Proactively Placing Static Relays With Social-Link Awareness in Mobile Social Networks

Xiaomei Zhang, Member, IEEE, and Guohong Cao, Fellow, IEEE

Abstract—Mobile social networks have been exploited for data forwarding due to its low cost and better robustness. Existing data forwarding strategies in mobile social networks rely on the pairwise contacts among mobile users. However, these pairwise contacts only provide limited forwarding capabilities and most data fail to be delivered before the expiration time. In this paper, we improve the performance of data forwarding by proactively placing low-cost static relays to increase the opportunistic contacts among the mobile nodes. Based on this idea, an important question is where to place the static relays in the network to best facilitate data forwarding among nodes. To answer this question, we first analyze four real-world datasets of mobile social networks, and identify that data forwarding only appears in a small group of social links. Then, we formulize the problem of static relay placement as an optimization problem, and propose a heuristic based solution to improve the data forwarding performance along these social links. Considering the fact that the social links may evolve over time, we also propose an efficient relay replacement algorithm which replaces the outdated relays with new relays to further improve the data forwarding performance. Evaluation results show that the proposed relay based solutions can significantly improve the performance of data forwarding in mobile social networks.

Index Terms—Mobile social networks, static relays, data forwarding.

I. INTRODUCTION

MOBILE social networks [1]–[3] consist of human-carried mobile devices\(^1\) which opportunistically connect with each other through peer-to-peer wireless communications such as Bluetooth or WiFi-direct. The major advantage of mobile social network is that it does not rely on any infrastructure, and thus it has low cost and better robustness, and has potential applications in battlefield, disaster recovery, and underdeveloped countries. Since mobile devices are usually carried by human beings, whose mobilities are usually dynamic and unpredictable, achieving efficient data forwarding in mobile social networks is extremely challenging. To address this problem, “carry-and-forward” is used, i.e., the mobile nodes physically “carry” the data and “forward” the data upon contacts with other nodes, until reaching the destination. Epidemic [4] is one of the early data forwarding algorithm, where data is flooded upon contacts with other nodes. Later solutions attempt to reduce the number of data copies created by Epidemic, and these strategies are known as controlled flooding [5]. Among them, some strategies limit the number of hops that data can traverse in the network, e.g., Spray-and-Wait [6]. In other solutions, data carrier only forwards data to another node with higher forwarding metric, measuring node’s capability of forwarding data to the destination, e.g. PROPHET [7] and BubbleRap [8].

Even though many approaches have been proposed for data forwarding in mobile social networks, most of them can only achieve low data delivery ratio (i.e., the percentage of successfully delivered data items) within the time constraint of the data. Even with the Epidemic approach [4], many data items still fail to be delivered. For example, Fig. 1 shows a network with five nodes, and the future contact times between nodes are shown inside the grey rounded rectangles on the edges. Assume there is one data item, Data 1, generated by A at 12 pm with destination F. The data item is forwarded based on flooding (Epidemic) and expires at 4 pm. After Data 1 is generated at 12 pm, node A forwards it to node B at 1 pm, which then forwards it to node C at 1:20 pm. When Data 1 expires at 4 pm, three nodes A, B, C have the data, but the destination F still hasn’t received the data, because there is no contact from A, B, C to E (and F) before 4 pm, and node E (and F) cannot receive the data before the expiration time. Although the example only shows one data item, many similar cases can occur, and hence many data items cannot be delivered on time. This is because the existing data forwarding strategies, including Epidemic, only rely on the opportunistic contacts between the mobile nodes to forward data, which can only provide limited forwarding capabilities.

To improve the performance of data forwarding in mobile social networks, we propose to proactively place static relays

\(^1\)Also referred to as mobile nodes or mobile users in the rest of the paper.
We further study the evolution of social links, i.e., whether at 2 pm, which finally delivers the data to destination \( F \). These static relays can be low-cost wireless devices with some storage capability, which can participate in the carry-and-forward process like normal mobile nodes. These relays can be deployed near buildings with power supply for long term access, or periodically replaced for short term access. For example, for disaster recovery or military operation, some static relays can be placed at some places to increase the contact opportunities of the mobile nodes. If the operation time of these static relays is short, power is not an issue; otherwise, they are either recharged or periodically replaced.

To realize the idea of using static relays to improve the performance of data forwarding, we have to answer an important question, i.e., where to place the static relays in the network to best facilitate data forwarding among nodes. It is challenging to address this question, especially considering nodes may scatter in a wide geographical area. In this paper, by identifying that most data are sent among people with social connections (or social links) such as relatives, friends or classmates, we focus on placing static relays to help data forwarding along these social links. In most existing research \([4], [8]\), data forwarding is between two randomly selected nodes. However, in mobile social networks, this is not always true. Nodes with social links are more likely to communicate with each other, whereas two nodes without any social connection have less chance to communicate with each other. Although it is hard to improve the performance of data forwarding for two randomly selected nodes, by strategically placing some static nodes, the performance of data forwarding for nodes with social links can be significantly improved. Specifically, in this paper, we propose novel solutions to place static relays to maintain efficient data forwarding along given social links. The main contributions of the paper are summarized as follows:

- We formalize static relay placement as an optimization problem, and propose a heuristic based solution to improve the data forwarding performance along social links. Specifically, a social-link betweenness metric is defined to quantify the importance of static relays on improving the data delivery ratio along the social links, and then static relays with higher social-link betweenness are placed.
- We further study the evolution of social links, i.e., whether traffic loads on social links change over time, through dataset analysis. An efficient relay replacement algorithm is proposed to replace the existing outdated relays to further improve data forwarding performance.
- Evaluations based on four real-world datasets show that the proposed static relay based solutions can significantly enhance the data forwarding performance.

The rest of the paper is organized as follows. Section II reviews related work and Section III presents the preliminaries including the dataset analysis and the data forwarding strategy. We present our solutions for static relay placement and relay replacement in Section IV. Section V presents the performance evaluation results and Section VI concludes the paper.

II. RELATED WORK

Because of the intermittent network connectivity in mobile social networks, data are forwarded based on “carry-and-forward”. The simplest and earliest approach for "carry-and-forward" is Epidemic \([4]\), in which data are forwarded based on flooding. Due to the high overhead (i.e., number of data copies) associated with Epidemic, later solutions attempt to reduce the overhead by controlling the data forwarding process and these strategies are known as controlled flooding. There are two categories of controlled flooding. The first category limits the number of hops data can traverse in the network. For example, spray-and-wait \([6]\) limits the number of hops to two. The second category is known as compare-and-forward, in which the data carrier only forwards data to nodes with higher forwarding metric. The forwarding metric measures node’s capability of forwarding data to the destination, which is usually determined based on node’s contact probability with the destination, such as PROPHET \([7]\). Later, social-aware metrics, e.g., centrality \([8]\), and communities \([9]\) are further used as forwarding metrics. More recently, core-based solution \([10]\), security-based solutions \([11], [12]\), incentive-based solutions \([3], [13]\) and solutions that consider energy consumption \([14], [15]\) and context awareness \([16]\) are also proposed to facilitate data forwarding in mobile social networks. In addition, Han et al. \([17]\) proposed a single-copy multi-path (SCMP) transmission strategy that satisfies the delay requirement and, at the same time, minimizes communication cost (using only single data copy). Different from these existing works where data forwarding is between two randomly selected nodes, we focus on data forwarding among nodes with social links. This is because nodes with social links are more likely to communicate with each other, whereas two nodes without any social links have less chance to communicate with each other. Although it is hard to improve the performance of data forwarding for
two randomly selected nodes, by strategically placing some static relays, the performance of data forwarding for nodes with social links can be significantly improved.

