Efficient Routing Algorithms Combining History and Social Predictors in Mobile Social Networks

Xiao Chen[1], Chengyin Liu[2], Cong Liu[2]

[1]Department of Computer Science, Texas State University, San Marcos, TX USA

[2]Department of Computer Science, Sun Yat-Sen University, Guangzhou, China

Email: xc10@txstate.edu, liuchyin@163.com, liucong3@mail.sysu.edu.cn

***Abstract*—A mobile Social Network (MSN) is a type of wireless networks formed by people moving around carrying mobile devices. In this paper, we specifically study the MSNs that are formed impromptu, e.g. when people gather together for a conference, event, or festival. We refer to them as *Impromptu Mobile Social Networks* (IMSNs), which allow people to communicate in a lightweight fashion based on contact opportunities via local wireless bandwidth. In IMSNs, node connections are time-dependent and short-term. The existing MSN routing algorithms using network analysis of social network graphs and static node social features may not be suitable for IMSNs. Thus, we propose novel hybrid routing algorithms based on two time-related factors *node contact history* and *dynamic social features* to capture node mobility in IMSNs. We first propose a hybrid algorithm called *Hisso* that makes a weighted combination of the *his*tory and the *so*cial predictors based on these two factors. And then we upgrade it to *Enhanced Hisso* by introducing a novel concept called *social circle* in the social predictor to improve message delivery. Simulation results comparing our algorithms with the existing ones and with the ones that only consider one factor show that our algorithms outperform the others in terms of delivery ratio and latency with a slight increase in the number of forwardings. The results also confirm the effectiveness of using a node's social circle in our algorithms.**

***Index Terms*—history-based, mobile social networks, routing, social-based, social features**

I. Introduction

Mobile Social Networks (MSNs), formed by people moving around carrying mobile devices such as smartphones, tablets, laptops, and so on, are becoming ubiquitous in our daily lives. Different from the popular social networks like Facebook [1] and LinkedIn [2], in this paper, we study a specific kind of MSNs that are formed impromptu. We refer to them as *Impromptu Mobile Social Networks* (IMSNs). The IMSNs can be formed when people carrying mobile devices attend a conference, event, or festival. The IMSNs allow people to communicate in a lightweight mechanism based on contact opportunities via local wireless bandwidth such as Bluetooth without a network infrastructure. The links between nodes in IMSNs are time-dependent and short-term and thus continuous network connectivity is not guaranteed.

Nodes in IMSNs communicate through a store-and-forward fashion. When two nodes move within each other's transmission range, they communicate directly and become *neighbors* during that time period. When they move out of their ranges, their contact is lost. The message to be delivered needs to be stored in the local buffer until a contact occurs in the next hop. Due to the uncertainty and time-varying nature of IMSNs, routing poses unique challenges: The conventional ad-hoc network routing schemes, such as DSR [11], LAR [13], DSDV [16], AODV [17], etc., would fail. Routing in IMSNs requires a new model that consists of a sequence of independent, local forwarding decisions, based on the current connectivity information and the predictions of future connectivity to suit its distributed and dynamic nature.

In IMSNs, routing algorithms can be measured by three performance metrics. (1) *Delivery ratio*. It is the fraction of generated messages that are successfully delivered to the destination. (2) *Delivery latency*. It is the time between when a message is generated and when it is received by the destination. (3) *Number of forwardings*. It is the number of forwardings needed to deliver a message to its destination. An efficient routing algorithm should entail a high delivery ratio and low latency with an acceptable number of forwardings.

Despite their simplicity, rudimentary approaches such as Flooding [25] where a message holder epidemically sends a message to all of the nodes it contacts and Wait [12] where a source only directly delivers a message to the destination can still work in IMSNs. But Flooding has a high cost of message forwardings and Wait can have a long latency.

As mentioned above, the key to designing an efficient IMSN routing protocol is to have a good predictor for a message holder to locally identify a forwarder which is most likely to deliver a message to the destination based on the message holder's current connectivity information. In the literature, researchers use different current connectivity information to predict the future connectivity. Some researchers generate social network graphs by linking nodes with past encounters and apply complex social analysis methods based on node *centrality* and *similarity* to predict nodes' future meeting probabilities [5], [10]. To facilitate discussion, we refer to this method as *social-analysis-based* method. In this method, a link in the social network graph means that two nodes have met in the past, which has predictive value for future contacts. Nonetheless, the *aggregation* of contacts between nodes over time into a "static" social graph presents an inherent mapping tradeoff, where some information about timing of contacts is lost [9], [27]. Some other researchers use social features in user profiles as the current information to predict future connectivity [14], [26]. We refer to this method as *social-profile-based* method. The social features $F_1, F_2, F_3, \cdots$ may refer to people's *nationality*, *city*, *language*, etc. And $f_1, f_2, f_3, \cdots$ represent the values of these social features. For example, the value of *language* can be *English*. In this paper, for convenience's sake, when we mention "comparing node social features", we mean "comparing the values of their social features". The intuition of this method is that people come in contact more frequently if they have more social features in common. In each hop of the routing process, the message holder selects a node that has the most common social features with the destination as the next forwarder. The advantage of this method is that it does not need to record node contact history. Nevertheless, it may not work well if users' activities do not match the static social features in their profiles. For example, someone who puts New York as his *state* in his profile may actually attend a conference in Texas.

