Community and Social Feature-based Multicast in Opportunistic Mobile Social Networks

Charles Shang[1], Britney Wong[2], Xiao Chen[3], Wenzhong Li[4], Suho Oh[5]
[1]Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA
[2]Department of Computer Science, Cornell University, Ithaca, NY, USA
[3]Department of Computer Science, Texas State University, San Marcos, TX USA
[4]State Key Laboratory for Novel Software Technology, Nanjing University, China
[5]Department of Mathematics, Texas State University, San Marcos, TX USA
Email: cshang4@illinois.edu, bmw227@cornell.edu, xc10@txstate.edu, lwz@nju.edu.cn, suhooh@txstate.edu

***Abstract*—Opportunistic Mobile Social Networks (OMSNs), formed by people moving around carrying mobile devices such as smartphones, PDAs, and laptops, have become popular in recent years. They are a special kind of delay tolerant networks (DTNs) that exploit human social characteristics to perform message routing and data sharing. Multicast is an important routing service in OMSNs which supports the dissemination of messages to a group of users. Most of the existing multicast algorithms are designed for general-purpose DTNs where social factors are neglected or reflected in static social features which are not updated to catch nodes' dynamic contact behavior. In this paper, we introduce the concept of dynamic social features and its enhancement to capture nodes' dynamic contact behavior, consider more social relationships among nodes, and adopt the community structure in the multicast compare-split scheme to select the best relay node for each destination in each hop to improve multicast efficiency. We propose two multicast algorithms based on these new features. The first community and social feature-based multicast algorithm is called *Multi-CSDO* which involves destination nodes only in community detection, and the second one is called *Multi-CSDR* which involves both the destination nodes and the relay candidates in community detection. The analysis of the algorithms is given and simulation results using a real trace of an OMSN show that our new algorithms outperform the existing one in terms of delivery rate, latency, and number of forwardings.**

***Index Terms*—community, dynamic social features, mobile social networks, routing, static social features**

I. Introduction

With the proliferation of smartphones, PDAs, and laptops, Opportunistic Mobile Social Networks (OMSNs), formed by people moving around carrying these mobile devices, have become popular in recent years [7], [20], [22], [25], [26]. OMSNs are a special kind of delay tolerant networks (DTNs) that exploit the human social characteristics, such as similarities, daily routines, mobility patterns, and interests to perform message routing and data sharing [11], [18]. In such networks, the communication takes place on-the-fly by the opportunistic contacts among mobile nodes in a lightweight mechanism via local wireless bandwidth such as Bluetooth or WiFi without a network infrastructure. Due to the time-varying network topology of OMSNs, end-to-end communication path is not guaranteed, which poses special challenges to routing, either unicast or multicast. Nodes in OMSNs can only communicate through a store-carry-forward fashion. When two nodes move within each other's transmission range, they communicate directly and 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.

Multicast, a service where a source node sends messages to multiple destinations, widely occurs in OMSNs. For example, in a conference, presentations are delivered to inform the participants about the newest technology; In an emergency scenario, information regarding local conditions and hazard levels is disseminated to the rescue workers; And in campus life, school information is sent to a group of student mobile users over their wireless interfaces.

Most of the existing multicast algorithms are proposed for the general-purpose DTNs [12], [13], [19], [21], [24] without social characteristics. There are a few multicast algorithms involving social factors [6], [23] and taking advantage of the fact that *people having more similar social features in common tend to meet more often* in OMSNs. Social features F_1, F_2, $\cdots$, F_i, $\cdots$ can refer to $Nationality, City, Language, Affiliation$, and so on. Each social feature F_i can take multiple values f_1, f_2, $\cdots$, f_i, $\cdots$. For example, a social feature F_i can be $Language$ and its values can be $English$, $Spanish$, and so on. Deng et al. propose a social profile-based multicast algorithm (SPM) [6] based on static social features in user profiles. In our previous work [23], we argued that the static social features may not always reflect nodes' dynamic contact behavior and introduced dynamic social features to capture nodes' contact frequency with people having a certain social feature and then developed a social similarity-based multicast algorithm named Multi-Sosim based on dynamic social features. Simulation results showed that Multi-Sosim outperforms SPM.

In multicast, a message holder is expected to forward a message to multiple destinations. To reduce the overhead and forwarding cost, the destinations will share the routing path until the point that they have to be separated, which usually results in a tree structure. A compare-split scheme to determine the separation point is critical to the efficiency of a multicast.

In Multi-Sosim, when a message holder x meets another node y, they become *relay candidates* as they will be responsible for relaying the message to the destinations. We compare the social similarity of each of the destinations with the relay candidates based on the dynamic social features, and split the destinations according to the comparison results: whichever relay candidate is more socially similar to the destination will be responsible for relaying the message to that destination due to its higher delivery probability.

In this paper, we believe that Multi-Sosim can be further improved in two ways: (1) by enhancing the definition of dynamic social features, and (2) by adding the community structure among nodes into the compare-split scheme. The definition of the dynamic social features in Multi-Sosim is based on node contact frequency, which can be easily obtained and inexpensive to maintain in OMSNs. It also reflects the aforementioned intuition that people having more similar social features in common tend to have higher contact frequencies in OMSNs. But it cannot distinguish the cases when two nodes have the same meeting frequency with nodes having a certain social feature. Thus we upgrade dynamic social features to *enhanced dynamic social features* to break the tie. Moreover, the compare-split scheme in Multi-Sosim only considers the social relationship between each destination and each relay candidate, and ignores the relationships among the destinations. To identify socially similar nodes including the destinations, community detection technique is an ideal tool to be used in the compare-split scheme to enhance the efficiency of multicast. Different from the community structure where node social relationships are long-term and less volatile than node mobility in several social-aware routing schemes [5], [8], [10], our community detection involves dynamic social features which adapt to node mobility in OMSNs.