There is some research relying on third-party facilities to support data forwarding. For example, in message ferrying based solutions [18]–[20], a set of special mobile nodes called message ferries are responsible for carrying data for nodes in the network. Their objective is to design the ferry routes so that the performance can be improved. However, message ferries have to follow some moving path which may not always be possible, and the cost of deploying and maintaining these message ferries is high. In our work, the added relays are static, which have much lower cost to deploy and maintain.

In mobile social networks, caching techniques [21] have been used to improve the performance of data forwarding. However, caching is limited to those moving nodes, and their benefit is limited compared to placing extra static nodes. Other research [22], [23] tries to combine mobile social networks with static facilities, e.g., WiFi access points or cellular base stations, but their objective is to utilize mobile social networks to offload traffic which is different from the proposed work. To the best of our knowledge, our work is the first to utilize low-cost static relays to improve the performance of data forwarding along given social links.

III. PRELIMINARIES

In this section, we first describe how to place static relays in mobile social network. Then, we introduce four real-world datasets on mobile social networks, and present dataset analysis results on social links and define the network model. We finally present a data forwarding strategy that incorporates static relays.

A. Placement of Static Relays

In this paper we focus on intentionally placing new static relays into a mobile social network that can best facilitate the communications among mobile nodes in the network. Nodes in mobile social network usually move within a geographical area. Since the geographical area can be large, it is challenging to decide where to place the static relays. To make the problem feasible, we first set aside a set of candidate static relays throughout the area, and then select the best static relays from them.

In most real-world scenarios, the static relays can only be placed at limited locations, e.g., limited by the hardware restriction, the environmental constraint, or even the construction code. In these cases, the locations of candidate static relays are usually predetermined and fixed. For example, as shown in Fig. 3(a), if the area includes several buildings, the static relays may be predetermined to be installed around the building or in the center point of the building, as noted by the white dots.

In other scenarios where there is no location constraint, the candidate static relays’ locations can be flexibly chosen. To cover the entire geographical area, the candidate static relays are evenly scattered throughout the area with equal intervals, denoted by the white dots in Fig. 3(b). The length of intervals can be set flexibly according to the node density in the network.

The static relays will be selected from these candidates. Specifically, with the locations of the candidate static relays, we are able to analyze the contact patterns between mobile nodes and candidate static relays (a contact occurs when a mobile node moves into the transmission range of a candidate static relay). Based on the contact patterns, an optimal set of static relays can be selected that best facilitates the communication among mobile nodes. The specific strategy for placing static relays will be presented in next section (Section IV).

B. A Summary of Datasets

In this paper, we study four datasets of mobile social networks: Social Evolution [24], Infocom [25], UCSD [26] and San Francisco Taxi [27].

All these datasets have a group of mobile nodes and record the contacts between mobile users that are in proximity. In addition to mobile nodes, the first three datasets also record a set of static nodes in fixed locations around the network, and these nodes will be used as the candidate static relays with location fixed (corresponding to the case with location constraint in Fig. 3(a)). The last dataset does not have static nodes recorded, but it keeps track of the location of each mobile node. In this dataset, the candidate static relays will be flexibly set to the locations that are evenly distributed in the whole area (corresponding to the case without location constraint in Fig. 3(b)).

A summary of these datasets is shown in Table I. To distinguish between different types of contacts, the contacts between mobile nodes are referred to as peer contacts, and the contacts between mobile users and candidate static relays are referred to as static contacts.

1) Social Evolution: It records the contacts among students using mobile devices in an undergraduate dormitory. The mobile devices periodically detect their peers via Bluetooth interfaces. A peer contact is recorded when two mobile devices move into the detection range of each other. In addition to the peer contacts, Social Evolution also records mobile devices’ detection of nearby WiFi access points. In our experiments, these access points are considered as candidate static relays. When a mobile
TABLE I
DATASET SUMMARY

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Social Evolution</th>
<th>Infocom</th>
<th>UCSD</th>
<th>San Francisco Taxi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device</td>
<td>Cell Phone</td>
<td>iMote</td>
<td>PDA</td>
<td>Vehicles</td>
</tr>
<tr>
<td>Network type</td>
<td>Bluetooth</td>
<td>Bluetooth</td>
<td>WiFi</td>
<td>GPS</td>
</tr>
<tr>
<td>Number of mobile devices</td>
<td>80</td>
<td>78</td>
<td>275</td>
<td>535</td>
</tr>
<tr>
<td>Number of candidate static relays</td>
<td>108</td>
<td>20</td>
<td>162</td>
<td>379</td>
</tr>
<tr>
<td>Number of peer contacts</td>
<td>200,841</td>
<td>113,977</td>
<td>11,713</td>
<td>3,435,130</td>
</tr>
<tr>
<td>Number of static contacts</td>
<td>477,945</td>
<td>27,237</td>
<td>97,911</td>
<td>670,562</td>
</tr>
<tr>
<td>Start Date</td>
<td>2008-10-01</td>
<td>2006-04-23</td>
<td>2002-09-22</td>
<td>2008-05-17</td>
</tr>
<tr>
<td>Durations (days)</td>
<td>243</td>
<td>4</td>
<td>776</td>
<td>30</td>
</tr>
<tr>
<td>Granularity (secs)</td>
<td>300</td>
<td>120</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>

Fig. 4. Social links in the Social Evolution dataset, where a blue marker represents a social link.

device detects an access point, a static contact is recorded. Moreover, Social Evolution also records the SMS history and phone call history between participants, which are good indicators of friendship and potential social links among users.

2) Infocom: It was collected in a conference environment by recording the contacts between conference attendants carrying imotes. Similar to Social Evolution, peer contacts are detected via Bluetooth. In addition to mobile users, the imotes are installed at several static locations, and these imotes are considered as candidate static relays.

3) UCSD: It was collected in a campus environment, where the devices are WiFi-enabled PDAs. These devices search for nearby WiFi Access Points (APs), and a peer contact is recorded when two devices detect at least one common AP at the same time. The APs are regarded as candidate static relays, and when a mobile device detects one AP, a static contact is recorded.

4) San Francisco Taxi (Taxi): The dataset records mobility traces of taxi cabs in San Francisco Bay Area by tracking their locations with GPS. As mentioned, the candidate static relays are put at fixed locations with equal intervals (set to 500 meters) that scatter the whole area (as in Fig. 3(b)). Excluding the locations where nodes never pass by, there are a total of 380 candidate static relays throughout the network. Since the original dataset does not record the peer contacts, in our experiment, a peer contact is recorded when two nodes move within 50 meters of each other. As long as a node moves into 100 meters of a candidate static relay, a static contact is recorded.

