Efficient Multicast Algorithms in Opportunistic Mobile Social Networks using Community and Social Features

Xiao Chen [a,*], Charles Shang [b], Britney Wong [c], Wenzhong Li [d], Suho Oh [e]

[a] Dept. of Computer Science, Texas State University, San Marcos, TX 78666, United States
[b] Dept. of Computer Science, Univ. of Illinois at Urbana-Champaign, Champaign, IL, 61801, United States
[c] Dept. of Computer Science, Cornell University, Ithaca, NY, 14850, United States
[d] Dept. of Computer Science and Technology, Nanjing University, China, 210093, China
[e] Dept. of Mathematics, Texas State University, San Marcos, TX 78666, United States

ABSTRACT

Opportunistic Mobile Social Networks (OMSNs), formed by people moving around carrying mobile devices, enhance spontaneous communication among users that opportunistically encounter each other without additional infrastructure. Multicast is an important communication service in OMSNs. Most of the existing multicast algorithms neglect or adopt static social factors that are inadequate to catch nodes' dynamic contact behavior. In this paper, we introduce dynamic social features and its enhancement to capture nodes' contact behavior, consider more social relationships among nodes, and adopt community structure in the multicast compare-split schemes to select the best relay nodes to improve multicast efficiency. We propose two multicast algorithms based on these new features. The first one Multi-CSDO involves destination nodes only in community detection while the second one Multi-CSDR involves both the destination nodes and the relay candidates in community detection. The analysis of the algorithms is given and simulation results using two real OMSN traces show that our new algorithms outperform the existing ones in delivery rate, latency, and number of forwardings.

1. 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 [1–8]. Unlike popular online social networks such as Facebook and LinkedIn, the OMSNs we discuss here are a special kind of delay tolerant networks (DTNs) where the communication takes place on-the-fly by the opportunistic contacts among mobile users in a lightweight mechanism via local wireless bandwidth such as Bluetooth or WiFi without a network infrastructure [5,9,10]. Due to the time-varying network topology of OMSNs, an 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, the message 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 [11–15] without social characteristics. There are a few multicast algorithms involving social factors [16,17] 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, ..., F_i can refer to *Nationality, City, Language, Affiliation*, and so on. Each social feature F_i can take multiple values f_1, f_2, ..., f_i. For example, a social feature F_i can be *Language* and its values can be *English, Spanish*, and so on. Social features and their values can be obtained from user profiles when they register for an event. Deng et al. propose a social profile-based multicast algorithm (SPM) [16] based on static social features in user profiles. In our previous work [17], we argued that static social features may not always reflect nodes' dynamic contact behavior and introduced dynamic social features to capture nodes' contact frequency with

* Corresponding author.
E-mail address: xc10@txstate.edu (X. Chen).

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 results in a tree structure. A compare-split scheme to determine the separation point is critical to the efficiency of a multicast. In multicast, when a message holder x meets another node y, they become *relay candidates* as one of them or both will be responsible for relaying the message to the destinations.

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 case when two nodes have the same meeting frequency with nodes having a certain social feature. Thus we can 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. Different from the community structure where node social relationships are long-term and less volatile than node mobility in several social-aware routing schemes [18–20], 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 a 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. We provide theoretical analysis to the algorithms and simulation results show that our new algorithms outperform the existing ones in terms of delivery rate, latency, and the number of forwardings.

The rest of the paper is organized as follows: Section 2 references the related works; Section 3 introduces the preliminaries; Section 4 presents our new multicast algorithms; Section 5 gives the analysis of the algorithms; Section 6 shows the simulation results; and Section 7 is the conclusion.

2. Related works

A multicast algorithm in OMSNs can be implemented using rudimentary approaches such as Epidemic routing [21], 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. [15] introduce some new semantic models for multicast and conclude that the group-based strategy is suitable for multicast in DTNs. Lee et al. [11] 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. [14] present an encounter-based multicast routing, and Chuah et al. [22] develop a context-aware adaptive multicast routing scheme. Mongiovi et al. [12] use graph indexing to minimize the remote communication cost of multicast. Wang et al. [13] 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 Mobile Social Networks (MSNs). Gao et al. [19] 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. [16] propose a social-profile-based multicast (SPM) algorithm that uses social features in user profiles to guide multicast in MSNs. Yet the static social features may not fully 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 [17], we put forward a multicast algorithm Multi-Sosim based on dynamic social features that keep track of users' contact behavior. Simulation results show that it outperforms SPM. In this paper, we will design new algorithms to further improve multicast efficiency.

3. Preliminaries

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

3.1. Static social features and related social similarity

Suppose we consider m social features $\langle F_1, F_2, \ldots, F_m \rangle$ in the network. We associate a node with a vector of static social feature values $< f_1, f_2, \cdots, f_m >$ obtained from the user profile [16]. 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: $< Student, NewYork, English >$ and y's static social feature vector is: $< Student, Texas, English >$, 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, $S(x, y)$ is $\frac{2}{4} = 0.5$.

3.2. 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 [17] 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 change of user contact behavior over time. Thus we can have more accurate information to make routing decisions.