Based on the enhanced dynamic social features and the idea of the new compare-split scheme using community detection, we propose two novel ***C***ommunity and ***S***ocial feature-based multicast algorithms named *Multi-CSDO* that involves ***D***estination nodes ***O***nly in community detection and *Multi-CSDR* that involves both the ***D***estination nodes and the ***R***elay candidates in community detection in case the relay candidates are also socially similar. We provide theoretical analysis to the algorithms and to evaluate their performance, we compare them with Multi-Sosim, and Epidemic as a benchmark. Simulation results show that the enhanced dynamic social features can improve the performance of multicast and our new algorithms outperform Multi-Sosim in terms of delivery rate, latency, and the number of forwardings.

The rest of the paper is organized as follows: Section II references the related works; Section III introduces the preliminary; Section IV presents our new multicast algorithms; Section V gives the analysis of the algorithms; Section VI shows the simulation results; and Section VII is the conclusion.

II. Related works

The multicast algorithm in OMSNs can be implemented using rudimentary approaches such as Epidemic routing [17], but it has inevitable high forwarding cost. Most of the existing multicast algorithms are designed for DTNs where social features are not factored in. Zhao et al. [24] introduce some new semantic models for multicast and conclude that the group-based strategy is suitable for multicast in DTNs. Lee et al. [12] study the scalability property of multicast in DTNs and introduce RelayCast to improve the throughput bound of multicast using mobility-assist routing algorithm. By utilizing mobility features of DTNs, Xi et al. [21] present an encounter-based multicast routing, and Chuah et al. [4] develop a context-aware adaptive multicast routing scheme. Mongiovi et al. [13] use graph indexing to minimize the remote communication cost of multicast. Wang et al. [19] exploit the contact state information and use a compare-split scheme to construct a multicast tree with a small number of relay nodes.

There are a few papers that study multicast in MSN. Gao et al. [8] propose a community-based multicast routing scheme by exploiting node centrality and social community structures. This approach is applicable to the MSNs where social relationships among mobile users are long-term and less volatile than node mobility. It may not be suitable for OMSNs where social relationships are newly established and short-term. Deng et al. [6] propose a social-profile-based multicast (SPM) algorithm that uses social features in user profiles to guide multicast in MSNs. But the static social features may not capture users' dynamic contact behavior. For example, someone who puts *New York* as his *state* in his profile may actually attend a conference in *Texas*. In our previous work [23], we put forward a multicast algorithm Multi-Sosim based on dynamic social features which keep track of users' contact behavior. Simulation results show that it outperforms the SPM algorithm. In this paper, we will design new algorithms to further improve multicast efficiency.

III. Preliminary

In this section, we present the concepts of static and dynamic social features, the enhanced dynamic social features, and the calculation of nodes' social similarity based on social features to prepare for the later proposed multicast algorithms.

A. Static social features and related social similarity

Suppose we consider m social features $\langle F_1, F_2, \cdots, F_m \rangle$ in the network. We associate a node with a vector of static social feature values $\langle f_1, f_2, \cdots, f_m \rangle$ obtained from the user profile [6]. For convenience's sake, when we mention a node's social features, we mean the vector of the node's social feature values. We define the social similarity $S(x, y)$ of two nodes x and y using their static social features as the ratio of their common social feature values to all of their social feature values. For example, if x's static social feature vector is: $\langle Student, NewYork, English \rangle$ and y's static social feature vector is: $\langle Student, Texas, English \rangle$, then they have 2 social feature values $Student$ and $English$ in common out of 4 total unique social feature values $Student$, $NewYork$, $Texas$, and $English$. Therefore, their social similarity $S(x, y)$ is $\frac{2}{4} = 0.5$.

B. Dynamic social features

A node x's dynamic social features are contained in a vector $x = \langle x_1, x_2, \cdots, x_m \rangle$, where x_i $(0 \leq x_i \leq 1)$ is defined based on frequency [23] as follows:

$$x_i = \frac{M_i}{M_{total}} \tag{1}$$

Here, M_i is the number of meetings of node x with nodes having social feature value f_i, and M_{total} is the total number of nodes x has met in the history we observe.

Dynamic social features not only record if a node has certain social feature values, but also record the frequency this node has met other nodes with the same social feature values. Unlike the static ones, they are time-related and adjusted to the user contact behavior change over time. Thus we can have more accurate information to make routing decisions.

C. Enhanced dynamic social features

The above definition of dynamic social features is based on frequency, which cannot distinguish the cases, for example, if A has met 1 *Student* out of 2 people it has met in total and B has met 5 *Students* out of 10 people it has met in total in the history we observe. Both of them have the same frequency of $1/2$ to meet a *Student*, but B is more active in meeting people. To break the tie and favor the more active node, there are many ways to do it. Here, we come up with the following definition (2) for the enhanced dynamic social features which is proved to satisfy our needs in the later analysis section.

The x_i $(0 \leq x_i \leq 1)$ in node x's enhanced dynamic social features $x = \langle x_1, x_2, \cdots, x_m \rangle$ is defined as follows:

$$x_i = (\frac{M_i + 1}{M_{total} + 1})^{p_i} * (\frac{M_i}{M_{total} + 1})^{1-p_i} \tag{2}$$

Here, $p_i = \frac{M_i}{M_{total}}$, M_i and M_{total} are the same as above. The meaning of the formula is that, in the next hop, if x meets another node with the same social feature, then the meeting frequency will be $\frac{M_i+1}{M_{total}+1}$; otherwise, the meeting frequency will be $\frac{M_i}{M_{total}+1}$. Since the meeting frequency with the nodes having a certain social feature is p_i, then the probability for the first case to occur is p_i and the probability for the second case to occur is $1 - p_i$. We raise the two frequencies in the next hop to their respective powers and multiply the results.