In the IMSNs we discuss in this paper, where links between nodes are time-dependent, we need to use information, especially time-related information, that can capture users' dynamic behavior to be used as the current information to predict future connectivity. Thus, we consider two pieces of information: *node contact history* and *dynamic social features*. Node contact history records the contact times of nodes when they move within each other's range. Node contact history is often used by researchers in designing various utility functions such as average number of contacts [3], number of times nodes met last [6], and elapsed time since last contact [22] in Delay Tolerant Network (DTN) routing protocols. As pointed out by [6], the history of contact between nodes contains valuable, but noisy information about the current network topology. Dynamic social features, relative to the static social features in user profiles, are used to capture nodes' dynamic behavior. We first introduced the idea in our paper [19]. In dynamic social features, we not only record if a node has particular social feature values, but also record the frequency this node has met other nodes which have these social feature values. For example, we not only record that a node A is a *New Yorker* and a *Student*, but also record that it has met *New Yorkers* 90% of the time and *Students* 80% of the time during the time interval we observe. Unlike static social features from user profiles, dynamic social features are time-related. So they change as user activities change over time. And thus we can have a more accurate way to choose the next best forwarder. For example, suppose the destination has social feature values *New Yorker* and *Student* and we have two candidate nodes A and B, both of which are *New Yorkers* and *Students*. Nodes A and B are equally good forwarders if we just look at their static social feature values. However, if we know that A has met *New Yorkers* 90% of the time and *Students* 80% of the time and B has met *New Yorkers* 60% of the time and *Students* 40% of the time during the time interval we observe, then obviously A is a better forwarder.

Based on the above discussion, in this paper, we will use node contact history and dynamic social features as the current information and design efficient hybrid routing algorithms using predictors combining the history predictor based on the node contact history and the social predictor based on the dynamic social features. As far as we know, we are the first to combine these two time-related factors to capture the dynamic behavior of nodes to make routing decisions in IMSNs. We first put forward a routing algorithm called *Hisso* which selects the next best forwarder based on the weighted combination of the *his*tory and the *so*cial predictors. In Hisso, the history predictor is obtained by applying linear regression to the node contact history with the destination. And the social predictor is calculated by utilizing social similarity metrics derived from data mining [8] on the dynamic social features. Next, we upgrade Hisso to *Enhanced Hisso* by introducing a novel concept called *social circle* in the social predictor calculation. A node's social circle here refers to a group of nodes that are socially similar to this node based on the dynamic social features. In the social predictor in Enhanced Hisso, we not only consider the social similarity of a node with the destination but also the meeting probability and the social similarity of this node with the destination's social circle. The intuition is that if a node is socially similar to the destination or often meets a set of nodes that are socially similar to the destination, then this node will have a higher probability to forward the message to the destination by itself or through one of the nodes in the destination's social circle.

To evaluate the performance of our algorithms, we compare them with the social-analysis-based and social-profile-based methods, and the pure history-based and pure dynamic-social-feature-based algorithms that just adopt node contact history or dynamic social features as the current information by simulations using a real trace reflecting the scenarios of the IMSNs we discuss. We implement single-copy (only one copy of the message exists in the network in the routing process) and multi-copy (multiple copies of the message exist in the network in the routing process) versions of the algorithms. In both versions, the results consistently show that our algorithms outperform the existing algorithms in terms of delivery ratio and latency with a slight increase in the number of forwardings. Furthermore, the Enhanced Hisso algorithm is shown to be better than Hisso due to the utilization of the social circle concept. These results demonstrate the effectiveness of using time-related information and a node's social circle in the routing algorithms for IMSNs.

The rest of the paper is organized as follows: Section II references the related works; Section III presents our routing algorithms; Section IV explains the calculation of the predictors; Section V shows the simulation results and the conclusion and future work are in Section VI.

II. Related works

In this section, we reference some related routing algorithms in MSNs. The rudimentary and history-based approaches were originally developed for DTNs. But they can also be used in MSNs. The social-based and hybrid approaches are designed for the MSNs as they consider social factors.

A. Rudimentary Approaches

One efficient yet costly routing approach in DTNs is Flooding [25] where a message holder forwards a message to all of the hosts it meets. The opposite approach is Wait [12], where the source just waits and sends the message to the destination directly when they meet. It only has one forwarding but the latency can be very high.

B. History-based Approaches

In DTN routing algorithms, several papers make routing decisions using utility functions based on the contact history of nodes. Dubois-Ferriere et al. [6] predict the nodes' future meeting probability by the number of times two nodes met last. Chen et al. [3] consider not only that but also the frequency of nodes contacting the destination in the past and calculate the average. There are also some variations of these algorithms. For example, Spyropoulos et al. [22] record the time elapsed since every other node was last encountered as the elapsed time contains the relative location information of the nodes. And Chen et al. [4] develop an extended information model to capture more history information and use regression methods to predict nodes' future meeting probability to guide routing.

C. Social-based Approaches

As social network applications explode in recent years, analysis of these network graphs shows that some nodes are the common acquaintances of other nodes and act as communication hubs [15], [24]. Therefore, one promising way of predicting future contact probability is to use metrics such as *centrality* and *similarity* in network analysis to assess the message delivery probability of a node based on the connections in the graphs [5], [10], [18]. Nevertheless, in these network graphs, past node contacts have been aggregated into a "static" social graph. As pointed out by [9], [27], the "static" social graph has the tradeoff between time-related information lost and predictive capability.