3.3. Enhanced dynamic social features

The above frequency-based dynamic social features cannot distinguish the case, for example, if A has met 1 *Student* out of 2 people it has met and B has met 5 *Students* out of 10 people it has

met 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 favor the more active node, there are many ways to do it. Here, we come up with Definition 2 for the enhanced dynamic social features which will be 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:

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

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.

3.4. 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 [23] derived from data mining [24] to calculate the social similarity $S(x, y)$ of nodes x and y. 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 as evidenced by our experiments in the later simulation section.

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 $< City, Language, Position >$ of the nodes in the network. Assume destination d has social feature values $< NewYork, English, Student >$. 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 who 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.

4. Multicast algorithms

Next, we present two novel multicast algorithms using enhanced dynamic social features and new compare-split schemes based on community detection.

4.1. 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 and also the 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 and 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.

4.1.1. Similarity weighted graph and distance matrix

In Multi-CSDO, as shown in Fig. 2(a), when a message holder x encounters a node y, we construct a similarity weighted graph involving only the destination nodes. The weight of each edge is the social similarity of the two connected destination nodes calculated using static social features (denoted by dashed edges in Fig. 2(a)) because their dynamic social features are not known to the relay candidates in a distributed environment. Next we create a distance matrix as shown in Fig. 2(b) to represent the social difference or distance between each pair of destinations. The social distance between two destinations d_i and d_j is defined as $1 - S(d_i, d_j)$. The distance matrix will be used in the community detection algorithm to separate the destinations into two communities.

4.1.2. Community detection algorithm

Typical algorithms for community detection include the minimum-cut method, Girvan-Newman algorithm, hierarchical clustering, and so on [25]. Here, we use a hierarchical clustering algorithm called complete-linkage clustering [26] to split the destinations into two communities. We choose this one because it takes the distances between node pairs and the number of communities as inputs so that we can readily input our distance matrix and the number of communities into the algorithm. Better still, there is an existing Python package [27] 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. 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.

4.1.3. 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(a)). 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 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.

Algorithm Multi-CSDO: community and social feature-based multi-cast 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 as explained in Section 4.1.1

9:      Feed the distance matrix to the hierarchical clustering algorithm to generate two communities C_1 and C_2 as explained in Section 4.1.2

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.
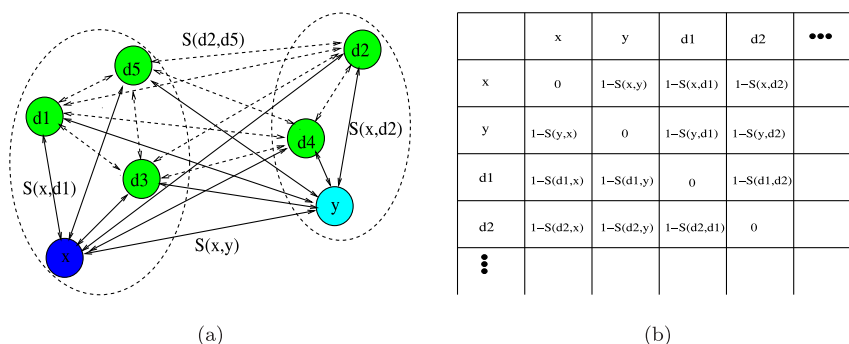

(a)   (b)
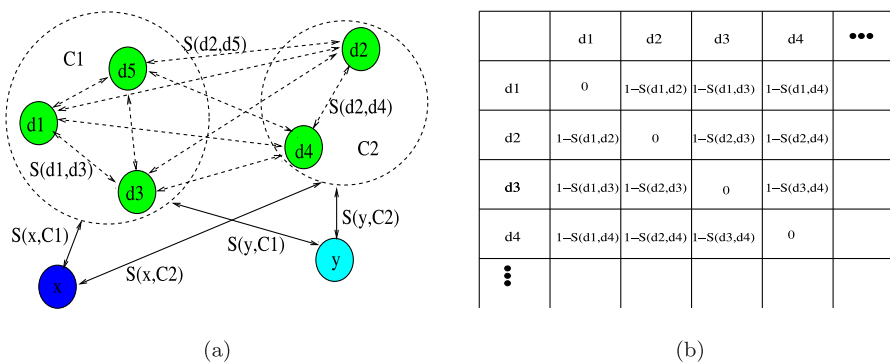
Fig. 2. (a) The similarity weighted graph and community detection in Algorithm Multi-CSDO involving destination nodes only. (b) The distance matrix in Algorithm Multi-CSDO, where d_1, d_2, $\cdots$ are destinations. The distance between any two nodes u and v is $1 - S(u, v)$ if $u \neq v$; otherwise 0.