D. Social similarity using dynamic social features

With the nodes' dynamic social features defined, we can use similarity metrics such as Tanimoto, Cosine, Euclidean, and Weighted Euclidean [14] derived from data mining [9] to calculate the social similarity $S(x, y)$ of nodes x and y. All of these metrics are normalized to the range of $[0, 1]$. We decide to use the Euclidean metric in our multicast algorithms since it does not require the calculation of additional weighting values and performs slightly better than Tanimoto and Cosine in latency in our simulations.

Euclidean similarity metric

After normalizing the original definition of the Euclidean similarity in data mining to the range of $[0, 1]$ and subtracting it from 1, it is now defined as

$$S(x, y) = 1 - \frac{\sqrt{\sum_{i=1}^{m}(y_i - x_i)^2}}{\sqrt{m}}.$$

Here is how it is used in our algorithms. Suppose we consider three social features $\langle City, Language, Position \rangle$ of the nodes in the network. Assume destination d has social feature values $\langle NewYork, English, Student \rangle$. The vector of d is set to $<1, 1, 1>$ because this is our target. Suppose there are two relay candidates x and y. We want to decide which is a better one to deliver the message to the destination. From the history of observation, 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. If we use definition (1) of the dynamic social features, node x has a vector of $x = \langle 0.7, 0.93, 0.41 \rangle$. Suppose y's vector is: $y = \langle 0.23, 0.81, 0.5 \rangle$. Using the Euclidean social similarity, $S(x, d) = 0.62$ and $S(y, d) = 0.46$. So x is more socially similar to d and therefore is more likely to deliver the message to the destination. Definition (2) of the dynamic social features can be used in the similar way.

IV. Multicast Algorithms

In this section, we present two novel multicast algorithms using enhanced dynamic social features and a new compare-split scheme based on community detection.

A. The Multi-CSDO algorithm

Our first multicast algorithm is called Multi-CSDO as shown in Fig. 1. Its basic idea is as follows: First, a source node s has a destination set to multicast a message to and s is the initial message holder or relay node x. When x meets a node y, if y is one of the destinations, y gets the message and is removed from the destination set. Next we use a compare-split scheme to make a decision of whether it is better to pass some destinations to y. Both x and y are called relay candidates in the decision. To separate the destinations into x's community or y's community, we use a community detection algorithm involving only the destination nodes based on their social similarities. The community detection algorithm we use takes a distance matrix coming from a similarity weighted graph as an input. The following are the details.

1) Similarity weighted graph and distance matrix: In Multi-CSDO, as shown in an example in Fig. 2, when a message holder x encounters a node y, we construct a similarity weighted graph involving only the destination nodes. The weight of the edges is the social similarity of the two connected destination nodes calculated using static social features (denoted by dashed edges in Fig. 2) as their dynamic social features are not known to the relay candidates in a distributed algorithm. With the similarity weighted graph, we can create a distance matrix as shown in Fig. 3 to indicate the distance between each pair of destinations. The distance between two nodes u and v is defined as $1 - S(u, v)$ here. The distance matrix will be used in the following community detection algorithm to separate the destinations into two communities.

Algorithm Multi-CSDO: community and social feature-based multicast involving destinations only in community detection

Require: The source node s and its destination set $D_s = \{d_1, d_2, \cdots, d_n\}$; s is the initial message holder x

1: **while** not all of the destinations receive the message **do**
2: On contact between a message holder x and node y:
3: **if** $y \in D_x$ **then**
4: /* Found destination y */
5: y gets the message and x removes y from D_x
6: **end if**
7: /* Compare node social similarity and split the destinations */
8: Construct a weighted graph and a distance matrix of the destination nodes only as explained in Section IV-A
9: Feed the distance matrix to the hierarchical clustering algorithm to generate two communities C_1 and C_2 as explained in Section IV-A
10: Compare the social similarity of C_1 and C_2 with x and y using enhanced dynamic social features, respectively
11: Whichever (x or y) is more socially similar to each of the communities will be the message carrier for that community
12: **end while**

Fig. 1. Our multicast algorithm Multi-CSDO

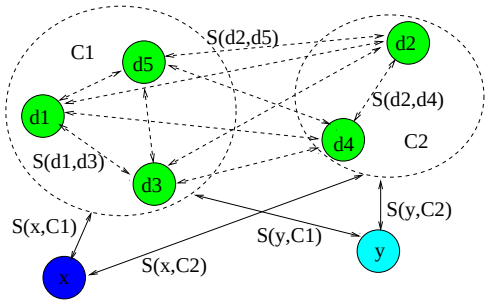


Fig. 2. The similarity weighted graph and community detection involving destination nodes only. Node x is a message holder and y is a newly met node. The green nodes are the destinations. The weight of a dashed edge is the social similarity calculated using static social features while the weight of a solid edge is the social similarity calculated using the enhanced dynamic social features. The destinations are split into two communities C_1 and C_2 based on their social similarities.

2) Community detection algorithm: Typical algorithms for community detection include minimum-cut method, Girvan-Newman algorithm, hierarchical clustering, and so on [1]. Here, we use a hierarchical clustering algorithm called complete-linkage clustering [2] to split the destinations into two communities. We choose this one because it best matches our needs and there is an existing Python package [3] available for this algorithm so that we do not have to reinvent the wheel.

The idea of the complete-linkage hierarchical community detection algorithm we adopt is as follows: At the beginning of the process, each node is in a community of its own. The communities are then sequentially combined into larger communities, until all nodes end up being in one community.

	d1	d2	d3	d4	•••
d1	0	1−S(d1,d2)	1−S(d1,d3)	1−S(d1,d4)	
d2	1−S(d1,d2)	0	1−S(d2,d3)	1−S(d2,d4)	
d3	1−S(d1,d3)	1−S(d2,d3)	0	1−S(d3,d4)	
d4	1−S(d1,d4)	1−S(d2,d4)	1−S(d3,d4)	0	
⋮					

Fig. 3. The distance matrix. The distance between nodes u and v is $1 - S(u, v)$ if $u \neq v$; otherwise 0.