Some other MSN routing algorithms use social features in user profiles to guide routing. Mei et al. [14] find that individuals with similar social features tend to meet more often in MSNs. The individuals are characterized by high dimensional feature profiles, though usually only a small subset of important features are extracted and used in routing. Wu et al. [26] provide a systematic multicast routing approach to resolving social feature differences between a source and destinations by taking advantage of the structural property of hypercubes. The advantage of these social-profile-based approaches is that they do not need to record node contact history. They work well in social networks where the activities of individuals follow the information in their profiles.

In our recent work [19], we make our first attempt to design routing algorithms for the IMSNs we address in this paper. We find that in many mobile social networks, user activities are time-dependent and may deviate from their social features in their profiles. For example, someone who puts New York as his *state* in his user profile may actually attend a conference in Texas. Therefore, in order to capture nodes' dynamic behavior to steer the routing in the right direction, we use dynamic social features which not only record a node's social features but also its meeting frequencies with other nodes having the considered social features. With the similarity metrics derived from data mining [8], we put forward a routing algorithm called *Sosim* [19] which makes routing decisions based on the similarity of nodes' dynamic social features.

D. Hybrid Approaches

In the literature, there are also some hybrid approaches that consider more than one factor to make routing decisions. SimBet [5] routing algorithm uses a hybrid forwarding rule that makes effective use of two metrics (*centrality* and *similarity*) to determine the "bridge" nodes and nodes' social similarity to the destination to deliver data from one node to another. The centrality here refers to the betweenness centrality of each node, which is estimated only in its local neighborhood to avoid exchanging the network topology information. For the similarity metric, the number of common neighbors of the current node with the destination is calculated. The Bubble rap algorithm [10] considers two characteristics of a node- its *community* and *centrality* to make routing decisions. Here, a node can be a member of more than one community and it has a local centrality showing its popularity in its own community and a global centrality showing the popularity of the node in the network. Both the SimBet and the Bubble rap algorithms apply complex network analysis methods to social network graphs built by aggregating past node contacts.

III. Our Routing Algorithms

In this section, we propose two hybrid routing algorithms Hisso and Enhanced Hisso that select the next best forwarder based on its highest probability to meet the destination resulted from the predictors using node contact history and dynamic social features as the current information. Though these two algorithms use different predictors, they use the same routing algorithm framework which is presented in the next paragraph.

A. Routing Framework of Our Algorithms

Our algorithms Hisso and Enhance Hisso use the same routing framework as shown in Fig. 1 to deliver messages from a source to a destination. The difference between them is that they use a different predictor $r_{u_i d}$ to make routing decisions. In the routing framework, we adopt delegation forwarding [7], which is proved by the authors to bring down the expected cost of message delivery from $O(N)$ to $O(\sqrt{N})$, where N is the number of nodes in the network. In delegation forwarding, each node u_i is assigned a quality $r_{u_i d}$ which in our algorithms indicates u_i's probability to meet destination d in the future based on node contact history and dynamic social features, and a level value τ_i. Initially, the level of each node is equal to its quality. In each hop of delegation forwarding, a message holder u_i only considers forwarding the message to a node u_j which has a higher quality than u_i's level hoping that u_j has a better chance to deliver the message to the destination. At the same time, node u_i improves its level to the quality of u_j. In the rest of the routing process, each message holder does the same thing until the destination receives the message. The essence of delegation forwarding is that a copy is transferred

Routing Algorithm Framework based on Node Contact History and Dynamic Social Features

Let $u_1, u_2, \cdots, u_{N-1}$ be nodes in the network, d be the destination, and $r_{u_i d}$ be the probability that node u_i will meet d in the future based on node contact history and dynamic social features. The calculations of $r_{u_i d}$ in Hisso and Enhanced Hisso are explained in Sections III-B and III-C.
Each node u_i has quality $r_{u_i d}$ and level τ_i.
INITIALIZE $\forall i : \tau_i \leftarrow r_{u_i d}$
On contact between message holder u_i and node u_j:
if u_j is the destination d **then**
    u_i forwards the message to u_j and the algorithm is terminated
else if $\tau_i < r_{u_j d}$ **then**
    $\tau_i \leftarrow r_{u_j d}$
    if u_j does not have the message **then**
        u_i forwards the message to u_j
    end if
end if

Fig. 1. The routing algorithm framework

to a newly encountered node if the node is "closer" to the destination based on a certain predictor than other nodes that the current node has already met.

B. The Hisso Algorithm

The predictor r_{ud} of the Hisso algorithm is shown in Formula (1). It predicts the probability of u meeting destination d using a weighted combination of the history predictor h_{ud} based on node contact history and the social predictor s_{ud} based on the similarity of nodes' dynamic social features. The detailed calculations of h_{ud} and s_{ud} will be explained in Sections IV-A and IV-B, respectively. In the formula, parameters α and β are weights and $\alpha + \beta = 1$. Note that when α is 1, the algorithm becomes the pure history-based algorithm and when β is 1, it becomes the pure dynamic-social-feature-based algorithm. The intuition of this formula is that if u often met d directly in history and/or it is very socially similar to d based on the dynamic social features, then it will very likely meet d again in the future.

$$\begin{aligned} r_{ud} &= \alpha \cdot \text{history predictor} + \beta \cdot \text{social predictor} \\ &= \alpha \cdot h_{ud} + \beta \cdot s_{ud} \end{aligned} \tag{1}$$