One thing to note is that, different to the original datasets in which the static nodes (e.g., WiFi access points and static imotes in first three datasets) are already installed and can be used to relay data, in our experiments, we assume that there are no relays installed in these locations yet, and they are only regarded as candidate locations where relays may be installed. Our objective is to select a limited number of optimal static relays from those locations.

C. Social Link

A social link exists between a pair of users with social connection, e.g., friends, classmates or teammates in a sport team. Because of the social connection, the two users are more likely to initiate data or message forwarding between them, e.g., SMS or MMS. In this subsection, we study the characteristics of social links based on the Social Evolution dataset. SMS or phone calls are indicators of social links.

1) Number of Social Links: We first conduct an experiment to record the number of social links among all pairs of users. The social links based on SMS and phone calls are studied and shown in Fig. 4(a) and (b), respectively. The Figures not only show the user pairs with social links, but also show user pairs with physical peer contacts (move into proximity with each other). As shown in the figures, even though most user pairs have peer contacts, only a limited number of them have social links, either for SMS-based or phone call-based social links. Among all 80 users in the dataset, there are 4994 user pairs with peer contacts, but only 64 user pairs have SMS-based social links and 510 have call-based social links. This indicates that social links only exist among a limited number of user pairs, and social links may not necessarily be associated with peer contacts.

2) Traffic Load on Social Links: In addition to the number of social links, the traffic load on social links (i.e., the number of
data items forwarded on the links) is also studied. Specifically, in the experiment, we record the number of SMS messages sent on each SMS-based social link, and the number of phone calls made on each call-based social link. The complementary cumulative distribution functions (complementary CDFs) of the traffic loads are shown in Fig. 5 in log-log scale. As can be seen, for SMS-based social links and the phone call-based social links, the load distributions both follow power-law distributions (indicated by straight lines in the log-log scale). The power-law distributions indicate that the traffic loads on different social links are different. There exist a small group of social links with heavy loads, and such social links have strong social connections.

Evolution of traffic loads on social links: The traffic loads on social links may change over time because of users’ dynamic social behaviors. To test this, we then conduct an experiment to track the traffic loads on social links for different time periods. Fig. 6 shows the change of traffic load on one social link (link between user 1 and user 61) over 25 weeks, based on both SMS and phone call respectively. From the figure, we can see a clear evolution of traffic loads on the link, for both SMS and phone call. Evolutions are also detected from other social links in our experiment, but not shown here to save space.

D. Network Model

The network studied in this paper is a mobile social network $G(V, E, R, E')$, where $V$ is the set of mobile users, $E$ is the set of edges (i.e., contact processes) between mobile users in $V$, $R$ is a set of candidate static relays in the network, and $E'$ is the set of edges (i.e., contact processes) between mobile users in $V$ and relays in $R$. More specifically, the edge $e_{u,v}^c$ in $E'$ represents the pairwise contact process between mobile user $u$ and candidate static relay $r$ ($u \in V$ and $r \in R$).

The inter-contact time for $e_{u,v}^c$, i.e., between two mobile users $u, v$, has been experimentally validated in [28] to follow exponential distribution with rate parameter $\lambda_{u,v}$, thus the contact process between $u, v$ is a Poisson process. Since the datasets used for their validation are different from the datasets used in this paper, the validation on these four datasets is still needed. Also, the distribution of the inter-contact time for $e_{u,r}^c$, i.e., between mobile user $u$ and candidate static relay $r$, has not been validated.

First, we use three hypothesis testing methods to perform a goodness-of-fit testing for the hypothesis “the inter-contact time for contact processes in $E$ follows exponential distribution”. The three testing methods are Chi-square test [29], Kolmogorov-Smirnov test [30], and Lilliefors test [31]. Table II shows the acceptance ratio of the testing on all pairs of mobile users in the four datasets. As can be seen from Table II, high acceptance ratio is achieved for all four datasets and three testing methods. This illustrates that the inter-contact time between mobile users follows exponential distributions.

Then, we perform hypothesis testing for the hypothesis “the inter-contact time for contact processes in $E'$ follows exponential distribution”, and Table III shows the acceptance ratio of the testing for the four datasets based on the three testing methods. As can be seen, high acceptance ratio is achieved for all four datasets and three testing methods. This illustrates that the inter-contact time between mobile users and candidate static relays also follows exponential distribution.
TABLE III
ACCETANCE RATIO (%) FOR TESTING EXPONENTIAL DISTRIBUTION OF
INTER-CONTACT TIME BETWEEN MOBILE USER AND
CANDIDATE STATIC RELAY

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SE</th>
<th>Infocom</th>
<th>UCSD</th>
<th>Taxi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chi-Square Test</td>
<td>98.42</td>
<td>93.04</td>
<td>100</td>
<td>88.59</td>
</tr>
<tr>
<td>Kolmogorov-Smirnov Test</td>
<td>100</td>
<td>94.78</td>
<td>97.59</td>
<td>93.96</td>
</tr>
<tr>
<td>Lilliefors test</td>
<td>100</td>
<td>97.39</td>
<td>99.40</td>
<td>96.64</td>
</tr>
</tbody>
</table>

![Image of Fig. 7 showing the double-phase spray-and-wait.

E. Double-Phase Spray-and-Wait Data Forwarding

To proactively place static relays to improve the performance of data forwarding, we need to have a data forwarding strategy that incorporates static relays. In this paper, we utilize a Double-Phase Spray-and-Wait (DPSW) data forwarding strategy which extends the conventional Spray-and-Wait [6] strategy to incorporate static relays in data forwarding. Spray-and-wait is a two-hop strategy. In the first hop, the source sprays the data to a group of mobile nodes. In the second hop, the mobile nodes wait until contacting the destination and then deliver the data. In this paper, we cannot apply spray-and-wait directly due to its limitation. Spray-and-wait mainly relies on the physical movement of the mobile node that receives data in the first hop to deliver the data to the destination. However, this may not be possible in a large mobile social network where each node only has limited moving range. To be worse, if spray-and-wait is used in network with static relays and the node that receives data in the first hop is a static relay, it is harder for the destination to receive data in the second hop.

With Double-Phase Spray-and-Wait, the spray-and-wait process is executed in two phases. In the first phase, the spray-and-wait is used to forward data from the source to a static relay, and in the second phase, data is forwarded from the static relay to the destination, as shown in Fig. 7. In Fig. 7, the middle node is the static relay (R), where the data item is buffered before being forwarded out. The two nodes besides the static relay are referred to as the “post nodes” (P1 and P2), which send data to and fetch data from the static relay. As a result, there are a maximum of four hops in the entire data forwarding process from the source to destination, with the first hop from source S to P1, second hop from P1 to R, third hop from R to P2, and fourth hop from P2 to the destination D. Note that a data item does not have to go through all four hops before reaching the destination. It can bypass one or both of the post nodes P1, P2, resulting in two or three hops. It is also possible that the data forwarding does not go through the static relay R, i.e., post node P1 reaches destination D directly, and then it becomes the same as the traditional spray-and-wait with a maximum of two hops.

