# Social-Similarity-based Multicast Framework in Opportunistic Mobile Social Networks


Xiao Chen[a,*], Yuan Xu[a], Suho Oh[b]

[a] *Department of Computer Science, Texas State University, San Marcos, TX 78666*
[b] *Department of Mathematics, Texas State University, San Marcos, TX 78666*


---


**Abstract**

With the proliferation of mobile devices, Opportunistic Mobile Social Networks (OMSNs) where the communication takes place on-the-fly by the opportunistic contacts among mobile users when they gather together at events have become increasingly popular. Multicast is an important routing service which supports the dissemination of messages to a group of users. Some existing multicast algorithms are designed by taking advantage of the internal social features of nodes in the network. This approach is motivated by the fact that nodes come in contact more frequently if they have more social features in common. These social features are obtained from nodes' profiles and thus static. Different from these multicast protocols that utilize static social features, in this paper, we adopt dynamic social features to more accurately capture node contact behavior and thereafter propose a novel Social-Similarity-based Multicast framework using the dynamic social features and a compare-split scheme to improve multicast efficiency in OMSNs. We instantiate the framework with two multicast algorithms named Multi-SoSim and E-Multi-SoSim that adopt the dynamic and enhanced dynamic social features, respectively. A detailed analysis of the proposed algorithms is given and simulations are conducted to evaluate our proposed algorithms by comparing them with their variations and the existing one using static social features.



*Corresponding author
*Email address:* `xc10@txstate.edu` (Xiao Chen)

*Keywords:* multicast, multicast tree, opportunistic mobile social networks, social features, social similarity

## 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, 2, 3, 4, 5]. Unlike popular online social networks such as Facebook and LinkedIn, the OMSNs we discuss here are a special kind of Delay Tolerant Networks (DTNs) [6] where the communication takes place on-the-fly by the opportunistic contacts among mobile users when they gather together at conferences, social events, rescue sites, campus activities, and so on. This type of communication relies on a lightweight mechanism via local wireless bandwidth such as Bluetooth or WiFi without a network infrastructure [2, 7, 8]. In OMSNs, node connections are usually short-term, time-dependent, and unstable as people come and go at events.

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.

Due to the uncertainty and time-dependent nature of OMSNs, there does not guarantee a path from a source to the destinations at any time, which poses special challenges to routing, either unicast or multicast. Nodes in OMSNs can only communicate in a store-carry-forward fashion: When two nodes move within each other's transmission range, they *meet* each other and can communicate directly, and when they are out of the range, 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.

Most existing multicast algorithms focus on DTNs [9, 10, 11, 12, 13] without considering social factors. Recently, a few algorithms propose to take advantage of the social features in user profiles to facilitate routing [2, 14, 15]. Among these algorithms, the one proposed by Deng et al. [14] addresses multicast. Specifically, the researchers found, through the study of the Infocom06 trace, that the social features in user profiles could effectively reflect nodes' contact behavior and developed a social profile-based multicast (SPM) scheme based on the two most important social features: *Affiliation* and *Language*. In their scheme, social features $F_i$ can refer to non-private user attributes such as *Nationality*, *City*, *Language*, *Affiliation*, and so on and these social features can take different values $f_i$. For example, a social feature can be *Language* and its value can be *English*. The intuition is that nodes having more common social features come to meet more often. Thus the nodes having more common social features with the destination are better forwarders to deliver the message to it. We believe, in the dynamic environment of OMSNs, the multicast algorithm can be further improved because the static social features may not always capture nodes' dynamic contact behavior. For example, a student who puts *New York* as his *state* in his profile may actually attend a conference in Texas. In that case, the static information in his profile can not reflect his behavior in Texas. The information that is helpful in making multicast decisions can only be gathered from the nodes' contact behavior at the conference. Therefore, in this paper, we extend static social features to *dynamic social features* to better reflect nodes' contact behavior and thereafter develop a new multicast algorithm specifically for OMSNs based on the dynamic social features.

In dynamic social features, we want to embed information that can reflect users' dynamic behavior to facilitate routing and that can be easy to obtain and inexpensive to maintain in OMSNs. Thus, we not only record if a node has the same social feature value with the destination, but also record the frequency this node has met other nodes that have the same social feature value during the time interval we observe. For example, we not only record if node $A$, same as the destination, is a *New Yorker* but also record that it has met *New Yorkers*

90% of the time during the observation interval. Unlike the static social features from user profiles, dynamic social features are time-related. So they change as user contact behavior changes over time. So we can have a more accurate way to choose the best forwarders in multicast. In this paper, we first apply the frequency-based dynamic social features and then the enhanced dynamic social features to multicast to improve its performance.