C. The Enhanced Hisso Algorithm

In the predictor r_{ud} shown in Formula (2) for the Enhanced Hisso, the social predictor is enhanced by considering not only the social similarity between u and d but also covering a broader scope - the social closeness between u and d's social circle. A node's social circle means a group of nodes that are socially similar to the node. Here, specifically, social similarity refers to the similarity of nodes based on their dynamic social features. The intuition of this idea is that if u is socially similar to d or often meets a set of nodes that are socially similar to d, node u will have a higher probability to successfully forward the message to d by itself or through one of the nodes in d's social circle. In Formula (2), parameters α and β are weights and $\alpha + \beta = 1$. Set L_u contains the top L nodes that are socially similar to d and frequently meet u, and s_{zd} is the social similarity between a node $z \in L_u$ and d. Again, how to calculate h_{ud} and s_{zd} will be explained in Sections IV-A and IV-B, respectively.

$$\begin{aligned} r_{ud} &= \alpha \cdot \text{history predictor} + \beta \cdot \text{social predictor} \\ &= \alpha \cdot h_{ud} + \beta \cdot \frac{\sum_{z \in L_u} s_{zd} h_{uz}}{\sum_{z \in L_u} h_{uz}} \end{aligned} \tag{2}$$

The Enhanced Hisso becomes Hisso when we only consider the social similarity of u to d. In other words, when L_u only contains u.

IV. Predictor Calculation

In this section, we explain the calculations of the history and social predictors. We normalize each predictor to the range of $[0, 1]$ so that they can be of similar order of magnitude - otherwise they may overshadow each other when combined.

A. History Predictor

For the history predictor, in [4], we proposed an information model to capture more contact information of nodes in history and used linear regression to predict future contacts for routing algorithms in DTNs. Since it can catch more history information and make good predictions [4], we adopt it to calculate the history predictor for routing algorithms in IMSNs as well. Its idea is as follows: We observe the past t time intervals $\{1, 2, \cdots, t\}$ in history. Each time interval i has a length of l units. Let n_i ($n_i \leq l$) be the number of contacts between node u and destination d in time interval i. Let X-axis represent the time interval i and Y-axis represent the number of contacts n_i. For t intervals, we can obtain t points (i, n_i) ($1 \leq i \leq t$) in the 2-D space as shown in Fig. 2. Although these t points (i, n_i) may not all lie on a line, it is reasonable to examine $d_i = n_i - (ai + b) = n_i - ai - b$, which is the difference between the y-coordinates of the point (i, n_i) and the corresponding point on the line $y = ax + b$. The Least-Squares-Method criterion for the "best" linear model approximation is to determine the values of a and b that minimize the sum of squares of all of the y-differences denoted by $F(a, b)$ as follows:

$$F(a, b) = \sum_{i=1}^{t} (n_i - ai - b)^2.$$

To minimize $F(a, b)$, we take the partial derivatives of $F(a, b)$ and set them equal to 0 to find the unique critical point for $F(a, b)$.

$$\begin{aligned} F'_a(a, b) &= -\textstyle\sum_{i=1}^{t} 2i(n_i - ai - b) &= 0, \\ F'_b(a, b) &= -\textstyle\sum_{i=1}^{t} 2(n_i - ai - b) &= 0. \end{aligned}$$

Thus

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{t} i^2 & \sum_{i=1}^{t} i \\ \sum_{i=1}^{t} i & t \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{t} i n_i \\ \sum_{i=1}^{t} n_i \end{bmatrix}$$

and

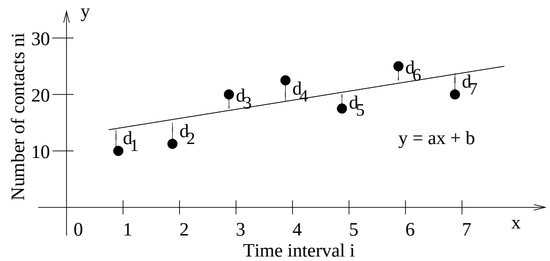


Fig. 2. Node contacts in the past time intervals

$$
\begin{aligned}
n_{t+1} &= a(t+1)+b = \begin{bmatrix} t+1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \\
&= \begin{bmatrix} t+1 & 1 \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{t} i^2 & \sum_{i=1}^{t} i \\ \sum_{i=1}^{t} i & t \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{t} in_i \\ \sum_{i=1}^{t} n_i \end{bmatrix} \\
&= \frac{\begin{bmatrix} t+1 & 1 \end{bmatrix} \begin{bmatrix} t & -\sum_{i=1}^{t} i \\ -\sum_{i=1}^{t} i & \sum_{i=1}^{t} i^2 \end{bmatrix} \begin{bmatrix} \sum_{i=1}^{t} in_i \\ \sum_{i=1}^{t} n_i \end{bmatrix}}{t\sum_{i=1}^{t} i^2 - \left(\sum_{i=1}^{t} i\right)^2}.
\end{aligned}
$$

The value of n_{t+1} is the predicted future contacts of u and d in time interval $i+1$ based on their contacts in the past t intervals. We set $h_{ud} = \frac{n_{t+1}}{lt}$ after normalization as our history predictor to estimate their future meeting probability.