Different from the original spray-and-wait strategy, we do not set a limit to the number of sprayed data copies. Considering the limited forwarding opportunity due to the intermittent connectivity of mobile social networks, further limiting the number of sprayed data copies may miss some data forwarding opportunities. In DPSW, spray-and-wait is executed in two phases. Spray-and-wait can be executed in more than two phases, but doing so may introduce unnecessary network overhead.

To show how DPSW works, we still use the motivation example discussed in Fig. 2. With DPSW, Data 1 is forwarded from source A to B, which serves as the first post node and then forwards data 1 to relay R. After R receives the data, it forwards the data to E, which serves as the second post node and finally delivers the data to destination F. Note that with DPSW, B does not forward data to other mobile nodes, e.g., C, even though B encounters C before the data item expires. By doing so, many extra data copies can be avoided.

IV. PROACTIVE PLACEMENT OF STATIC RELAYS

In this section, we formalize the problem of proactive placement of static relays and propose a heuristic based solution.

A. Problem Formalization

Since most data forwarding occurs between nodes with social links, only the communications among a limited number of social links need to be actively maintained. The set of maintained social links is denoted as,

\[ L = \{l_1, l_2, ..., l_{|L|}\} \]

\[ = \{(s_1, d_1, \epsilon_1), (s_2, d_2, \epsilon_2), ..., (s_{|L|}, d_{|L|}, \epsilon_{|L|})\} \]  (1)

where \( s_j, d_j \) and \( \epsilon_j \) indicate the source, destination and traffic load of link \( l_j \), \( 1 \leq j \leq |L| \), and |L| is the number of actively maintained links. Load \( \epsilon_j \) refers to the average number of data items created on link \( l_j \) per unit time (e.g., per day, or per hour). Note that links included in \( L \) are “one-way” links, i.e., by maintaining \( l_j \), only the forwarding from \( s_j \) to \( d_j \) is maintained, but not vice versa. If two-way forwarding needs to be maintained between two nodes, both social links on both directions should be added to \( L \).

Different social links may have different delay requirements on data forwarding. For example, the communications between two command centers in a military environment need to be forwarded as soon as possible, but a picture sent between friends may tolerate more delay. Therefore, we assign different time constraints to the social links in \( L \), i.e., each social link \( l_j \) has a delay constraint of \( \tau_j \).

Relays are not arbitrarily placed in the network. As what we mentioned in last section, static relays are selected from a set of pre-determined candidate static relays. The set of candidate static relays is denoted as \( R = \{r_1, r_2, ..., r_{|R|}\} \). Due to deploying and maintaining cost constraints, there are only \( \theta \) static relays to be placed, with \( \theta \leq |R| \). The problem of static relay placement is defined as follows:

**Problem 1:** In a mobile social network, given
- a set of social links \( L \) that need to be maintained, and
• a set of candidate static relays $R$ where static relays can be placed,

the problem of static relay placement is how to select an optimal subset $R^* (|R^*| \leq \theta)$ of nodes from $R$, where static relays are placed, so that the data delivery ratio along the social links in $L$ is maximized.

To actively maintain a set of social links $L$, the objective of this optimization problem is to maximize the expected data delivery ratio. If the probability of successfully forwarding data along link $l_j$ is $p_j^{R^*}$ by placing static relays in $R^*$, the expected number of delivered data items per unit time on all social links is $\sum_{j=1}^{L} \epsilon_j \ast p_j^{R^*}$, where $\epsilon_j$ is the load on link $l_j$. The optimization problem for maximizing the expected delivery ratio can be formalized as follows:

$$\max \frac{\sum_{j=1}^{L} \epsilon_j \ast p_j^{R^*}}{\sum_{j=1}^{L} \epsilon_j}$$ (2)

subject to $|R^*| \leq \theta$, $R^* \subseteq R$ (3)

where (3) limits the number of selected relays $|R^*|$ by $\theta$.

To solve this optimization problem, it is necessary to calculate the probability $p_j^{R^*}$ for a given $R^*$, but unfortunately, it is intractable to calculate $p_j^{R^*}$. This is because calculating $p_j^{R^*}$ requires the calculation of forwarding probabilities on all possible paths, which are then aggregated to compute the overall forwarding probability. The aggregation requires the forwarding paths to be independent, but in this case, the paths can share hops. For example, in a small network shown in Fig. 8, there are three paths from $s_1$ to $d_1$, $p_1 : s_1 \rightarrow u_1 \rightarrow r_1 \rightarrow d_1$, $p_2 : s_1 \rightarrow u_2 \rightarrow r_1 \rightarrow d_1$, and $p_3 : s_1 \rightarrow u_2 \rightarrow r_2 \rightarrow d_1$. It is easy to find that $p_1$, $p_2$, and $p_3$ are overlapped on the first hop, and $p_2$, $p_3$ are overlapped on the last hop. Such overlaps among multiple paths make it infeasible to calculate the cumulative data forwarding probability between $s_1$ and $d_1$.

As an attempt to address this problem, an approximation approach is to select a maximum number of independent paths from all forwarding paths and let them forward data, and then the forwarding probability on these independent paths can be calculated. However, finding the maximum number of independent paths for a pair of nodes is also proven to be NP-hard in graph theory [32], [33]. Therefore, computing forwarding capability $p_j^{R^*}$ for link $l_j$ is intractable, and it is infeasible to find a closed-form solution to the optimization problem.

**B. Heuristic Based Algorithm for Relay Placement**

Since the optimization problem is hard to solve, we propose a heuristic based algorithm. The basic idea is to greedily select relays that can mostly increase the delivery ratio. To achieve this goal, we first define a metric to quantify the importance of relays in increasing the data delivery ratio along the social links. Then, a greedy algorithm is proposed to select relays based on the defined metric.

1) Social-Link Betweenness: For a relay node $r_j$, we use a centrality metric called social-link betweenness (SL-betweenness), to quantify its importance, which is the number of data items that can be delivered along all social links per unit time through $r_j$. The formal definition is shown as follows:

**Definition 1:** Let $\epsilon'_j$ denote the current load (i.e., data items sent per unit time) along social link $l_j$. The SL-betweenness ($r_j$) of static relay $r_j$ is the data items that $r_j$ can help to deliver along all social links per unit time:

$$SL-betweenness(r_j) = \sum_{j=1}^{L} \epsilon'_j \ast p'_j$$ (4)

where $p'_j$ is the probability that the data on social link $l_j$ is delivered by going through $r_j$ within the time constraint $\tau_j$.