4.2. The Multi-CSDR algorithm

Our second multicast algorithm Multi-CSDR has a similar structure to the first algorithm (omitted due to space), but has several differences. As shown in Fig. 3(a), first, the community detection algorithm involves both the destination nodes and 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. 3(a). 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. 3(a). We still use the same community detection algorithm. But the distance matrix now also includes the distance between each relay candidate and each destination as shown in Fig. 3(b). After applying the community detection algorithm, the destinations in x's community will be carried by x and those in y's will be carried by y. For other cases, x will still be the carrier for the original destination set. In this algorithm, we hope to improve the accuracy of the compare-split scheme by adding more social relationships among nodes.

	x	y	d1	d2	•••
x	0	1−S(x,y)	1−S(x,d1)	1−S(x,d2)	
y	1−S(y,x)	0	1−S(y,d1)	1−S(y,d2)	
d1	1−S(d1,x)	1−S(d1,y)	0	1−S(d1,d2)	
d2	1−S(d2,x)	1−S(d2,y)	1−S(d2,d1)	0	
⋮					

(a)   (b)

Fig. 3. (a) The similarity weighted graph and community detection in Algorithm Multi-CSDR involving both destination nodes and relay candidates x and y. (b) The distance matrix in Algorithm Multi-CSDR, where x and y are relay candidates, and d_1, d_2, $\cdots$ are destinations. The distance between any two nodes u and v is $1 − S(u, v)$ if $u \neq v$; otherwise 0.

5. Analysis

5.1. 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 proving that $x_i - y_i \leq 0$. Expanding x_i and y_i and replacing M_{xi} by $p_i M_{xtotal}$ and M_{yi} by $p_i M_{ytotal}$, we need to show that $\frac{(p_i M_{xtotal}+1)^{p_i} M_{xtotal}^{1-p_i}}{M_{xtotal}+1} - \frac{(p_i M_{ytotal}+1)^{p_i} M_{ytotal}^{1-p_i}}{M_{ytotal}+1} \leq 0$. Multiplying the two sides by $(M_{xtotal}+1)(M_{ytotal}+1)M_{xtotal}^{p_i} M_{ytotal}^{p_i}$, we get $(p_i M_{xtotal}+1)^{p_i} M_{xtotal}(M_{ytotal}+1)M_{ytotal}^{p_i} - (p_i M_{ytotal}+1)^{p_i} M_{ytotal}(M_{xtotal}+1)M_{xtotal}^{p_i} \leq 0$. Rearranging the inequality, we need to prove that $(\frac{p_i M_{xtotal} M_{ytotal}+M_{ytotal}}{p_i M_{xtotal} M_{ytotal}+M_{xtotal}})^{p_i} \leq \frac{M_{xtotal} M_{ytotal}+M_{ytotal}}{M_{xtotal} M_{ytotal}+M_{xtotal}}$. Let us look at the left side first. Since $M_{ytotal} \geq M_{xtotal}$, $\frac{p_i M_{xtotal} M_{ytotal}+M_{ytotal}}{p_i M_{xtotal} M_{ytotal}+M_{xtotal}} \geq 1$ holds. Thus 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 $\leq$ 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 breaks the tie by favoring the more active node. □

5.2. The number of forwardings

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

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 source s to d to 1, the gap within which to reach d in one hop (forwarding) to β as shown in Fig. 4(a). So gap β is equal to $\frac{1}{g}$.

Now let us calculate the probability to reach d in h hops from s. The probability to reach d in 1 hop from s is β.
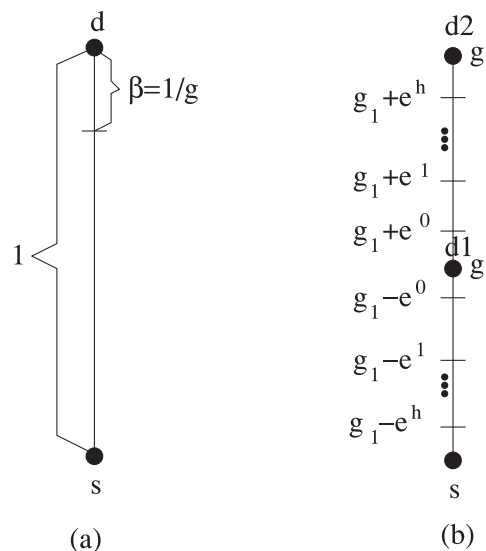
Fig. 4. (a) One destination d, whose gap to source s is 1. The range to reach d in one hop is $\beta = 1/g$. (b) Two destinations d_1 and d_2, whose gaps to s are g_1 and g, respectively.

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: $\int_0^{1-\beta} \int_{x_1}^{1-\beta} \cdots \int_{x_{h-1}}^{1-\beta} \frac{\beta}{(1-x_1)(1-x_2)\cdots(1-x_{h-1})} dx_{h-1} \cdots dx_1 = \frac{\beta}{h!} (\ln \frac{1}{\beta})^h$, etc. These probabilities form a distribution as their summation $\sum_{h=0}^{h} \frac{\beta}{h!} (\ln \frac{1}{\beta})^h$ is 1 by 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 Multi-CSDO and Multi-CSDR 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.*