At each step, the two communities separated by the shortest distance are combined. The distance between communities is defined as the distance between those two nodes (one in each community) that are farthest away from each other. We feed our distance matrix and the number of communities 2 into the package and obtain two communities as the result.

3) Destinations split: After applying the community detection algorithm, the destinations are separated into two communities C_1 and C_2. Next we decide which relay candidate, x or y, should carry the destinations in which community. We compare the social similarity of each relay candidate with each community using enhanced dynamic social features (denoted by the solid edges in Fig. 2). The social similarity between a node and a community should include all of the social feature values of the nodes involved. After calculation, whichever is more socially similar to a community will be the relay node for the destinations in that community.

In Multi-CSDO, x and y are supposed to be in different communities, which may not be true if they are socially similar. Thus, in the next section, we introduce the Multi-CSDR algorithm by incorporating both x and y in the community detection and make our decision more accurate by considering more node relationships.

B. The Multi-CSDR algorithm

Our second multicast algorithm is called Multi-CSDR (omitted due to space limit). It has a similar structure with the first algorithm, but has several differences. As shown in the example in Fig. 4, first, the community detection algorithm involves both the destination nodes and the relay candidates x and y. Thus the similarity weighted graph adds the social similarity between each relay candidate and each destination node. The social similarity between two destination nodes is still calculated using static social features and is denoted by a dashed edge in Fig. 4. However, the social similarity between a relay candidate and a destination is calculated using enhanced dynamic social features as they can be obtained and is denoted by a solid edge in Fig. 4. We still use the same community detection algorithm. But the distance matrix now also includes the distance between each relay candidate and each destination. After applying the community detection algorithm, the destinations in x's community will be carried

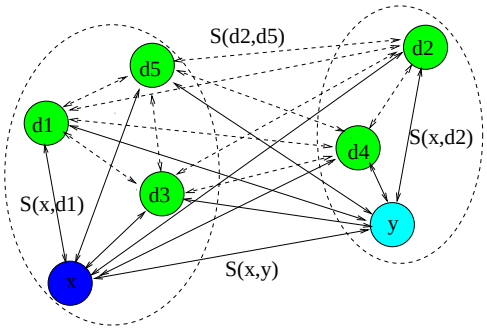


Fig. 4. The similarity weighted graph and community detection involving both destination nodes and relay candidates x and y. Node x is a message holder and y is a newly met node. The green nodes are the destinations. The weight of a dashed edge is the social similarity calculated using static social features while the weight of a solid edge is the social similarity calculated using the enhanced dynamic social features. The nodes are split into two communities based on their social similarities.

by x and those in y's will be carried by y. For other cases, for example, if x and y are in the same community, then x will still be the carrier for the original destination set.

In this algorithm, by adding the social similarity of each relay node with each destination using enhanced dynamic social features, we hope to improve the accuracy of the compare-split scheme.

V. Analysis

A. Property of dynamic social features definition (2)

Theorem 1. *Suppose node x has met M_{xi} nodes with a certain feature out of M_{xtotal} nodes it has met so far and node y has met M_{yi} nodes with the same certain feature out of M_{ytotal} nodes it has met so far. We assume they have the same meeting frequency $p_i = M_{xi}/M_{xtotal} = M_{yi}/M_{ytotal}$ with these nodes, and $M_{xtotal} \leq M_{ytotal}$. According to definition (2) of the dynamic social features, $x_i = (\frac{M_{xi}+1}{M_{xtotal}+1})^{p_i} * (\frac{M_{xi}}{M_{xtotal}+1})^{1-p_i}$ and $y_i = (\frac{M_{yi}+1}{M_{ytotal}+1})^{p_i} * (\frac{M_{yi}}{M_{ytotal}+1})^{1-p_i}$. Then $x_i \leq y_i$.*

Proof. To prove the result $x_i \leq y_i$, it is equivalent to prove that $x_i - y_i \leq 0$. Expand x_i and y_i and replace M_{xi} by $p_i M_{xtotal}$ and M_{yi} by $p_i M_{ytotal}$, it is to prove that

$$\frac{(p_i M_{xtotal}+1)^{p_i} M_{xtotal}^{1-p_i}}{M_{xtotal}+1} - \frac{(p_i M_{ytotal}+1)^{p_i} M_{ytotal}^{1-p_i}}{M_{ytotal}+1} \leq 0.$$

Multiply the two sides by $(M_{xtotal}+1)(M_{ytotal}+1)M_{xtotal}^{p_i}M_{ytotal}^{p_i}$, we get $(p_i M_{xtotal}+1)^{p_i} M_{xtotal}(M_{ytotal}+1)M_{ytotal}^{p_i} - (p_i M_{ytotal}+1)^{p_i} M_{ytotal}(M_{xtotal}+1)M_{xtotal}^{p_i} \leq 0$. Rearrange the inequality, it is to prove that

$$(\frac{p_i M_{xtotal} M_{ytotal} + M_{ytotal}}{p_i M_{xtotal} M_{ytotal} + M_{xtotal}})^{p_i} \leq \frac{M_{xtotal} M_{ytotal} + M_{ytotal}}{M_{xtotal} M_{ytotal} + M_{xtotal}}.$$

Since $M_{ytotal} \geq M_{xtotal}$, $\frac{p_i M_{xtotal} M_{ytotal} + M_{ytotal}}{p_i M_{xtotal} M_{ytotal} + M_{xtotal}} \geq 1$, so the left side is a non-decreasing function with the increase of p_i. The maximum p_i is 1, so the maximum value of the left side is $\frac{M_{xtotal} M_{ytotal} + M_{ytotal}}{M_{xtotal} M_{ytotal} + M_{xtotal}}$, which is the right side. So the left side is less or equal to the right side. This proves the theorem. This result shows that even if nodes x and y have the same frequency meeting nodes of a certain social feature, definition (2) favors the more active node to break the tie. □

B. The number of forwardings

Due to limited space, we only provide sketches of proofs for the following two theorems. Details can be found in [16].