Now let us consider how to compute $p'_j$. Let $T'_j$ denote the data forwarding time by going through $r_j$, we have

$$p'_j = P(T'_j \leq \tau_j)$$ (5)

The forwarding time $T'_j$ can be determined based on the forwarding paths shown in Fig. 9. If the data forwarding process goes through $r_j$, the data forwarding process can be divided into two phases. Let $T_1$ denote the forwarding time on the first phase from $s_j$ to $r_j$, and let $T_2$ denote the forwarding time on the second phase from $r_j$ to $d_j$. Then, we have

$$p'_j = P(T'_j = P(T_1 + T_2 \leq \tau_j) = \int_0^{\tau_j} f_1(t) \otimes f_2(t) dt (6)$$

where $f_1(t)$ and $f_2(t)$ are the probability density functions of $T_1$ and $T_2$, and $f_1(t) \otimes f_2(t)$ represents the convolution between $f_1(t)$ and $f_2(t)$. The convolution is hard to compute, so we find a lower bound as a tight approximation to $p'_j$.

Assume there is a small constant $\bar{\tau} < \tau_j$, which is used to represent the length of a short time step. By breaking $\tau_j$ into time steps of length $\bar{\tau}$, the following theorem gives a lower bound to $p'_j$:
Theorem 1: For any fixed \( \bar{t} \), \( 0 < \bar{t} < \tau_j \), we have

\[
p_j^\tau \geq \sum_{i=0}^{\lfloor \frac{\tau_j}{\bar{t}} \rfloor - 1} P(i\bar{t} \leq T_1 
\leq (i + 1)\bar{t})P(T_2 \leq \tau_j - (i + 1)\bar{t})
\]

(7)

where

\[P(i\bar{t} \leq T_1 \leq (i + 1)\bar{t}) = P(T_1 \leq (i + 1)\bar{t}) - P(T_1 \leq i\bar{t})\]

(8)

and \( p_j^\tau \) is equal to the lower bound as \( \bar{t} \to 0 \).

Proof: From (6), we have

\[
p_j^\tau = \int_0^{\tau_j} f_1(t) \otimes f_2(t) dt
\]

\[
= \int_0^{\tau_j} f_1(t) \left[ \int_0^{\tau_j - t} f_2(t') dt' \right] dt
\]

\[
= \int_0^{\tau_j} f_1(t) [P(T_2 \leq \tau_j - t)] dt
\]

\[
= \sum_{i=0}^{\lfloor \frac{\tau_j}{\bar{t}} \rfloor - 1} \int_i^{(i+1)\bar{t}} f_1(t) [P(T_2 \leq \tau_j - t)] dt
\]

\[
+ \int_{\lfloor \frac{\tau_j}{\bar{t}} \rfloor \bar{t}}^{\tau_j} f_1(t) [P(T_2 \leq \tau_j - t)] dt
\]

(9)

From (9) to (10), the time period \([0, \tau_j]\) is divided into time steps of length \( \bar{t} \), i.e., \([i\bar{t}, (i + 1)\bar{t}]\), with \( 0 \leq i \leq \lfloor \frac{\tau_j}{\bar{t}} \rfloor - 1 \), and the final time step is shorter than \( \bar{t} \), i.e., \([\lfloor \frac{\tau_j}{\bar{t}} \rfloor \bar{t}, \tau_j]\).

In (10), for \( P(T_2 \leq \tau_j - t) \) with \( i\bar{t} \leq t \leq (i + 1)\bar{t} \) in the first part, it has the following lower bound,

\[
P(T_2 \leq \tau_j - t) \geq P(T_2 \leq \tau_j - (i + 1)\bar{t})
\]

(11)

This is because the cumulative distribution function (CDF) \( P(T_2 \leq t) \) is a monotonically increasing function with \( t \). With \( t \leq (i + 1)\bar{t} \), we have \( \tau_j - t \geq \tau_j - (i + 1)\bar{t} \), and we get the above lower bound (11). Similarly, for \( P(T_2 \leq \tau_j - t) \) with \([\lfloor \frac{\tau_j}{\bar{t}} \rfloor \bar{t}, \tau_j]\) in the second part, we have,

\[
P(T_2 \leq \tau_j - t) \leq P(T_2 \leq \tau_j - \tau_j) = 0
\]

(12)

Substituting (11), (12) to (10), we have

\[
p_j^\tau \geq \sum_{i=0}^{\lfloor \frac{\tau_j}{\bar{t}} \rfloor - 1} \int_i^{(i+1)\bar{t}} f_1(t) [P(T_2 \leq \tau_j - (i + 1)\bar{t})] dt
\]

\[
= \sum_{i=0}^{\lfloor \frac{\tau_j}{\bar{t}} \rfloor - 1} \int_i^{(i+1)\bar{t}} f_1(t) [P(T_2 \leq \tau_j - (i + 1)\bar{t})] dt
\]

\[
= \sum_{i=0}^{\lfloor \frac{\tau_j}{\bar{t}} \rfloor - 1} P(i\bar{t} \leq T_1 \leq (i + 1)\bar{t})P(T_2 \leq \tau_j - (i + 1)\bar{t})
\]

The equation signs in (11), (12) fulfill as \( \bar{t} \to 0 \), and therefore \( p_j^\tau \) is equal to the lower bound as \( \bar{t} \to 0 \). 

From the above theorem, as \( \bar{t} \) becomes smaller, \( p_j^\tau \) becomes closer to the lower bound, but this also generates more computation overhead because more time steps need to be added when calculating (7). In later experiments, we set \( \bar{t} = 0.5 \) hour, which is small enough considering the time constraints in our datasets, usually several hours, but also does not generate too much computation overhead.

Based on Theorem 1, as long as \( P(T_1 \leq t) \) and \( P(T_2 \leq t) \), i.e., the CDFs of \( T_1 \) and \( T_2 \), can be determined, we can calculate a lower bound for \( p_j^\tau \).

As shown in Fig. 9, for the first-phase spray-and-wait from \( s_j \) to \( r_i \), there are multiple possible forwarding paths, i.e., \( s_j \to r_j \), and \( s_j \to u_k \to r_i \) with \( 1 \leq k \leq |V|, u_k \notin \{s_j, d_j\} \). All of these forwarding paths are independent with each other because no forwarding hop is shared among them. If we denote the forwarding time on the paths as \( T_{s_j \to r_i} \) and \( T_{s_j \to u_k \to r_i} \), respectively, \( P(T_1 \leq t) \) can be computed as:

\[
P(T_1 \leq t) = 1 - (1 - P(T_{s_j \to r_i} \leq t))
\]

\[
\prod_{k=1, u_k \notin \{s_j, d_j\}}^{V} \left(1 - P(T_{s_j \to u_k \to r_i} \leq t)\right)
\]

(13)

where \( P(T_{s_j \to r_i} \leq t) \) and \( P(T_{s_j \to u_k \to r_i} \leq t) \), i.e., the probabilities of successful delivery on paths \( s_j \to r_i \) and \( s_j \to u_k \to r_i \), can be calculated given the following theorem:

Theorem 2: For a k-hop forwarding path \( P(\lambda_1, \lambda_2, \ldots, \lambda_k) \), in which \( \lambda_1, \lambda_2, \ldots, \lambda_k \) are the rate parameters on the \( k \) hops respectively, the probability that the data is successfully delivered on this path within time \( t \) is

\[
p'(\lambda_1, \lambda_2, \ldots, \lambda_k) = \sum_{i=1}^{k} \left[ (1 - e^{-\lambda_i t}) \cdot \prod_{j=1, j \neq i}^{k} \lambda_j \right]
\]

(14)

The above theorem can be proved by considering that the delivery time on the \( k \)-hop forwarding path satisfies hyper-exponential distribution. The proof is similar to [28], and will not be presented here due to space limitation.