In multicast, a message holder is expected to forward a message to multiple destinations. To reduce the overhead and forwarding cost, the destinations should share the routing path as much as possible until the point that they have to be separated. Thus, the overall multicast process results in a tree structure. A compare-split scheme to determine the separation point is critical to the efficiency of a multicast. In our multicast, if a message holder $x$ meets another node $y$, the scheme of compare-split is based on the social similarity of each of the destinations with $x$ and $y$ using dynamic social features. That is, whichever, either $x$ or $y$, is more socially close to the destination will have a higher chance to deliver the message and thus should relay the message to that destination.

Based on the notions of dynamic social features and the scheme of compare-split, we propose a novel *social-similarity-based multicast* framework for OMSNs. Two algorithms instantiate this framework: the *social-similarity-based multicast* (Multi-Sosim) algorithm which utilizes dynamic social features to capture node contact behavior and a compare-split scheme to select the best relay node for each destination in each hop to improve multicast efficiency and the *enhanced social-similarity-based multicast* (E-Multi-Sosim) algorithm which upgrades the dynamic social features in Multi-Sosim to enhanced dynamic social features to further improve multicast efficiency. To evaluate the performance of our algorithms, we conduct an analysis and compare them with the existing algorithm that uses static social features and some variations of the proposed algorithms. Simulation results conclude that using dynamic social features can make better multicast routing decisions than using the static ones, letting destinations share the paths longer can reduce the cost, and separating destinations and allocating them to better forwarders can reduce latency.

The rest of the paper is organized as follows: Section 2 references the related works; Section 3 gives the definitions of dynamic social features and the calculation of social similarity; Section 4 presents our multicast algorithms; Section 5 gives the analysis of the algorithms; Section 6 shows the simulation results; and the conclusion is in Section 7.

## 2. Related Works

The multicast algorithm in Mobile Social Networks (MSNs) can be implemented using rudimentary approaches such as flooding [16], but it has inevitable high forwarding cost. Most of the existing multicast algorithms are designed for DTNs where social factors are not considered. Zhao et al. [13] introduce some new semantic models for multicast and conclude that the group-based strategy is suitable for multicast in DTNs. Lee et al. [9] 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. [12] present an encounter-based multicast routing, and Chuah et al. [17] develop a context-aware adaptive multicast routing scheme. Mongiovi et al. [10] use graph indexing to minimize the remote communication cost of multicast. And Wang et al. [11] 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 multicast papers that involve social factors. Gao et al. [18] propose a community-based multicast routing scheme by exploiting node centrality and social community structures. This approach is based on the fact that “social relations among mobile users are more likely to be long-term characteristics and less volatile than node mobility” [18] in MSNs. Hu et al. [19, 20] put forward multicast algorithms to disseminate data in MSNs. In [19], the content owners multicast to their social contacts which are defined by the geographic social strength between nodes and in [20], node centrality in the social contact graph extracted from node contact trace is adopted to select the initial receiver set [20]. Deng et al. [14] propose a social-profile-based multicast (SPM) algorithm that uses social features in user profiles to guide the multicast routing in MSNs. This approach has the advantage of not having to record node contact history, but the static social features may not catch people's dynamic contact behavior in the OMSNs. So the multicast algorithm for OMSNs can be further improved by catching the dynamic features of the network.

## 3. Preliminary

In this section, we first introduce static social features used in the existing papers [14], then define dynamic social features and its enhanced version, and then give the formula to calculate nodes' social similarities which will be used in the compare-split scheme in our multicast algorithms.

### *3.1. Definition of Static Social Features*

Suppose we consider $m$ social features $\langle F_1, F_2, \cdots, F_m \rangle$ in an OMSN. Note that the choice of which social features to include in a network depends on the situation and the nodes can agree on the selection through message exchange or manually when the network was first set up. A node $x$'s static social features is a a vector in the form of $\langle x_1, x_2, \cdots, x_m \rangle$, where $x_i$ is the social feature value for $F_i$ obtained from the user's profile.

### *3.2. Definitions of Dynamic Social Features*

In dynamic social features, we define $x_i$ as follows based on nodes' encounter history to capture nodes' contact behavior.

#### *3.2.1. Dynamic Social Features*

One definition of $x_i$ is the frequency of node $x$ meeting nodes with the same $f_i$ out of all of the nodes it has met in the history we observe. That is,

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

In definition (1), $M_i$ is the number of times that $x$ has met nodes with the same $f_i$ in the history we observe and $M_{total}$ is all of the nodes that $x$ has met in that interval. For example, if $f_i$ refers to *Student* and if $x$ has met 20 *Students* out of a total of 100 people, then $x_i = 20/100 = 0.2$.

Nevertheless, one problem with the frequency definition of $x_i$ is that if node $x$ has met one *Student* out of two people it has met in total in the history we observe and node $y$ has met five *Students* out of ten people it has met in total, using definition (1), both of their frequencies are 0.5 in meeting *Students*. So which one is more likely to meet *Students* in the future? From the intuition, node $y$ should be given a higher priority because it is more active in meeting people. There are many formulas we can design to favor $y$. In the following, we present one formula, which will be proved in the later Analysis section, that can break the tie and favor the more active node.

