactions on Mobile Computing*, 2014.

[22] K. Wei, D. Zeng, S. Guo, and K. Xu, "On social delay-tolerant networking: aggregation, tie detection, and routing," *IEEE Transactions on Parallel and Distributed Systems*, 2014.

[23] K. Wei, S. Guo, D. Zeng, K. Xu, and K. Li, "Exploiting small world properties for message forwarding in delay tolerant networks," *IEEE Transactions on Computers*, 2015.

[24] A. Balasubramanian, B. Levine, and A. Venkataramani, "Replication routing in dtns: A resource allocation approach," *IEEE/ACM Transactions on Networking*, 2010.

[25] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2009-05-29)," 2009.

[26] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, 2009.

[27] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti, "CRAWDAD trace set st_andrews/sassy/mobile (v. 2011-06-03)," 2011.

[28] A.-K. Pietilainen and C. Diot, "CRAWDAD data set thlab/sigc- omm2009 (v. 2012-07-15)," 2012.

[29] M. Newman, *Networks: An Introduction*. New York, NY, USA: Oxford University Press, Inc., 2010.

[30] A. W.-C. Fu, H. Wu, J. Cheng, and R. C.-W. Wong, "Is-label: an independent-set based labeling scheme for point-to-point distance querying," *Proceedings of the VLDB Endowment*, 2013.

[31] A. Broder and M. Mitzenmacher, "Network applications of Bloom filters: A survey," in *Internet Mathematics*, 2002.

[32] M. Sarela, C. E. Rothenberg, T. Aura, A. Zahemszky, P. Nikander, and J. Ott, "Forwarding anomalies in bloom filter-based multicast," in *IEEE INFOCOM*, 2011.

[33] H. Chen, H. Jin, L. Chen, Y. Liu, and L. M. Ni, "Optimizing bloom filter settings in peer-to-peer multikeyword searching," *IEEE Transactions on Knowledge and Data Engineering*, 2012.

[34] O. Rottenstreich, Y. Kanizo, and I. Keslassy, "The variable-increment counting bloom filter," in *IEEE INFOCOM*, 2012.

[35] D. Li, H. Cui, Y. Hu, Y. Xia, and X. Wang, "Scalable data center multicast using multi-class bloom filter," in *IEEE ICNP*, 2011.

[36] H. Zheng and J. Wu, "Up-and-down routing in mobile opportunistic social networks with bloom-filter-based hints," in *IEEE/ACM IWQoS*, 2014.

[37] A.-L. Barabási, "Scale-Free Networks: A Decade and Beyond," *Science*, 2009.

[38] A. Clauset, "Fitting a power-law distribution," Downloaded from http://tuvalu.santafe.edu/ãaronc/powerlaws/, 2012.

[39] J. Yang and J. Leskovec, "Overlapping communities explain core-periphery organization of networks," *Proceedings of the IEEE*, 2014.

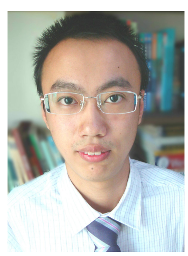
[40] P. Csermely, A. London, L.-Y. Wu, and B. Uzzi, "Structure and dynamics of core/periphery networks," *Journal of Complex Networks*, 2013.

[41] M. P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha, "Core-periphery structure in networks," *SIAM Journal on Applied mathematics*, 2014.

[42] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility," in *Prceedings of IEEE PerCom Workshops*, 2007.

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

[44] S.-C. Lo, N.-W. Luo, J.-S. Gao, and C.-C. Tseng, "Quota-based multicast routing in delay-tolerant networks," *Wireless Personal Communications*, 2014.



Huanyang Zheng received his B.Eng. degree in Telecommunication Engineering from Beijing University of Posts and Telecommunications, China, in 2012. He is currently a Ph.D. candidate in the Department of Computer and Information Sciences, Temple University, USA. His research focuses on wireless and mobile networks, social networks and structures, and cloud systems.

Jie Wu is the Associate Vice Provost for International Affairs at Temple University. He also serves as Director of Center for Networked Computing and Laura H. Carnell professor in the Department of Computer and Information Sciences. Prior to joining Temple University, he was a program director at the National Science Foundation and was a 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 Service Computing and the Journal of Parallel and Distributed Computing. Dr. Wu was general cochair/chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, and ACM MobiHoc 2014, as well as program co-chair for IEEE INFOCOM 2011 and 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.