B. Social Predictor

For the social predictor, the key points are how to represent nodes' dynamic social features and how to measure the social similarity of two nodes based on them. We explain them as follows. Suppose we consider m social features $\langle F_1, F_2, \cdots, F_m \rangle$ of nodes in IMSNs. We associate each node with a vector of its social features. For convenience, we use a node's label as its vector's label. Thus, a node x has a vector x of $\langle x_1, x_2, \cdots, x_m \rangle$ and a node y has a vector y of $\langle y_1, y_2, \cdots, y_m \rangle$. A node x's dynamic social features are contained in its vector, which is $< x_1, x_2, \cdots, x_m >= \left\langle \frac{M_1}{M_{total}}, \frac{M_2}{M_{total}}, \frac{M_3}{M_{total}}, \cdots \frac{M_m}{M_{total}} \right\rangle$, where M_i is the number of meetings of node x with nodes whose value f_i of feature F_i is the same as that of destination d, and M_{total} is the total number of meetings of node x with any other node in the history we observe. Thus $0 \leq x_i \leq 1$ for all $1 \leq i \leq m$. With the node's dynamic social features defined, we can use the following similarity metrics derived from data mining [8] to compare the similarity s_{xy} of nodes x and y.

- **Tanimoto similarity**

  It measures the similarity of x and y as: $s_{xy} = \dfrac{x \cdot y}{x \cdot x + y \cdot y - x \cdot y}$. The notation $x \cdot y$ is the product of the two vectors. For example, suppose we consider three social features: $\langle City, Language, Position \rangle$. If the values of the social features of destination d are: $\langle NewYork, English, Student \rangle$ and node x has met people from New York 70% of the time, people that speak English 93% of the time, and students 41% of the time in the history we observe, then node x has a vector of $x = \langle 0.7, 0.93, 0.41 \rangle$. If y's vector is: $y = \langle 0.23, 0.81, 0.5 \rangle$, then using the Tanimoto metric, $s_{xy} = 0.82$.

- **Cosine similarity**

  It measures the similarity of x and y as: $s_{xy} = \dfrac{x \cdot y}{\sqrt{(x \cdot x)(y \cdot y)}}$.

- **Euclidean similarity**

  After normalizing the original Euclidean similarity to the range of $[0, 1]$ and subtract it from 1, it is now defined as $s_{xy} = 1 - \dfrac{\sqrt{\sum_{i=1}^{m} (y_i - x_i)^2}}{\sqrt{m}}$.

- **Weighted Euclidean similarity**

  In addition to the basic Euclidean similarity mentioned above, we also employ the weighted Euclidean similarity to favor the social features that are more influential to the delivery of the packet. To determine the weight of a social feature, we use the Shannon entropy [21] which quantifies the expected value of the information contained in the feature [26]. The Shannon entropy for a given social feature is calculated as: $w_i = -\sum_{i=1}^{k} p(f_i) \cdot log_2(f_i)$, where w_i is the Shannon entropy for feature F_i, vector $\langle f_1, f_2, \cdots f_k \rangle$ contains the possible values of feature F_i, and p denotes the probability mass function of F_i. The weighted Euclidean similarity normalized to the range of $[0, 1]$ is: $s_{xy} = 1 - \dfrac{\sqrt{\sum_{i=1}^{m} w_i \cdot (y_i - x_i)^2}}{\sqrt{\sum_{i=1}^{m} w_i}}$.

V. Simulations

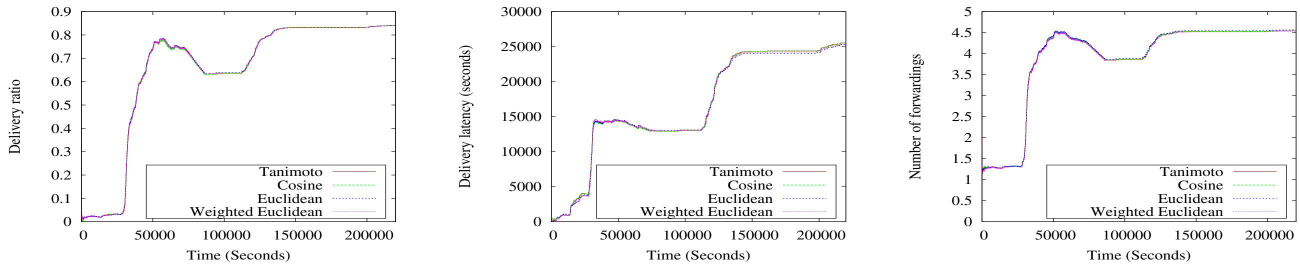
This section describes the simulations we conducted using a custom simulator written in Java. We first performed simulations to determine the values of the parameters in our algorithms and then we compared our algorithms with the existing ones in two versions: single-copy and multi-copy.

In our simulations, we used the INFOCOM 2006 trace [20], which recorded conference attenders' encounter history using Bluetooth small devices (iMotes) for three days at the IEEE INFOCOM 2006 conference in Miami. This data set consists of two parts: *contacts* between the iMote devices that were carried by participants and the self-reported *social features* of the participants, which are collected using a questionnaire form. The six social features extracted from the dataset and used for the social predictor were *Affiliation, City, Nationality, Language, Country,* and *Position.*

We use delivery ratio, delivery latency, and number of forwardings as important metrics to evaluate the performance of the algorithms. An efficient routing entails a high delivery ratio and low latency with an acceptable number of forwardings.