Theorem 2. *In both Multi-CSDO and Multi-CSDR algorithms, if there is only one destination d in the destination set D, the expected number of forwardings to reach the destination is $\ln g + 1$, where g is the social similarity gap from s to d.*

Sketch of proof. The source node s has a social similarity gap g to the destination d. To reach d, the message will be delivered to a node with a smaller gap to d in each forwarding. For the convenience of later deduction, we set the gap from the source s to d to 1, the gap within which to reach d in one hop (forwarding) to β as shown in Fig. 5(a). So gap β is equal to $\frac{1}{g}$. Now the probability to reach d in 1 hop from s is β. The probability to reach d in 2 hops from s is $\int_0^{1-\beta} \frac{\beta}{1-x} dx = \beta \ln \frac{1}{\beta}$, 3 hops is $\int_0^{1-\beta} \int_{x_1}^{1-\beta} \frac{\beta}{(1-x_1)(1-x_2)} dx_2 dx_1 = \frac{\beta}{2!}(\ln \frac{1}{\beta})^2$, $\cdots$, h hops is: $\int_0^{1-\beta} \int_{x_1}^{1-\beta} \cdots \int_{x_{h-1}}^{1-\beta} \frac{\beta}{(1-x_1)(1-x_2)\cdots(1-x_{h-1})} dx_{h-1} \cdots d_{x_1} = \frac{\beta}{h!}(\ln \frac{1}{\beta})^h$, and so on. These probabilities form a distribution as their summation is 1 using the Taylor series for the exponential function e^x. Therefore, the expected number of forwardings is: $\beta \cdot 1 + \beta \ln \frac{1}{\beta} \cdot 2 + \frac{\beta}{2!}(\ln \frac{1}{\beta})^2 \cdot 3 + \cdots = 1 + (\ln \frac{1}{\beta}) \sum_{h=1}^{\infty} \frac{\beta}{(h-1)!} (\ln \frac{1}{\beta})^{h-1}$. Using the Taylor series for e^x again, it is equal to $1 + \ln \frac{1}{\beta} \cdot \beta \cdot e^{\ln \frac{1}{\beta}} = 1 + \ln \frac{1}{\beta} = \ln g + 1$. □

Theorem 3. *The expected number of forwardings in the Multi-CSDO and Multi-CSDR algorithms with $k (k > 1)$ destinations is $\sum_{i=1}^{k-1} \ln(min(g - g_i, g_i)) + \ln g + O(k)$, where $g_i (1 \leq i \leq k-1)$ is the social similarity gap from source s to destination d_i and $g_k = g$ is the social similarity gap from the source to the farthest destination d_k.*

Sketch of proof. In our algorithms, the rule of compare-split is that when a message holder with k destinations meets another node, a destination d_i should be carried by the relay candidate that has a closer social similarity gap to that destination. Let us first look at the 2-destination case as shown in Fig. 5(b). Assume the social similarity gaps from source s to the farther destination d_2 and to the closer destination d_1 are $g_2 = g$ and g_1, respectively. We know from Theorem 2 that the expected number of forwardings to reach d_2 is $\ln g + 1$. Now we calculate the extra number of forwardings needed to reach d_1 after the two destinations split. From Theorem 2, the expected number of forwardings h to reach a destination with gap g from source is $\ln g + 1$. So $g = e^{h-1}$. That means, if the message holder meets a node within the range of $[g_1 - e^0, g_1 + e^0]$, the expected number of hops to reach d_1 is $1 (h = 1)$. If the message holder meets a node within the range of $[g_1 - e^1, g_1 + e^1]$ but not within the range of $[g_1 - e^0, g_1 + e^0]$, the expected number of hops to reach d_1

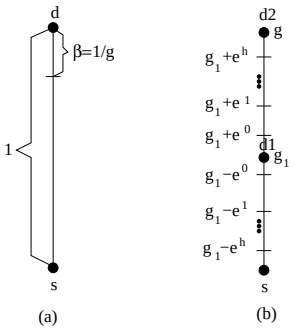


Fig. 5. (a) One destination d, whose gap to source s is 1. The range to reach d in one hop is $\beta = 1/g$. (b) Two destinations d_1 and d_2, whose gaps to s are g_1 and g, respectively. We construct the range $[g_1 - e^h, g_1 + e^h]$ around g_1 to calculate the expected number of extra forwardings to reach d_1 after spliting.

is $2(h = 2)$. In general, if the message holder meets a node within the range of $[g_1 - e^h, g_1 + e^h]$ but not within the range of $[g_1 - e^{h-1}, g_1 + e^{h-1}]$, the expected number of hops to reach d_1 is $h + 1$ and the probability to meet such a node is $\frac{2e^h}{g-g_1+e^h}$ from the gap range. Now we discuss two cases: (1). $g_1 \leq \frac{g}{2}$ and (2). $g_1 > \frac{g}{2}$.