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. 4(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 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

$$1 \cdot \frac{2e^0}{g - g_1 + e^0} + 2 \cdot \left(\frac{2e^1}{g - g_1 + e^1} - \frac{2e^0}{g - g_1 + e^0} \right)$$

$$+ 3 \cdot \left(\frac{2e^2}{g - g_1 + e^2} - \frac{2e^1}{g - g_1 + e^1} \right)$$

$$+ \cdots + \lceil \ln g_1 \rceil \left(1 - \frac{2e^{\lfloor \ln g_1 \rfloor - 1}}{g - g_1 + e^{\lfloor \ln g_1 \rfloor - 1}} \right) = \lceil \ln g_1 \rceil$$

$$- \sum_{h=0}^{\lfloor \ln g_1 \rfloor - 1} \frac{2e^h}{g - g_1 + e^h}.$$

From $g_1 \leq \frac{g}{2}$ and $e^{\lfloor \ln g_1 \rfloor} \leq g_1$, we have $\sum_{h=0}^{\lfloor \ln g_1 \rfloor - 1} \frac{2e^h}{2(g - g_1)} \leq \sum_{h=0}^{\lfloor \ln g_1 \rfloor - 1} \frac{2e^h}{g - g_1 + e^h} \leq \sum_{h=0}^{\lfloor \ln g_1 \rfloor - 1} \frac{2e^h}{g - g_1}$. That is, $\frac{1}{2} \frac{2(g_1 - 1)}{(g - g_1)(e - 1)} \leq \sum_{h=0}^{\lfloor \ln g_1 \rfloor - 1} \frac{2e^h}{g - g_1 + e^h} \leq \frac{2(g_1 - 1)}{(g - g_1)(e - 1)}$.

Again from $g_1 \leq \frac{g}{2}$, we have $\frac{1}{2} \cdot \frac{2}{e - 1} \leq \sum_{h=0}^{\lfloor \ln g_1 \rfloor - 1} \frac{2e^h}{g - g_1 + e^h} \leq \frac{2}{e - 1}$. This means that $\sum_{h=0}^{\lfloor \ln g_1 \rfloor - 1} \frac{2e^h}{g - g_1 + e^h}$ is a constant. So the expected number of extra forwardings to reach d_1 is $\ln g_1 + O(1)$.

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

Combining cases (1) and (2), the expected number of extra forwardings to 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 of the k destinations is $\sum_{i=1}^{k-1} \ln(min(g - g_i, g_i)) + \ln g + O(k)$. □

5.3. 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.*

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. □

6. Simulations

In this section, we evaluate the performance of our multicast algorithms by comparing them with the existing ones. Since the existing DTN simulators such as THE ONE [28] only produce simulated node behavior and do not generate social features, we used the following two real traces named Infocom 2006 [29] and unical/socialblueconn 2015 [30] reflecting OMSNs created at IEEE Infocom 2006 in Miami and on the campus of University of Calabria in Italy, respectively. We wrote a customized simulator in Python to apply the traces to our algorithms.

6.1. Traces

6.1.1. The infocom 2006 trace

The Infocom 2006 trace has been widely used to test routing algorithms in mobile social networks [16,17]. 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.

6.1.2. The unical/socialblueconn 2015 trace

This trace collects Bluetooth encounter records of 15 students on the University of Calabria campus in Italy using an ad-hoc Android application called SocialBlueConn. The trace dataset consists of the *contacts* between Bluetooth devices carried by the participating students and their *social profiles* including Facebook friends and self-declared interests. There are 9 interest categories labeled from *A* to *I* representing their preferred transportation methods, sports, music, cinema, etc. We used the students' self-declared interests as their social features. In this trace, 15 nodes with social features were considered in our multicast process.

6.2. Algorithms compared

We compared the following multicast protocols.

1. *The Epidemic Algorithm* (Epidemic) [21]: The message is spread epidemically throughout the network until it reaches all of the destinations.
2. *The Social-Profile-based Multicast Routing Algorithm* (SPM) [16]: The multicast algorithm based on static social features in user profiles.
3. *The Social-Similarity-based Multicast Algorithm* (Multi-Sosim) [17]: The multicast algorithm based on dynamic social features in our previous work.
4. *The Enhanced Social-Similarity-based Multicast Algorithm* (E-Multi-Sosim): The multicast algorithm that applies enhanced dynamic social features to Multi-Sosim.
5. *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.
6. *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.
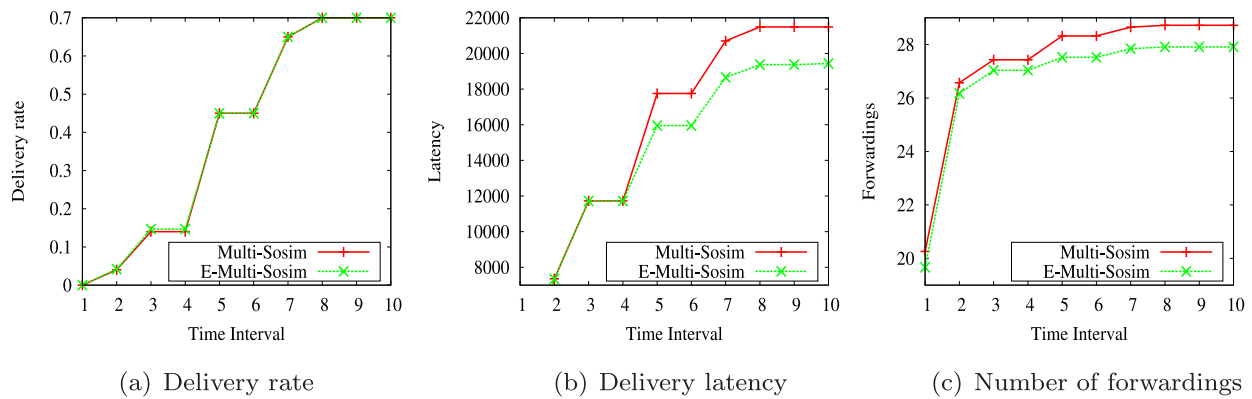
Fig. 5. Comparison of Multi-Sosim and E-Multi-Sosim with 10 destinations using Infocom trace.