A. Comparison of social similarity metrics

To find the best fit for our simulated context, we compared Tanimoto, Cosine, Euclidean, and Weighted Euclidean social similarity metrics by performing delegation forwarding routing algorithm. We utilized the first two days of the data as the initial history and performed our simulations on the remaining one day. We generated messages from a randomly chosen



(a) Delivery ratio (b) Delivery latency (c) Number of forwardings

Fig. 3. Comparison of Tanimoto, Cosine, Euclidean, and Weighted Euclidean social similarity metrics

source to a randomly chosen destination every two seconds in the first 24 hours of the simulation. We then averaged five separate simulations for each similarity metric with identical setups to mitigate the effect of any outliers in the performance. We set time-to-live of all of the packets to 9, meaning that a given packet can be transferred at most nine times so that the delivery ratio will not always be 100% during the whole time frame of the trace. Results in Fig. 3 show that all of the metrics performed similarly in delivery ratio, latency, and forwardings. We therefore decided to use the Euclidean metric since it did not require the calculation of additional weighting values and performed slightly better than Tanimoto and Cosine in latency.

B. Determine α and β values

To give reasonable weights for the history and social predictors in our algorithms, we need to find out the values for α and β depending on the trace we use. We tried $< \alpha, \beta >$ pair to be: $< 0, 1 >$, $< 0.25, 0.75 >$, $< 0.5, 0.5 >$, $< 0.75, 0.25 >$, and $< 1, 0 >$ in Hisso and Enhanced Hisso algorithms. The results of both algorithms showed that $< \alpha = 0.75, \beta = 0.25 >$ has the highest delivery ratio, comparatively lower latency, and lower number of forwardings. Therefore, we set α to 0.75 and β to 0.25 for our trace. The performance of Hisso using different α and β values is shown in Fig. 4.

C. Determine the number of copies in multi-copy schemes

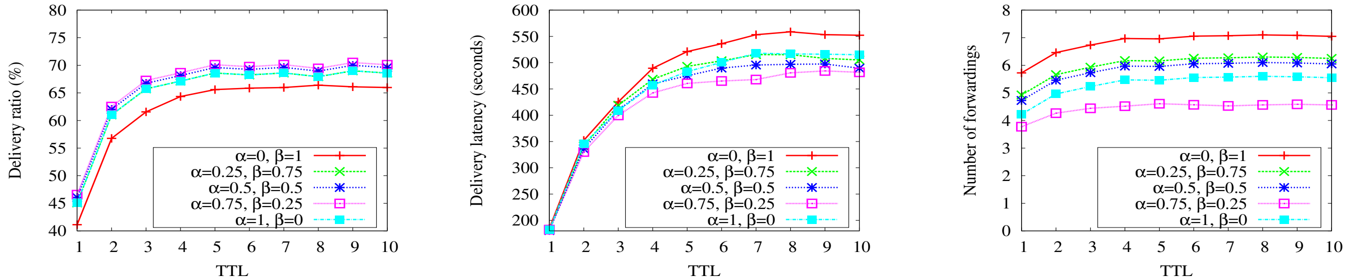
We implemented Hisso and Enhanced Hisso in two versions: single-copy and multi-copy schemes. In the single-copy version, only one copy of the message exists in the network during delivery. That is, each time a message holder forwards the message to the next forwarder, it does not keep a copy for itself. In the multi-copy version, we adopt the idea of binary Spray-and-Focus [23] in DTN as the authors showed that binary Spray minimizes the time to spray the message to newly encountered nodes and Focus can actively deliver the message to the destination. In the multi-copy version of our algorithms, the source of a message initially starts with $C (C \geq 1)$ copies. Then routing is carried out in the Spray and Focus phases. In the Spray phase, any node with $c > 1$ copies will forward half ($\lfloor c/2 \rfloor$) of the copies to the encountered node with no copy. Then in the Focus phase, if the destination is not found in the Spray phase, each message holder forwards the copy to the best encountered node selected by Hisso or Enhanced Hisso. To determine the value for C, we tried multi-copy versions of Hisso and Enhanced Hisso with C setting to 4, 8, 31, and 64. The simulation results using different C values of both algorithms are consistent. As shown by the multi-copy version of Hisso in Fig. 5, with the increase of the number of copies, the delivery ratio improves, but the latency and forwardings also increase. To reduce the cost, we therefore decided to set C to 4.

D. Comparison with existing algorithms

To evaluate the performance of our algorithms considering both history and social predictors, simulations were conducted to compare them with the existing social-analysis-based and social-profile-based algorithms, and pure history-based and pure dynamic-social-feature-based algorithms. The Flooding and Wait algorithms were included as benchmarks. The following is the list of algorithms we compare. To fit the legend in each figure later, we create a short name for each algorithm.

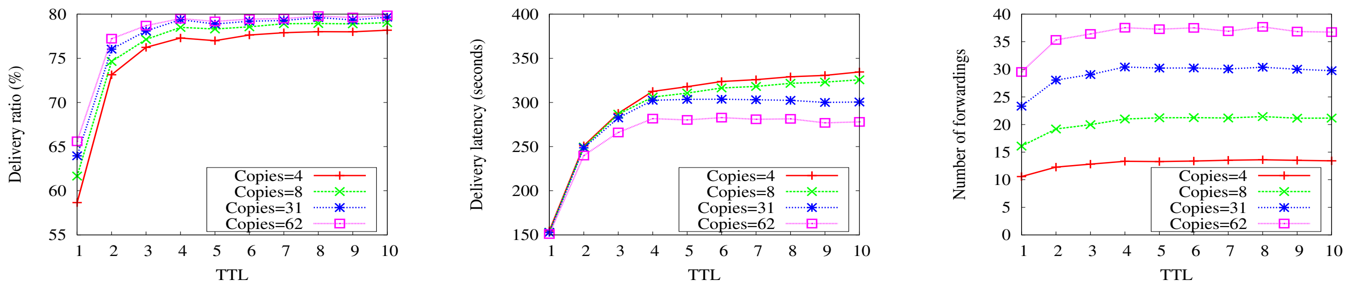
1) *The Flooding algorithm* (Flooding) [25]: Every message is spread epidemically throughout the network until it reaches its destination.
2) *The Wait algorithm* (Wait) [12]: The source holds the message until it meets the destination.
3) *The Social-profile-based algorithm* (Profile) [26]: It takes the idea from [26] where routing is guided by resolving social feature differences between source and destination using social features in user profiles.
4) *The Social-analysis-based algorithm* (Analysis) [5]: This algorithm takes the idea from [5] where routing decisions are made using social analysis methods on the social network graphs reflecting past node encounters.
5) *The History-based algorithm* (History) [4]: It just considers the history predictor using linear regression.
6) *The Dynamic-social-feature-based algorithm* (Sosim) [19]: It just considers the social predictor using dynamic social features.
7) *The Hisso algorithm* (Hisso): Both the history and social predictors are considered with weights.
8) *The Enhanced Hisso algorithm* (Enhanced Hisso): Both the history and social predictors are considered with the enhanced social predictor.