In case (1), if the two destinations split at the $h+1$ ($h \geq 0$) hop, the expected number of extra forwardings to reach d_1 is $1 \cdot \frac{2e^0}{g-g_1+e^0} + 2 \cdot (\frac{2e^1}{g-g_1+e^1} - \frac{2e^0}{g-g_1+e^0}) + 3 \cdot (\frac{2e^2}{g-g_1+e^2} - \frac{2e^1}{g-g_1+e^1}) + \cdots + \lceil \ln g_1 \rceil (1 - \frac{2e^{\lfloor \ln g_1 \rfloor - 1}}{g-g_1+e^{\lfloor \ln g_1 \rfloor - 1}}) = \ln g_1 + O(1)$. Following the same idea, the expected number of extra forwardings in case (2) is $\ln(g - g_1) + O(1)$. So the expected number of extra forwardings to reach d_1 is $\ln(min(g - g_1, g_1)) + O(1)$. Adding the expected number of forwardings to reach d_2, the total expected number of forwardings to reach the two destinations is $\ln(min(g - g_1, g_1)) + \ln g + O(1)$.

We extend the same analysis idea to the k-destination case. The expected number of forwardings to reach the farthest destination d_k is $\ln g + 1$, and the expected number of extra forwardings to reach each other destination $d_i (i \neq k)$ is $\ln(min(g - g_i, g_i)) + \ln g + O(1)$. Then the total expected number of forwardings to reach all k destinations is $\sum_{i=1}^{k-1} \ln(min(g - g_i, g_i)) + \ln g + O(k)$. □

C. The number of copies

Theorem 4. *The number of copies produced by the Multi-CSDO and Multi-CSDR algorithms is k, where k is the number of destinations in the multicast set.*

Proof. It is trivial to see that each split of the destinations will produce one extra copy. There are k destinations, so it takes $k - 1$ splits to separate the k destinations into individual ones. Adding the original one copy, the number of copies produced by the Multi-CSDO and Multi-CSDR algorithms is k. □

VI. Simulations

In this section, we evaluate the performance of our multicast algorithms by comparing them with the existing ones using a custom simulator written in Python. The simulations were conducted using a real conference trace [15] reflecting an OMSN created at IEEE Infocom 2006 in Miami. There are very few available OMSN traces containing both usable node social features and node contact information. Infocom 2006 trace has been widely used to test routing algorithms in mobile social networks [6], [23]. The trace recorded conference attenders' encounter history using Bluetooth small devices (iMotes) for four days at the conference. The trace dataset consists of two parts: *contacts* between iMote devices that were carried by participants and self-reported *social features* of the participants collected using a questionnaire form. The six social features extracted from the dataset were *Affiliation, City, Nationality, Language, Country,* and *Position*. In this trace, 62 nodes with complete social feature information were considered in our multicast process.

A. Comparison with existing algorithms

We compared the following multicast protocols.

1) *The Epidemic Algorithm* (Epidemic) [17]: The message is spread epidemically throughout the network until it reaches all of the destinations.
2) *The Social-Similarity-based Multicast Algorithm* (Multi-Sosim) [23]: The multicast algorithm based on dynamic social features in our previous work.
3) *The Enhanced Social-Similarity-based Multicast Algorithm* (E-Multi-Sosim): The multicast algorithm that applies enhanced dynamic social features to Multi-Sosim.
4) *The Community and Social Feature-based Multicast Algorithm involving destinations only in community detection* (Multi-CSDO): Our first multicast algorithm proposed in this paper using enhanced dynamic social features and community detection.
5) *The Community and Social Feature-based Multicast Algorithm involving both destinations and relay candidates in community detection* (Multi-CSDR): Our second multicast algorithm proposed in this paper using enhanced dynamic social features and community detection.

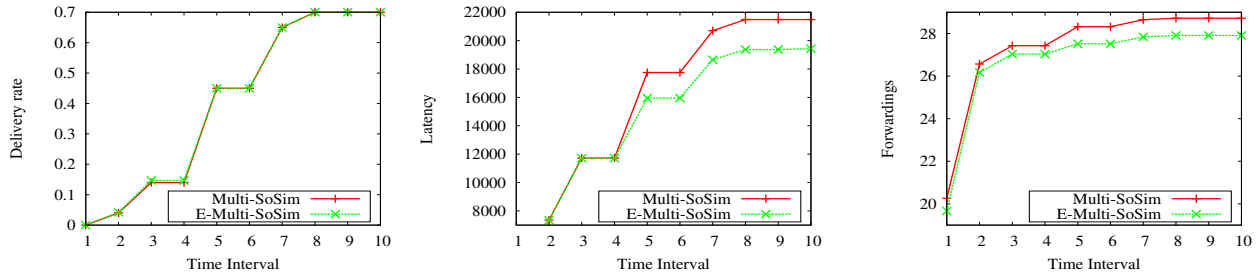
B. Evaluation metrics

We use three important metrics to evaluate the performance of the multicast algorithms. A *successful multicast* is the one that successfully delivers the message to all of the destinations.

1) *Delivery ratio*: The ratio of the number of successful multicasts to the number of total multicasts generated.
2) *Delivery latency*: The time from the start of multicast to when all of the multicast destinations have received the message.
3) *Number of forwardings*: The number of hops needed to deliver a message to all of the multicast destinations.

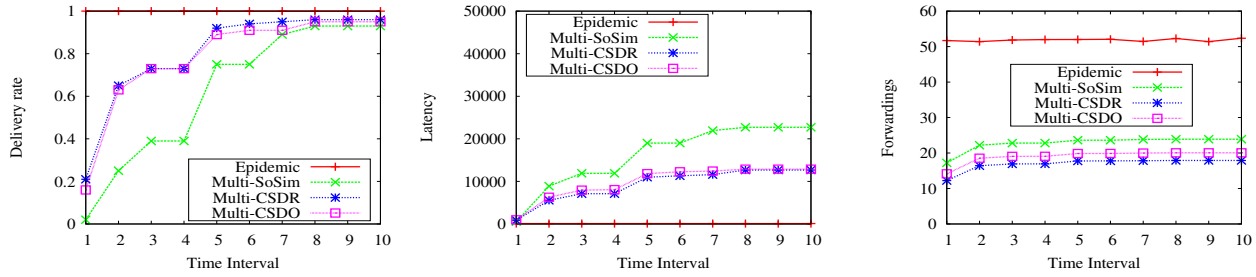
C. Simulation setup

In our simulations, we divided the whole trace time into 10 intervals. Thus, 1 time interval is $1/10$ of the total time length. For each algorithm, we tried 5 and 10 destinations. In each experiment, we randomly generated a source and its destination set. Since the whole trace only contains four days of node contact history, the time interval we observed to