6.3. Evaluation metrics

We used three important metrics to evaluate the performance of the multicast algorithms. Since a multicast involves multiple destinations, we define a *successful multicast* as the one that successfully delivers the message to all of the destinations. The three metrics are: (1) *Delivery rate*: 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.

6.4. Simulation setup and results using the infocom 2006 trace

In this simulation, 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, respectively. In each experiment, we randomly generated a source and its destination set. The time interval we observed to 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 [27] for the complete-linkage hierarchical clustering algorithm. We ran each algorithm 300 times and averaged the performance results.

In the simulation, we tried the Tanimoto, Cosine, Euclidean, and Weighted Euclidean similarity metrics to calculate the social similarity of two nodes in the algorithms where the dynamic social features is used. The results show that these similarity metrics produce little difference in the delivery rate, delivery latency, and number of forwardings among the algorithms. So in the following simulation figures, we present the results of using the Euclidean metric in the multicast algorithms since it is simpler than the Weighted Euclidean and has 0.9% and 0.2% lower latency than Tanimoto and Cosine, respectively.

The simulation results comparing Multi-Sosim and E-Multi-Sosim using 10 destinations are shown in Fig. 5. 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. Table 1 presents the latency samples of Multi-Sosim and E-Multi-Sosim with 95% confidence interval corresponding to Fig. 5(b). 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. 6 and 7, respectively. The Epidemic algorithm has the highest delivery rate (100%) and lowest delivery latency (almost close to 0) but highest number of forwardings. The improvement of Multi-Sosim over SPM con-

firms that using dynamic social features can more accurately capture node encounter behavior than using the static social features in OMSNs. With both 5 and 10 destinations, Multi-CSDO and Multi-CSDR consistently outperform Multi-Sosim in 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 further improve multicast performance.

We also tested our algorithms on a smaller random subset of the trace with 30 nodes that produces a sparse network. The results from the sparse network shown in Figs. 8 and 9 are consistent with those in the denser network.

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

6.5. Simulation setup and results using the unical/socialblueconn 2015 trace

In this simulation, we used a similar setting as that of the previous trace except that since it is a smaller trace, we divided the whole trace time into 5 intervals and tried 3 and 5 multicast destinations.

The simulation results using 3 and 5 destinations are shown in Figs. 10 and 11, respectively. The results of different destination numbers are consistent in terms of delivery rate, latency, and number of forwardings. For the delivery rate, Epidemic is the highest with 100% success rate. Multi-CSDR is the next, and then the Multi-CSDO, E-Multi-Sosim, Multi-Sosim, and SPM. For the latency, Epidemic is the lowest and SPM is the highest. E-Multi-Sosim, Multi-Sosim, Multi-CSDO, and Multi-CSDR all outperform SPM in latency with Multi-CSDR the lowest, then Multi-CSDO, E-Multi-Sosim, and Multi-Sosim. For the number of forwardings, from the highest to the lowest are Epidemic, SPM, Multi-Sosim, E-Multi-Sosim, Multi-CSDO, and Multi-CSDR. These results reaffirm that using more accurate dynamic information and better compare-split schemes can improve multicast efficiency.

7. 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
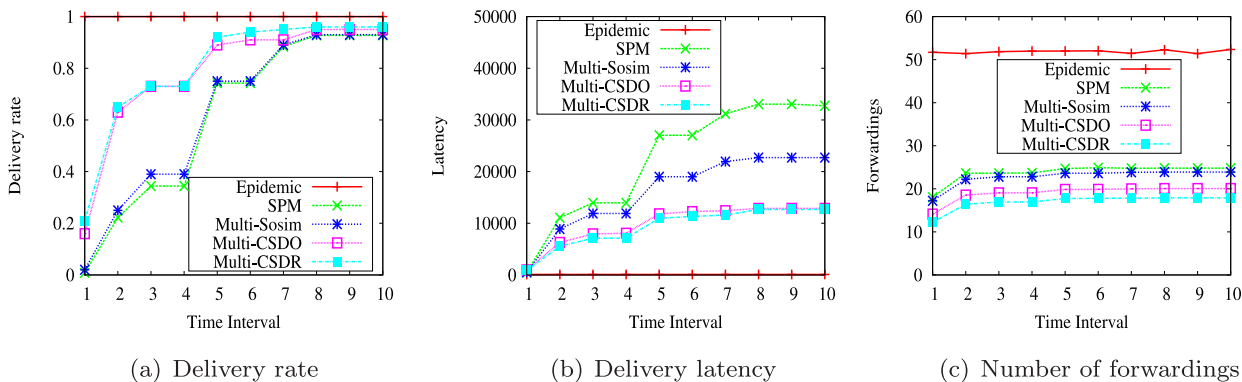
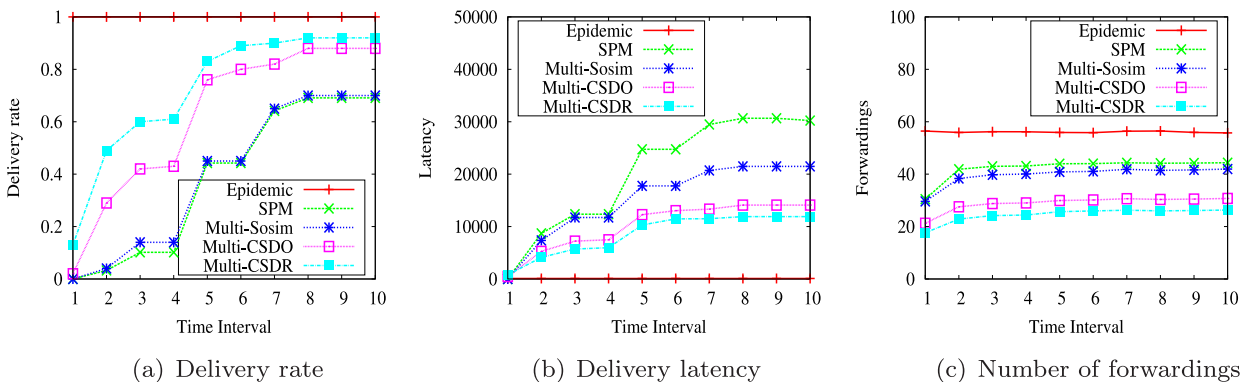Fig. 6. Comparison of algorithms with 5 destinations using Infocom trace.