In the algorithms that involve social features, namely Hisso, Enhanced Hisso, and Sosim, the actual number of iMotes used



(a) Delivery ratio (b) Delivery latency (c) Number of forwardings

Fig. 4. The performance of single-copy Hisso using different α and β values



(a) Delivery ratio (b) Delivery latency (c) Number of forwardings

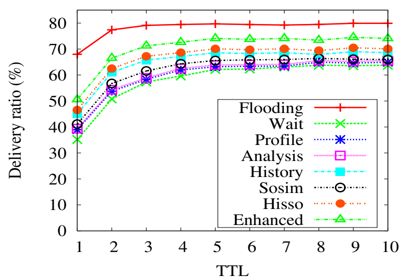
Fig. 5. The performance of multi-copy Hisso using different number of copies

was 62 in the trace after excluding 17 iMotes that have no or partial social features. We define 1 time interval (TTL) as $1/10$ of the whole trace time length. Then 10 TTLs is the time length of the whole trace. In the Hisso and Enhanced Hisso algorithms, we set α and β to 0.75 and 0.25, respectively according to our simulation results above. In the Enhanced Hisso algorithm, the size of L_u is decided by h_{uz}. We only select those nodes whose $h_{uz} > 0$. The total number of these nodes does not exceed 10 in this trace. For the multi-copy versions, we set C to 4 from the above discussion. All of the comparing algorithms have a multi-copy version using binary Spray-and-Focus. But Flooding and Wait are a little different in implementation. In Flooding, after the message is binary-sprayed to newly encountered nodes, the message holders still deliver the message epidemically to the destination without being constrained by the number of copies in the network. And in Wait, the source holds the initial copies of the message until it meets the destination. We generated 5000 packets between randomly chosen source-destination pairs and applied them to all of the algorithms. The three performance metrics, namely delivery ratio, latency, and number of forwardings, were calculated and averaged.

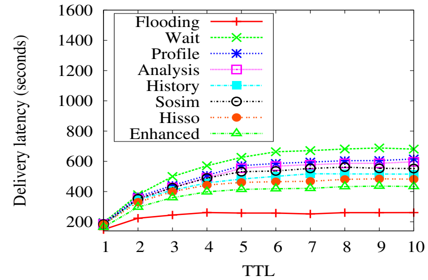
The simulation results of all of the algorithms in both single-copy and mulit-copy versions are consistent as shown in Figs. 6 and 7. As expected, Flooding has the highest delivery ratio and lowest delivery latency but highest number of forwardings. Wait has the lowest number of forwardings but lowest delivery ratio and highest delivery latency. The pure history-based algorithm History and pure dynamic-social-feature-based algorithm Sosim outperform Profile and Analysis in delivery ratio and latency at the cost of the slight increase in forwardings. That means, in our application scenario, the static social features in user profiles and the links in the social network graphs are less capable of capturing conference attenders' activities to make good predictions. The time-related predictors are better. The slight increase in the number of forwardings shows that History and Sosim are more actively forwarding messages than Profile and Analysis. Furthermore, our hybrid algorithms Hisso and Enhanced Hisso are better in delivery ratio and latency than the pure ones with a slight increase in the number of forwardings. This shows that the combination of the two time-related predictors works better. From the increase in forwardings, we can again conclude that Hisso and Enhanced Hisso are more active in message delivery. Comparing with Hisso, Enhanced Hisso performs better in delivery ratio and latency with a few more forwardings. This testifies our idea that considering a node's social circle can facilitate routing.

VI. Conclusion

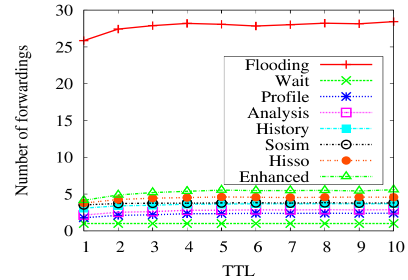
In this paper, we proposed novel hybrid routing algorithms for IMSNs where node connections are time-dependent and short-term by considering two time-related factors: node contact history and dynamic social features to capture node mobility in IMSNs. We first put forward a hybrid algorithm Hisso that makes a weighted combination of the history and the social predictors based on these two factors. And then we upgraded it to Enhanced Hisso by introducing a novel concept called social circle in the social predictor to improve message delivery. Simulations were conducted to compare our algorithms with the existing social-analysis-based and social-profile-based algorithms, and with the algorithms that only