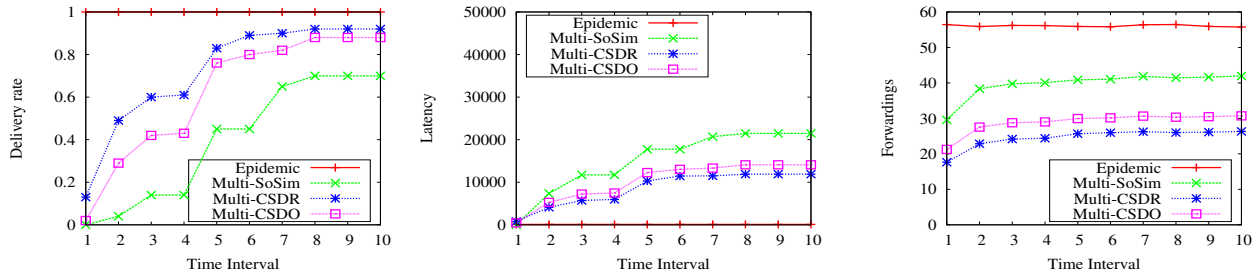
(a) Delivery ratio (b) Delivery latency (c) Number of forwardings

Fig. 6. Comparison of Multi-Sosim and E-Multi-Sosim with 10 destinations using all devices in the trace



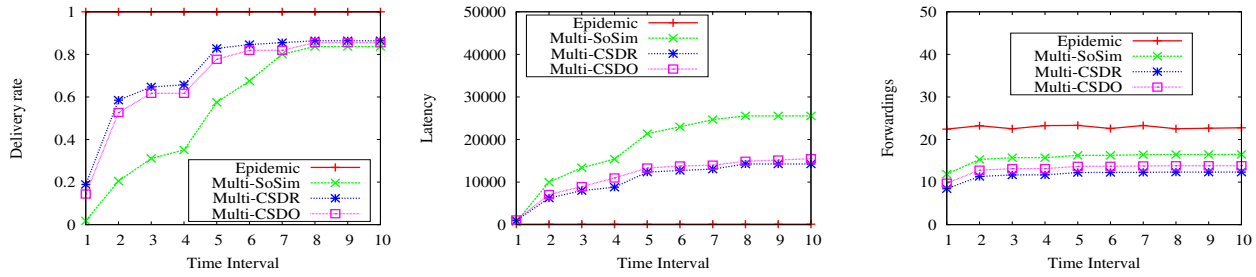
(a) Delivery ratio (b) Delivery latency (c) Number of forwardings

Fig. 7. Comparison of different algorithms with 5 destinations using all devices in the trace



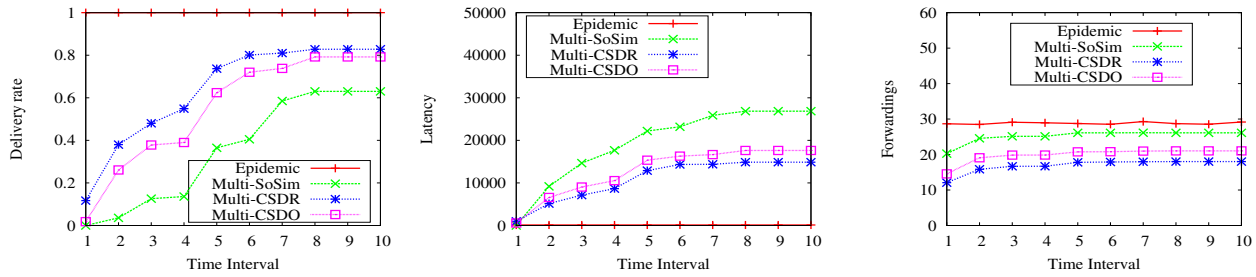
(a) Delivery ratio (b) Delivery latency (c) Number of forwardings

Fig. 8. Comparison of different algorithms with 10 destinations using all devices in the trace



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

Fig. 9. Comparison of different algorithms with 5 destinations in sparse network



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

Fig. 10. Comparison of different algorithms with 10 destinations in sparse network

calculate the dynamic and enhanced dynamic social features was counted from the beginning of the trace up until the time we needed to make a routing decision. For the community detection algorithm, we adopted the Python package available at [3] for the complete-linkage hierarchical clustering algorithm. We ran each algorithm 300 times and averaged the results.

D. Simulation results

The simulation results comparing Multi-Sosim and E-Multi-Sosim are shown in Fig. 6 with 10 destination nodes. These two algorithms have similar delivery rates with E-Multi-Sosim slightly better. But E-Multi-Sosim clearly outperforms Multi-Sosim in latency and number of forwardings. These results justify the enhancement of dynamic social features.

The simulation results comparing our algorithms with others using 5 and 10 destinations are shown in Figs. 7 and 8, respectively. For the Epidemic algorithm, as expected, it has the highest delivery ratio (100%) and lowest delivery latency (almost close to 0) but highest number of forwardings.

With both 5 and 10 destinations, Multi-CSDO and Multi-CSDR consistently outperform Multi-Sosim in terms of delivery rate, latency, and number of forwardings. This means that adding the social relationships among destinations in the compare-split scheme can facilitate multicast. Furthermore, Multi-CSDR has better delivery rate, lower latency, and lower number of forwardings than Multi-CSDO, which verifies that considering the social relationship between each relay candidate and each destination, and calculating their social similarity using enhanced dynamic social features can improve multicast performance.