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

Fig. 7. Comparison of algorithms with 10 destinations using Infocom trace.

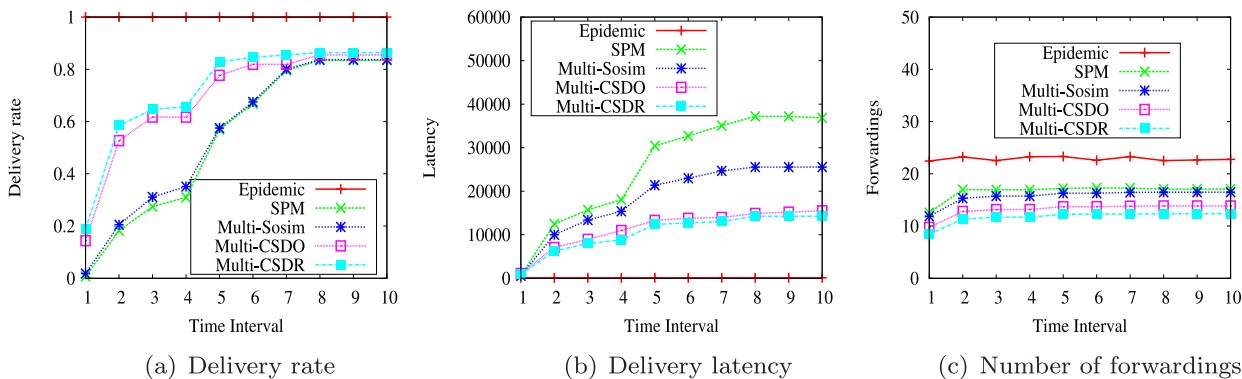(a) Delivery rate (b) Delivery latency (c) Number of forwardings

Fig. 8. Comparison of algorithms with 5 destinations in sparse network using Infocom trace.

(a) Delivery rate (b) Delivery latency (c) Number of forwardings
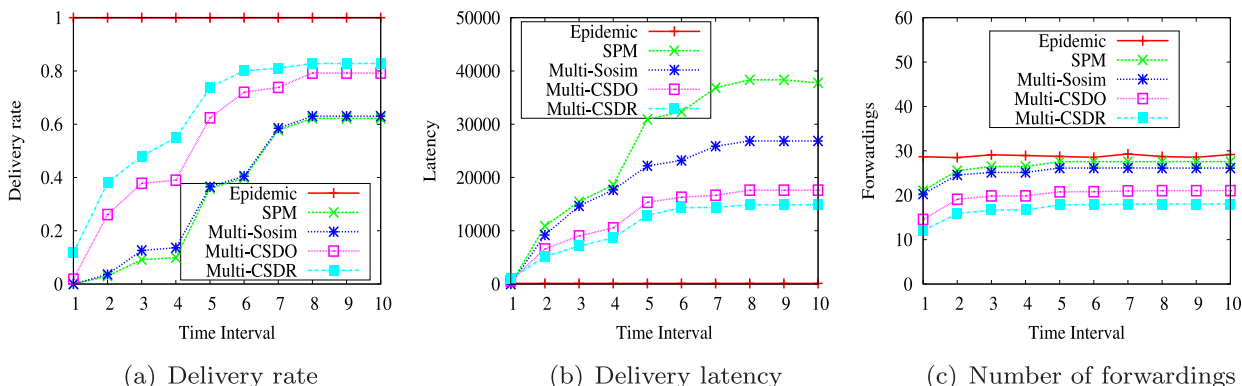
Fig. 9. Comparison of algorithms with 10 destinations in sparse network using Infocom trace.

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

Table 1

The latency samples of Multi-Sosim and E-Multi-Sosim with 95% confidence interval corresponding to Fig. 5(b).