From Theorem 2, we have \( P(T_{s_j \to r_i} \leq t) = p'(\lambda_{s_j, r_i}) \), where \( \lambda_{s_j, r_i} \) is the rate parameter of the contact process between nodes \( s_j \) and \( r_i \). In addition, \( P(T_{s_j \to u_k \to r_i} \leq t) = p'('\lambda_{s_j, u_k}, \lambda_{u_k, r_i}) \).

Finally, for \( P(T_1 \leq t) \) in (13), we have

\[
P(T_1 \leq t) = 1 - (1 - p'(\lambda_{s_j, r_i}))
\]

\[
\prod_{k=1, u_k \notin \{s_j, d_j\}}^{V} (1 - p'(\lambda_{s_j, u_k}, \lambda_{u_k, r_i}))
\]

(15)

Similarly, \( P(T_2 \leq t) \) can be computed as:

\[
P(T_2 \leq t) = 1 - (1 - p'(\lambda_{r_i, d_j}))
\]

\[
\prod_{k=1, u_k \notin \{s_j, d_j\}}^{V} (1 - p'(\lambda_{r_i, u_k}, \lambda_{u_k, d_j}))
\]

(16)

Based on (15) and (16), \( p_j^\tau \) in (7) can be calculated.

The time complexity of computing \( P(T_1 \leq t) \) or \( P(T_2 \leq t) \) is \( O(|V|) \). To calculate the lower-bound of \( p_j^\tau \) using (10) for
Algorithm 1: SL-Betweenness Based Relay Placement.

Input: The set of social links $L$, the set of candidate static relays $R$

Output: An optimal set of relays with size $\theta$, $R^*$

1: Initialize: $R^* \leftarrow \emptyset$
2: for Each link $l_i \in L$ do
3: \quad Compute $\epsilon_j^i$ according to Equation (17)
4: \quad end for
5: while $|R^*| \leq \theta$ do
6: \quad for Each candidate relay $r_i \in R$ do
7: \quad \quad Compute $SL$-betweenness($r_i$) according to Equation (4)
8: \quad \quad end for
9: \quad Find max SL-betweenness from $SL$-betweenness($r_i$) for $r_i \in R$
10: \quad $r_i^* \leftarrow$ the candidate relay that achieves the max SL-betweenness
11: \quad $R^* \leftarrow R^* \cup \{r_i^*\}$
12: \quad Update $\epsilon_j^i$ according to Equation (20)
13: end while

$r_i$ and $l_j$, the time complexity is $O(|L|/|V|)$. Then, calculating $p_j^R$ for all links and relays requires a time complexity of $O(|L||R||V||N_j|)$, where $N_j = \max_j \{\epsilon_j^i\}$.

2) The Heuristic Based Algorithm: The algorithm works by greedily selecting the relay with the highest SL-betweenness from the set of candidate static relays. The detailed algorithm for selecting relays is shown in Algorithm 1. Note that before any relay is added, a portion of the load (data) at social links can be delivered using the traditional spray-and-wait, which does not rely on the static relays. Specifically, with spray-and-wait, the data can be either delivered from source to destination directly with one hop (denoted as case (a)), or delivered using two hops by going through an intermediate mobile node (denoted as case (b)). Based on this, the remaining load at link $l_j$ is:

$$\epsilon_j^i = \epsilon_j \cdot (1 - p_j^{(a)})(1 - p_j^{(b)})$$

(17)

where $p_j^{(a)}$ and $p_j^{(b)}$ are the forwarding probabilities with case (a) and case (b), which can be calculated based on Theorem 2:

$$p_j^{(a)} = p_j^\tau (\lambda_{s_j,d_j})$$

(18)

$$p_j^{(b)} = 1 - \prod_{k=1,u \in \delta(s_j,d_j)} (1 - p_j^\tau (\lambda_{s_j,u,k},\lambda_{u,k,d_j}))$$

(19)

The above process of updating loads corresponds to Line 2–4 in Algorithm 1.

Then, the algorithm works by greedily selecting the relay with the highest SL-betweenness, which is calculated using Equation (4) (Line 6–11). After selecting every relay (e.g., $r_i$), the load at link $l_j$ is relieved and adjusted as follows:

$$\epsilon_j = \epsilon_j \cdot (1 - p_j^\tau)$$

(20)

corresponding to Line 12 in Algorithm 1. This selecting process continues until $\theta$ relays are selected.

Note that $p_j^R$ used for calculating SL-betweenness only needs to be calculated once. When updating SL-betweenness, the calculated $p_j^R$ can be used directly. If $p_j^R$ is known, the time complexity of Algorithm 1 is $O(\theta|L|R)$. Calculating $p_j^R$ for all links and relays requires $O(|L||R||V||N_j|)$ as discussed before. The overall time complexity is $O(\theta|L|R) + |L||R||V||N_j|$. Due to the complex structure of mobile social networks, it is intractable to formally derive the approximation ratio and theoretically prove the effectiveness of the proposed greedy algorithm. Instead, we conduct experiments in the next section to verify its effectiveness.

C. The Replacement of Static Relays

As discussed in Section III-C, the traffic loads on social links may evolve over time. With the evolution on social links, the placed static relays may become outdated, and hence the placed static relays should be updated periodically (e.g., per week). However, repeating the relay placement procedure in every time step is very time-consuming and may generate a lot of overhead. In many cases, although there are only minor changes on social links, the newly calculated relays may be totally different from the previous placed relays. However, replacing all the relays may require lots of resources, especially when there are many static relays.

In this paper, instead of repeating the complete relay placement procedure, we propose a relay replacement algorithm that only replaces a limited number of relays to achieve much better performance. The basic idea is as follows. At the end of every time step, if traffic load changes are detected on social links, the existing relay with the lowest SL-betweenness is removed and replaced with the candidate relay that achieves the highest SL-betweenness. The process continues until the SL-betweenness of the new relay is equal to or smaller than the SL-betweenness of the existing relay. The relay replacement algorithm is outlined in Algorithm 2.