In summary, these results confirm that obtaining more accurate dynamic information and using better compare-split schemes can make multicast more efficient.

E. Sparse Networks

In the experiments above, we used all 62 available nodes in the trace. We also tested our algorithms on a smaller random subset of the trace with 30 nodes which produces a sparse network. The results from the sparse network shown in Figs. 9 and 10 are consistent with those in the denser network.

VII. Conclusion

In this paper, we proposed two novel community and social feature-based multicast algoirthms Multi-CSDR and Multi-CSDO for OMSNs. In the algorithms, we used enhanced dynamic social features to more accurately capture nodes' contact behavior, considered more social relationships among nodes, and proposed compare-split schemes based on community detection to select the best relay node for each destination in each hop to improve multicast efficiency. Analysis of the algorithms was given and simulation results using a real trace of an OMSN showed that our new algorithms consistently outperform the existing one in delivery rate, latency, and number of forwardings. In the future, we will continue improving the efficiency of our algorithms and testing them using more traces in OMSNs as they become available.

Acknowledgements

This research was supported in part by DoD in partnership with NSF REU grant 1156712, NSF CNS grant 1305302, NSF ACI grant 1440637, and China NSF grant 61373128.

References

[1] Community structure. http://en.wikipedia.org/wiki/Community_structure.
[2] Complete-linkage clustering. http://en.wikipedia.org/wiki/Complete_linkage_clustering.
[3] Hierarchical clustering. http://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html.
[4] M. Chuah and P. Yang. Context-aware multicast routing scheme for disruption tolerant networks. *Journal of Ad Hoc and Ubiquitous Computing*, 4(5):269–281, 2009.
[5] E. Daly and M. Haahr. Social network analysis for routing in disconnected delay-tolerant MANETs. In *IEEE MobiHoc*, pages 32–40, 2007.
[6] X. Deng, L. Chang, J. Tao, J. Pan, and J. Wang. Social profile-based multicast routing scheme for delay-tolerant networks. In *IEEE ICC*, pages 1857–1861, 2013.
[7] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, and Y. Sun. Geo-community-based broadcasting for data dissemination in mobile social networks. *IEEE Trans. on Parallel and Distributed Systems*, 24(4):734–743, 2013.
[8] W. Gao, Q. Li, B. Zhao, and G. Cao. Multicasting in delay tolerant networks: a social network perspective. In *ACM MobiHoc*, 2009.
[9] J. W. Han, M. Kamber, and J. Pei. *Data Mining: concepts and techniques*. Morgan Kaufmann, MA, USA, 2012.
[10] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *IEEE MobiHoc*, pages 241–250, 2008.
[11] B. Jedari and F. Xia. A Survey on Routing and Data Dissemination in Opportunistic Mobile Social Networks. http://arxiv.org/abs/1311.0347.
[12] U. Lee, S. Y. Oh, Lee K.-W., and M. Gerla. Relaycast: scalable multicast routing in delay tolerant networks. In *IEEE ICNP*, pages 218–227, 2008.
[13] M. Mongiovi, A. K. Singh, X. Yan, B. Zong, and K. Psounis. Efficient multicasting for delay tolerant networks using graph indexing. In *IEEE INFOCOM*, 2012.
[14] D. Rothfus, C. Dunning, and X. Chen. Social-similarity-based routing algorithm in delay tolerant networks. In *IEEE ICC*, pages 1862–1866, 2013.
[15] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau. Crawdad trace cambridge/haggle/imote/infocom2006 (v.2009-05-29). http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006, May 2009.
[16] C. Shang, B. Wong, X. Chen, W. Z. Li, and S. Oh. Community and social feature-based multicast in opportunistic mobile social networks. Technical report, Dept. of Comp. Sci., Texas State Univ., 2015.
[17] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, Dept. of Comp. Sci., Duke Univ., 2000.
[18] N. Vastardis and K. Yang. Mobile Social Networks: Architectures, Social Properties, and Key Research Challenges. *IEEE Communications Surveys and Tutorials*, 15(3):1355–1371, 2013.
[19] Y. Wang and J. Wu. A dynamic multicast tree based routing scheme without replication in delay tolerant networks. *Journal of Parallel and Distributed Computing*, 72(3):424–436, 2012.
[20] J. Wu and Y. Wang. *Opportunistic Mobile Social Networks*. Taylor & Francis, 2014.
[21] Y. Xi and M. Chuah. An encounter-based multicast scheme for disruption tolerant networks. *Comp. Comm.*, 32(16):1742–1756, 2009.
[22] M. Xiao, J. Wu, and L. Huang. Community-Aware Opportunistic Routing in Mobile Social Networks. *IEEE Trans. on Computers*, 63(7):1682–1695, 2014.
[23] Y. Xu and X Chen. Social-similarity-based multicast algorithm in impromptu mobile social networks. In *IEEE Globecom*, 2014.
[24] W. Zhao, M. Ammar, and E. Zegura. Muticasting in delay tolerant networks: semantic models and routing algorithms. In *ACM WDTN*, pages 268–275, 2005.
[25] H. Zhou, J. Chen, J. Fan, Y. Du, and S. K. Das. ConSub: Incentive-Based Content Subscribing in Selfish Opportunistic Mobile Networks. *IEEE Jnl. on Selected Areas in Communications*, 31(9):669–679, 2013.
[26] H. Zhou, J. Chen, H. Y. Zhao, W. Gao, and P. Cheng. On Exploiting Contact Patterns for Data Forwarding in Duty-Cycle Opportunistic Mobile Networks. *IEEE Trans. on Veh. Tech.*, 62(9):4629–4642, 2013.