Time interval	Multi-Sosim Avg [Min, Max]	E-Multi-Soim Avg [Min, Max]
2	7354.09 [6331.79, 7733.50]	7343.02 [6482.472, 7881.93]
4	11726.84 [10673.13, 12907.80]	11717.45 [10733.54, 13050.73]
6	17752.63 [16449.79, 19833.12]	15956.06 [14873.26, 17914.35]
8	21483.89 [19436.04, 23530.30]	19374.14 [18242.57, 22085.47]

(a) Delivery rate   (b) Delivery latency   (c) Number of forwardings
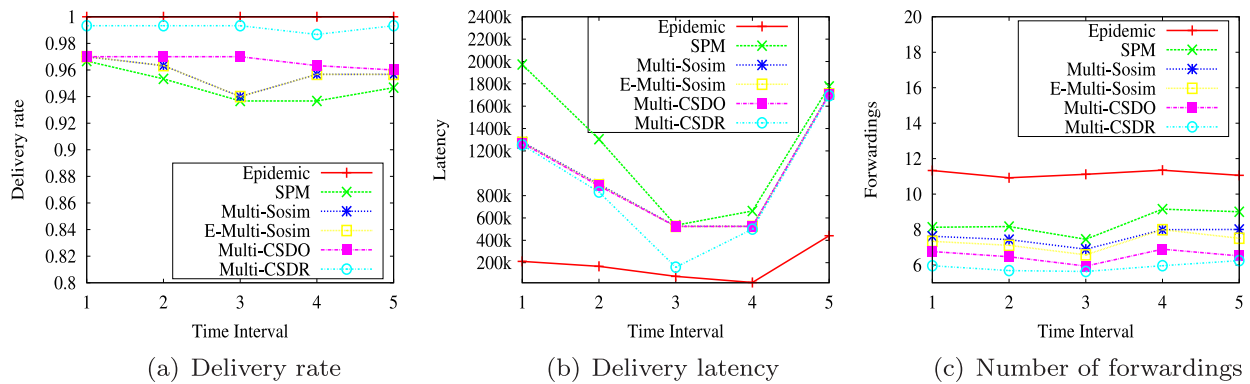
Fig. 10. Comparison of algorithms with 3 destinations using unical/socialblueconn trace.

(a) Delivery rate   (b) Delivery latency   (c) Number of forwardings
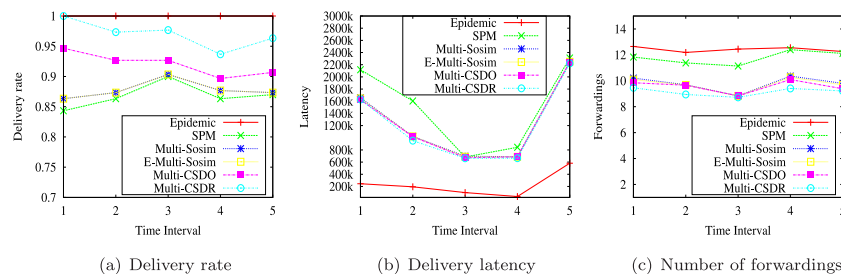
Fig. 11. Comparison of algorithms with 5 destinations using unical/socialblueconn trace.

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 two real traces of OMSNs showed that our new algorithms consistently outperform the existing ones in delivery rate, latency, and number of forwardings. Right now, the community detection in the proposed algorithms uses the social features in user profiles. Relative to the online social features such as friendship, we refer to them as offline social features. As observed by [31], the Facebook friendship (online social features) graph is always more similar to the Bluetooth contacts graph than the interests (offline social features) graph. In the future, we will explore the possibility of improving the multicast algorithms using online social features such as friendship and the combination of both online and offline social features and test them theoretically and experimentally.

Acknowledgment

References

[1] R.I. Ciobanu, R.C. Marin, C. Dobre, V. Cristea, C.X. Mavromoustakis, ONSIDE: Socially-aware and interest-based dissemination in opportunistic networks, in: IEEE Network Operations and Management Symposium (NOMS), 2014, pp. 1–6.

[2] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, Y. Sun, Geo-community-based broadcasting for data dissemination in mobile social networks, IEEE Trans. Parallel Distributed Syst. 24 (4) (2013) 734–743.

[3] F.D. Rango, A. Socievole, S. Marano, Exploiting online and offline activity-based metrics for opportunistic forwarding, Wireless Netw. 21 (2015) 1163–1179.

[4] A. Socievole, E. Yoneki, F.D. Rango, J. Crowcroft, ML-SOR: Message routing using multi-layer social networks in opportunistic communications, Comput. Netw. 81 (2015) 201–219.

[5] J. Wu, Y. Wang, Opportunistic Mobile Social Networks, Taylor & Francis, 2014.

[6] M. Xiao, J. Wu, L. Huang, Community-aware opportunistic routing in mobile social networks, IEEE Trans. Comput. 63 (7) (2014) 1682–1695.

[7] H. Zhou, J. Chen, J. Fan, Y. Du, S.K. Das, ConSub: incentive-based content subscribing in selfish opportunistic mobile networks, IEEE Jnl.Selected Areas Commun. 31 (9) (2013a) 669–679.