The relay replacement algorithm is an iterative process (as indicated by the while loop, Line 1). In each iteration, the SL-betweenness of each existing relay $r_i$ is first calculated (Line 2–7). When calculating the SL-betweenness for the existing relay $r_i$, based on Equation (4), we should note that the set of social links and the loads on them should be updated. If we denote the set of updated social links as $\hat{L} = \{\hat{l}_1,\hat{l}_2,\ldots,\hat{l}_L\}$, and the updated load on link $\hat{l}_j$ as $\hat{e}_j$. With the existing relays in $R^*$, the load on link $\hat{l}_j$ can be relieved and adjusted as:

$$\hat{e}_j,R^* = \hat{e}_j \cdot (1 - p_j^{(a)})(1 - p_j^{(b)}) \cdot \prod_{r \in R^*} (1 - p_j^R)$$

(21)

Based on this, the SL-betweenness for an existing relay $r_i$ can be calculated as:

$$SL$-betweenness$(r_i) = \sum_{j=1}^{|L|} \hat{e}_j,R^* \cdot p_j^{R_i}$$

(22)

where $\hat{e}_j,R^*$ is the remaining load on link $\hat{l}_j$ with the relays in $R^* \setminus \{r_i\}$, i.e., the set of existing relays excluding relay $r_i$. After calculating the SL-betweenness for each existing relay (Line 2–7), we find the relay $r_i^*$ that has the minimum
of iterations $m$ can be much smaller than $\theta$, resulting to much smaller time complexity.

V. PERFORMANCE EVALUATIONS

In this section, we evaluate the performance of the proposed static relay-based solutions. We first discuss how to generate social links in the network, and then describe the approaches in comparison and the evaluation results.

A. Social Link Generation

For the four datasets used in this paper, only Social Evolution includes information on social links, e.g., the SMS-based social links or phone call–based social links. For the other three datasets, a social network generating approach needs to be applied to generate the social links among mobile users. The social network, including all social links among mobile users, is different from a random network, in which the links are randomly generated between nodes. As suggested by the small-world experiment [34], the social network composed by human being is a small-world type network characterized by short path lengths between any pair of people. This feature of social network is called the small-world phenomenon.

The most notable network generation model on small-world phenomenon is the Watts-Strogatz model proposed by Watt and Strogatz [35]. The network generated by Watts-Strogatz model has many favorable social-network properties, including short average path lengths and high clustering. In this paper, we apply the Watts-Strogatz model to generate the social network, from which the social links are extracted.

The Watts-Strogatz model works by first creating a regular lattice network, like a ring network, and then replacing every edge (e.g. $u-v$) of the lattice with a new edge $u-w$ with some rewiring probability $p$, where $w$ is randomly chosen from existing nodes. The rewiring probability $p$ indicates how random the network is. $p=0$ indicates there is no rewiring, so the network generated is simply a regular ring lattice network with long average path length. As $p=1$, every edge in the lattice network needs to be rewired, generating a totally random network with no clusters. In our experiments, we set $p=0.5$, to generate high-quality social networks with high clustering and short average path length.

The network generated by Watts-Strogatz model is an undirected network. For every edge in the network, two social links or phone call-based social links. For the other three datasets, a social network generating approach needs to be applied to generate the social links among mobile users. The social network, including all social links among mobile users, is different from a random network, in which the links are randomly generated between nodes. As suggested by the small-world experiment [34], the social network composed by human being is a small-world type network characterized by short path lengths between any pair of people. This feature of social network is called the small-world phenomenon.

The most notable network generation model on small-world phenomenon is the Watts-Strogatz model proposed by Watt and Strogatz [35]. The network generated by Watts-Strogatz model has many favorable social-network properties, including short average path lengths and high clustering. In this paper, we apply the Watts-Strogatz model to generate the social network, from which the social links are extracted.

The Watts-Strogatz model works by first creating a regular lattice network, like a ring network, and then replacing every edge (e.g. $u-v$) of the lattice with a new edge $u-w$ with some rewiring probability $p$, where $w$ is randomly chosen from existing nodes. The rewiring probability $p$ indicates how random the network is. $p=0$ indicates there is no rewiring, so the network generated is simply a regular ring lattice network with long average path length. As $p=1$, every edge in the lattice network needs to be rewired, generating a totally random network with no clusters. In our experiments, we set $p=0.5$, to generate high-quality social networks with high clustering and short average path length.

The network generated by Watts-Strogatz model is an undirected network. For every edge in the network, two social links or phone call-based social links. For the other three datasets, a social network generating approach needs to be applied to generate the social links among mobile users. The social network, including all social links among mobile users, is different from a random network, in which the links are randomly generated between nodes. As suggested by the small-world experiment [34], the social network composed by human being is a small-world type network characterized by short path lengths between any pair of people. This feature of social network is called the small-world phenomenon.

The most notable network generation model on small-world phenomenon is the Watts-Strogatz model proposed by Watt and Strogatz [35]. The network generated by Watts-Strogatz model has many favorable social-network properties, including short average path lengths and high clustering. In this paper, we apply the Watts-Strogatz model to generate the social network, from which the social links are extracted.

The Watts-Strogatz model works by first creating a regular lattice network, like a ring network, and then replacing every edge (e.g. $u-v$) of the lattice with a new edge $u-w$ with some rewiring probability $p$, where $w$ is randomly chosen from existing nodes. The rewiring probability $p$ indicates how random the network is. $p=0$ indicates there is no rewiring, so the network generated is simply a regular ring lattice network with long average path length. As $p=1$, every edge in the lattice network needs to be rewired, generating a totally random network with no clusters. In our experiments, we set $p=0.5$, to generate high-quality social networks with high clustering and short average path length.

B. Approaches in Comparison

We evaluate the performance of the proposed SL-betweenness based approach (DPSW-SLB) by comparing it with three existing data forwarding approaches without static relays:

\[ e_j = e_{j,R\\backslash\{r\}} \quad (23) \]

then

\[ R^* \leftarrow R^* \backslash \{r^*\} \cup \{r_i\} \]

else

\[ \text{break} \quad */ The \ while \ loop \ terminates \ here */ \]

end if

end while

Algorithm 2: Relay Replacement.

Input: The set of updated social links $\hat{L}$, the set of candidate static relays $R$, and the current set of placed relays, $R^*$

Output: An updated set of relays, $R^*$

1: while true do
2:   for Each relay $r_i \in R^*$ do
3:      for Each link $\hat{l}_j \in \hat{L}$ do
4:         Calculate $e'_{j,R\\backslash\{r\}}$ according to Equation (21)
5:         end for
6:      Compute $SL$-betweenness($r_i$) according to Equation (22)
7:   end for
8:   Find min $SL$-betweenness from $SL$-betweenness($r_i$) for $r_i \in R^*$
9:   $r^*_i \leftarrow$ the relay that achieves the min $SL$-betweenness
10:  for Each link $\hat{l}_j \in \hat{L}$ do
11:      $e'_{j} \leftarrow e'_{j,R\\backslash\{r\}}$ according to Equation (21)
12:     end for
13:     for Each candidate relay $r'_i \in R\backslash R^*$ do
14:        Compute $SL$-betweenness($r'_i$) according to Equation (4)
15:     end for
16:     Find max $SL$-betweenness from $SL$-betweenness($r'_i$), $r'_i \in R\backslash R^*$
17:     $r^*_i \leftarrow$ the candidate relay that achieves the max $SL$-betweenness
18:     if $SL$-betweenness($r^*_i$) > $SL$-betweenness($r^*_i$)
19:        $R^* \leftarrow R^* \backslash \{r_i\} \cup \{r'_i\}$
20:     else
21:        break */ The while loop terminates here */
22:  end if
23: end while
Epidemic [4]: Upon contact, the data carrier forwards data to the contacted node if it does not have the data.