(a) Delivery ratio

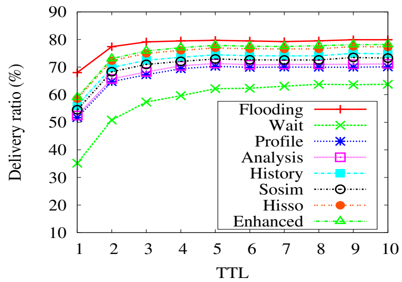


(b) Delivery latency

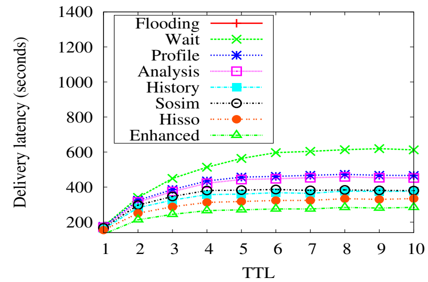


(c) Number of forwardings

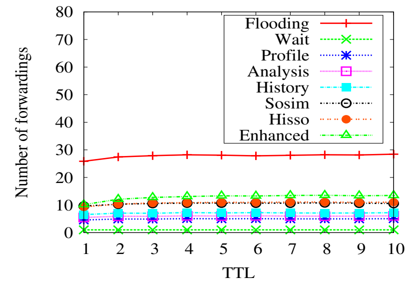
Fig. 6. The performance of the algorithms implemented in single-copy scheme



(a) Delivery ratio



(b) Delivery latency



(c) Number of forwardings

Fig. 7. The performance of the algorithms implemented in multi-copy scheme

consider one factor. Simulation results showed that our proposed algorithms outperformed the others in delivery ratio and latency with a slight increase in the number of forwardings. The results also confirmed that the idea of social circle can facilitate routing in our algorithms. In our future work, we will test our algorithms using more traces with social features as they become available and find better predictors to improve routing in IMSNs.

VII. Acknowledgement

This paper is supported in part by NSF CNS grant 1305302.

References

[1] Facebook. http://www.facebook.com.
[2] Linkedin. http://www.linkedin.com.
[3] X. Chen and A.L. Murphy. Enabling disconnected transitive communication in mobile ad hoc networks. In *Workshop on Principles of Mobile Computing*, pages 21–23, 2001.
[4] X. Chen, J. Shen, and J. Wu. Improving routing protocol performance in delay tolerant networks using extended information. *Journal of Systems and Software*, 83(8), August 2010.
[5] E. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *ACM MobiHoc*, 2007.
[6] H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli. Age matters: efficient route discovery in mobile ad hoc networks. In *ACM MobiHoc*, 2003.
[7] V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot. Delegation forwarding. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 251–260, 2008.
[8] J. W. Han, M. Kamber, and J. Pei. *Data Mining: concepts and techniques*. Morgan Kaufmann, MA, USA, 2012.
[9] T. Hossmann, T. Spyropoulos, and F. Legendre. From contacts to graphs: Pitfalls in using complex network analysis for dtn routing. In *IEEE Int. Workshop on Network Science For Communication Networks*, 2009.
[10] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-basedforwarding in delay tolerant networks. In *MobiHoc*, 2008.
[11] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, pages 153–181, 1996.
[12] E. P. Jones and P. A. Ward. Routing strategies for delay-tolerant networks. *SIGCOMM*, 2004.
[13] Y.-B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wireless Networks*, 6(4):307–321, July 2000.
[14] A. Mei, G. Morabito, P. Santi, and J. Stefa. Social-aware stateless forwarding in pocket switched networks. In *IEEE INFOCOM*, 2011.
[15] M. Motani, V. Srinivasan, and P. Nuggehalli. Peoplenet: engineering a wireless virtual social network. In *MobiCom*, pages 243–257, 2005.
[16] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM*, 1994.
[17] C. E. Perkins, E. M. Royer, and S. R. Das. Ad hoc on-demand distance vector (AODV) routing, IETF Internet draft, draft-ietf-manet-aodv-11.txt, June 2002.
[18] A. Pietilainen and C. Diot. Dissemination in opportunistic social networks: the role of temporal communities. In *ACM MobiHoc*, 2012.
[19] D. Rothfus, C. Dunning, and X. Chen. Social-similarity-based routing algorithm in delay tolerant networks. In *IEEE ICC*, 2013.
[20] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. Crawdad trace cambridge/haggle/imote/infocom2006 (v.2009-05-29), 2009.
[21] C. Shannon, N. Petigara, and S. Seshasai. A mathematical theory of communication. *Bell System Technical Journal*, 27(1):379–423, 1948.
[22] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Singlecopy routing in intermittently connected mobile networks. In *IEEE SECON*, 2004.
[23] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and focus: Efficient mobility-assisted routing for heterogeneous and correlated mobility. In *IEEE PERCOMW*, 2007.
[24] V. Srinivasan, M. Motani, and W. Ooi. Analysis and implications of student contact patterns derived from campus schedules. In *MobiCom*, pages 86–97, 2006.
[25] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Dept. of Comp. Sci., Duke Univ., 2000.
[26] J. Wu and Y. Wang. Social feature-based multi-path routing in delay tolerant networks. In *IEEE INFOCOM*, 2012.
[27] S. Yang and J. Wu. Adaptive backbone-based routing in delay tolerant networks. In *IEEE MASS*, October 2013.