[8] H. Zhou, J. Chen, H.Y. Zhao, W. Gao, P. Cheng, On exploiting contact patterns for data forwarding in duty-cycle opportunistic mobile networks, IEEE Trans. Veh. Tech. 62 (9) (2013b) 4629–4642.

[9] B. Guo, D. Zhang, Z. Yu, X. Zhou, Z. Zhou, Enhancing spontaneous interaction in opportunistic mobile social networks, Commun. Mobile Comput. 1 (2012) 1–6.

[10] B. Jedari, F. Xia, A Survey on Routing and Data Dissemination in Opportunistic Mobile Social Networks, (http://arxiv.org/abs/1311.0347).

[11] U. Lee, S.Y. Oh, L. K.-W., M. Gerla, Relaycast: scalable multicast routing in delay tolerant networks, in: IEEE ICNP, 2008, pp. 218–227.

[12] M. Mongiovi, A.K. Singh, X. Yan, B. Zong, K. Psounis, Efficient multicasting for delay tolerant networks using graph indexing, IEEE INFOCOM, 2012.

[13] Y. Wang, J. Wu, A dynamic multicast tree based routing scheme without replication in delay tolerant networks, J. Parallel Distributed Comput. 72 (3) (2012) 424–436.

[14] Y. Xi, M. Chuah, An encounter-based multicast scheme for disruption tolerant networks, Comp. Comm. 32 (16) (2009) 1742–1756.

[15] W. Zhao, M. Ammar, E. Zegura, Muticasting in delay tolerant networks: semantic models and routing algorithms, in: ACM WDTN, 2005, pp. 268–275.

[16] X. Deng, L. Chang, J. Tao, J. Pan, J. Wang, Social profile-based multicast routing scheme for delay-tolerant networks, in: IEEE ICC, 2013, pp. 1857–1861.

[17] Y. Xu, X. Chen, Social-similarity-based multicast algorithm in impromptu mobile social networks, IEEE Globecom, 2014.

[18] E. Daly, M. Haahr, Social network analysis for routing in disconnected delay–tolerant MANETs, in: IEEE MobiHoc, 2007, pp. 32–40.

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

[20] P. Hui, J. Crowcroft, E. Yoneki, Bubble rap: social-based forwarding in delay tolerant networks, in: IEEE MobiHoc, 2008, pp. 241–250.

[21] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Technical Report, Dept. of Comp. Sci., Duke Univ., 2000.

[22] M. Chuah, P. Yang, Context-aware multicast routing scheme for disruption tolerant networks, J. Ad Hoc Ubiquitous Comput. 4 (5) (2009) 269–281.

[23] D. Rothfus, C. Dunning, X. Chen, Social-similarity-based routing algorithm in delay tolerant networks, in: IEEE ICC, 2013, pp. 1862–1866.

[24] J.W. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, MA, USA, 2012.

[25] Community structure, http://en.wikipedia.org/wiki/Community _structure.

[26] Complete-linkage clustering, (http://en.wikipedia.org/wiki/Complete_linkage_clustering).

[27] Hierarchical clustering, (http://docs.scipy.org/doc/scipy-0.17.0/reference/generat/scipy.cluster.hierarchy.linkage.html).

[28] The one the opportunistic network environment simulator, (https://akeranen.github.io/the-one/).

[29] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, A. Chaintreau, CRAWDAD trace cambridge/haggle/imote/infocom2006(v.2009-05-29), 2009, (http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006).

[30] A. Caputo, A. Socievole, F.D. Rango, CRAWDAD dataset unical/socialblueconn (v. 201508), 2015, (downloaded from http://crawdad.org/unical/socialblueconn/20150208, doi:10.15783/C7GK5R).

[31] A. Socievole, F.D. Rango, A. Caputo, Wireless Contacts, Facebook Friendships and Interests: Analysis of a Multi-layer Social Network in an Academic Environment, in: 2014 IFIP Wireless Days (WD), 2014, pp. 1–7.

Xiao Chen is an associate professor of Computer Science at Texas State University, San Marcos. She received her Ph.D. degree in Computer Engineering from Florida Atlantic University. Her research interests include delay-tolerant networks, sensor networks, and ad hoc wireless networks. She has served as an associate editor, program committee member, session chair, and reviewer for numerous international journals and conferences.

Charles Shang is an undergraduate student of Department of Computer Science at University of Illinois at Urbana-Champaign. His research interests include wireless mobile social networks, delay-tolerant networks, and ad hoc wireless networks.

Britney Wong is an undergraduate student of Department of Computer Science at Cornell University. Her research interests are wireless mobile social networks, delay-tolerant networks, and ad hoc wireless networks.

Wenzhong Li received the B.S. and Ph.D.degrees from Nanjing University, Nanjing, China, both in computer science. He is currently an Associate Professor of computer science with Nanjing University. His current research interests include wireless networks, pervasive computing, and social networks. He published over 30 papers at international conferences and journals. He was the winner of the Best PaperAward of ICC in 2009. He is a member of ACM.

Suho Oh is an assistant professor of Mathematics at Texas State University. He received his Ph.D. degree in Mathematics from Massachusetts Institute of Technology. His research interest is in Algebraic Combinatorics and Discrete Mathematics.