PROPHET [7]: When the data carrier contacts a node without data, it only forwards the data if the contacted node has a higher forwarding metric, which is the node’s probability to contact the destination.

Spray-and-Wait [6]: The source sprays data copies to a group of nodes, and then these nodes wait until contacting with the destination. To achieve the best performance, we do not set a limit to the number of sprayed data copies. The first hop is forwarded by flooding.

In addition to the above three strategies without static relays, we also compare with two other approaches of using static relays. They use the same DPSW data forwarding strategy, but different relay placement methods.

DPSW-Random: the static relays are randomly chosen from the candidate static relays.

DPSW-Coverage: It works by greedily selecting the static relay that has contacted the largest number of users, so that the selected static relays can cover as many users as possible.

C. Evaluation Results

In the experiments, the time constraints of data items are generated uniformly within the range \([T_{\text{min}}, T_{\text{max}}]\). \(T_{\text{min}}\) is set to 1 hour. \(T_{\text{max}}\), the max time constraint, can be adjusted to show how different time constraints affect the performance.

Two performance metrics are used for evaluations, delivery ratio and overhead. The delivery ratio is the proportion of data items successfully delivered to the destination before they expire. The overhead is measured by the average number of data copies created in the network for each data item. Each experiment is repeated five times and the results are averaged for consistency.

1) Comparisons With Approaches Without Static Relays: We first compare the DPSW-SLB approach with other existing approaches under different time constraints \((T_{\text{max}})\), and the results are shown in Fig. 10 and Fig. 11. Here, different number of static relays are placed in the datasets. For Social Evolution, Infocom and UCSD datasets, DPSW-SLB with 1 and 4 static relays are used. For the Taxi dataset, DPSW-SLB with 5 and 15 static relays are used, because the taxi dataset covers a large city area and requires more relays. This experiment is conducted with two weeks in each dataset (excluding the Infocom dataset). The first week is used as the warm-up period to learn about social links and traffic loads, and the second week is used for executing the data forwarding strategies. For the Infocom dataset, two days are used, with the first day for warm-up and the second day for execution. The evolution of social links is not considered here due to the limited time period.

As can be seen from Fig. 10, DPSW-SLB consistently outperforms the approaches without static relays. DPSW-SLB with 4 relays (15 relays for taxi dataset) can even achieve better delivery ratio than the Epidemic approach. This demonstrates that placing static relays in the network can significantly improve the delivery ratio, as static relays can help data to reach more mobile users. Another observation is that the delivery ratio of all approaches increases as the max time constraint increases, because the data have more time to be delivered. We also find that DPSW with more relays consistently achieves better delivery ratio than DPSW with less relays, indicating that placing more relays in the network can improve the data delivery ratio.

As shown in Fig. 11, DPSW has much less overhead than Epidemic. This is because DPSW limits the number of forwarding hops to four, while Epidemic does not put any limit
on the number of forwarding hops. Although DPSW-SLB creates more overhead than PROPHET and Spray-and-Wait, it has much higher data delivery ratio.

2) Comparisons Among Relay Placement Approaches: In this subsection, we evaluate the performance of DPSW-SLB by comparing it with other relay placement approaches. We set $T_{max}$ to 4 hours, and change the number of static relays. As shown in Fig. 12, DPSW-SLB outperforms other approaches in terms of data delivery ratio, which demonstrates the effectiveness of the SL-betweenness based relay placement.

We can also see that the delivery ratio for DPSW-SLB generally increases as the number of relays increases for all four datasets, because placing more relays can help maintain more social links and therefore improve the performance of data forwarding. Another interesting observation is the number of relays needed to ensure an efficient data forwarding. As we can see, for the Social Evolution dataset, the delivery ratio of DPSW-SLB improves significantly from one relay to two relays, but keeps flat after two relays, indicating that two relays are enough. Similarly, the number of relays needed for the Infocom dataset is 6, as the delivery ratio becomes steady with more than 6 relays. For the UCSD dataset, the number of relays needed is 4. For the Taxi dataset, the delivery ratio keeps increasing as more relays are added, but becomes stable after 30 relays are placed. The Taxi dataset requires more relays since it covers a much larger area.

Based on the results, the number of static relays required to ensure efficient data forwarding is usually fixed. As long as the number of placed static relays $n$ is equal or higher than the number of needed static relays, efficient data forwarding can be achieved. Since social relationships are usually much stable, the network performance should be stable. Although not frequently happen, the social links and the traffic loads may evolve after some long period of time, e.g., there are more social links and traffic loads. Then, it may need more static relays or adjust the existing static relays to achieve efficient data forwarding. Thus, the network performance should be periodically monitored. If there is some large performance degradation on data delivery, and there are large changes on social links and traffic loads, it is necessary to increase the number of static relays or adjust the locations of the static relays. The same experiments that help obtain the result of Fig. 12 will be conducted to determine the number of static relays for the updated social links, and how to adjust them accordingly.

3) Evaluation of Relay Replacement: Finally, we conduct another experiment to evaluate the performance of the relay replacement algorithm. In this experiment, the traffic loads on social links are evolving over time. The relay replacement algorithm is used to replace relays at the end of every time step, based on the newly updated traffic loads in the time step. The time step can be set flexibly, like a day, a week, or a month. Here the time step is set as a week. This is because if the time step is too short, like a day, the change on social links may be just temporary change, but not permanent evolution. If the time step is too long, like a month, the social link information may become outdated in a month’s period. We compare the data delivery ratios with relay replacement and without relay replacement in two datasets Social Evolution and UCSD. The other two datasets are not used here because of their short durations. The number of static relays is set to two in this experiment. This experiment is conducted with five weeks in both two datasets. As we can see from Fig. 13, the approach with relay replacement can successfully improve the delivery ratio by 10%–20% in both two datasets, especially when the max time constraint of data is higher. This result demonstrates the effectiveness of the relay replacement algorithm.

VI. CONCLUSION

In this paper, we proposed relay based solutions to improve the performance of data forwarding in mobile social networks. We focused on addressing the problem of where to place the static relays. We analyzed four real-world datasets of mobile social networks, and identified that data forwarding only appears in a small group of social links. Then, we formalized the problem of static relay placement as an optimization problem, and proposed a heuristic based solution to improve the data forwarding performance along these social links. Specifically, a
social-link betweenness metric is defined to quantify the importance of static relays on improving the delivery ratio, and then static relays are selected based on this metric. Considering that social links may evolve over time, we also proposed an efficient relay replacement algorithm which replaces outdated relays to further improve the data forwarding performance. Evaluation results showed that the proposed relay based solutions can significantly improve the performance of data forwarding in mobile social networks.

REFERENCES