#### 3.2.2. Enhanced Dynamic Social Features

In this enhanced definition of dynamic social features, $x_i$ is calculated as:

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

In definition (2), $p_i = M_i/M_{total}$. This definition predicts $x_i$ by looking at the next meeting probability of node $x$ with another node having the same $f_i$. In the next time, the total meeting times will be $M_{total}+1$. The first part $(\frac{M_i+1}{M_{total}+1})^{p_i}$ means that there will be $p_i$ probability that $x$ will have a "good" meeting with another node having the same social feature value $f_i$ next time. In this case, $M_i$ will also be incremented by 1. The second part $(\frac{M_i}{M_{total}+1})^{1-p_i}$ means that there will be $1-p_i$ probability for $x$ not to meet a node with the same $f_i$ next time. In that case, $M_i$ will remain the same. The definition of $x_i$ then takes the geometric mean of the two parts.

With definition (2), we can break the tie in the example above. For node $x$, $M_i = 1, M_{total} = 2, p_i = 0.5$; and for node $y$, $M_i = 5, M_{total} = 10, p_i = 0.5$. Using definition (2), $x_i = (1+1)^{0.5} * \frac{1^{(1-0.5)}}{2+1} = 0.4714$ and $y_i = (5+1)^{0.5} * \frac{5^{(1-0.5)}}{10+1} = 0.4979$. These two results are close, reflecting that the two nodes had the same frequency using definition (1), yet they tell us that $y$ is better because it is more active meeting nodes.

Dynamic social features, as shown in the definitions, not only record if a node has certain social features, but also predict the probability of this node meeting other nodes with the same social features. Unlike the static social features, dynamic social features change as user activities change over time. So they can better reflect users' dynamic contact behavior in OMSNs.

### *3.3. Calculation of Social Similarity*

With nodes' dynamic social features defined, we can use similarity metrics such as Tanimoto [21], Cosine [22], Euclidean [23], and Weighted Euclidean [24] derived from data mining [25] to calculate the social similarity $S(x, y)$ of two nodes $x$ and $y$. We finally decide to use the Euclidean similarity metric because it does not require the calculation of additional weighting values and performs slightly better than Tanimoto and Cosine in terms of latency when these metrics are compared in our simulations [24].

**Euclidean Similarity Metric**

After normalizing the original definition of the Euclidean similarity [23] in data mining to the range of $[0, 1]$ and subtracting it from 1, it is now defined as

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

Using the Euclidean similarity metric, if two nodes $x$ and $y$ have the same dynamic social features, e.g., $x_i = y_i$, then $S(x, y) = 1$. In other words, they have 0 *social similarity gap*. So the social similarity gap of two nodes is defined as $1 - S(x, y)$.

Here is how the metric 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 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 $x$ has a social similarity gap of $1 - 0.62 = 0.38$ to $d$ and $y$ has a social similarity gap of 0.54 to $d$. Thus $x$ 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 a similar way.

## 4. Multicast Routing Protocols

In this section, we propose a social-similarity-based multicast framework that selects the best forwarding nodes depending on the social similarity of nodes using dynamic social features and a compare-split scheme. Here we assume that there is one multicast source. If there are multiple multicast sources, then the framework can be used by each individual source to multicast messages. In the framework, when the social similarities of the nodes are calculated using dynamic social feature definition (1) and definition (2), the resulting multicast algorithms are called Multi-Sosim and E-Multi-Sosim, respectively.

### *4.1. Social-Similarity-based Multicast Framework*

Our multicast framework is shown in Fig. 1. In the beginning, suppose a source node $s$ has a message to send to a set of destinations which we refer to as its destination set $D_s = \{d_1, d_2, \cdots, d_k\}$. The destination sets of all of the other nodes are initially empty. The message holder is denoted as $x$. Initially, $x$ is the source node $s$.

If the message holder $x$ meets a node $y$, we first check if $y$ is one of the destinations. If it is, $x$ will deliver the message to $y$ and remove it from its destination set. Next, we combine the destination sets of $x$ and $y$ into $D_{xy}$ and make the destination sets $D_x$ and $D_y$ empty. Then we use a compare-split scheme to split the destinations in $D_{xy}$ to $D_x$ and $D_y$ by comparing the social similarity of each of the destinations $d_i$ with $x$ and $y$. The social similarity $S(x, y)$ of $x$ and $y$ is calculated either by dynamic social feature definition (1) or (2). If $y$ is more socially similar to $d_i$, then $d_i$ will be placed into $D_y$, meaning

**Multicast Framework: social-similarity-based multicast framework**

**Require:** The source node $s$ and its destination set $D_s = \{d_1, d_2, \cdots, d_k\}$; the destination sets of all of the nodes except $s$ are empty; the initial message holder $x$ is $s$

/* On contact between a message holder $x$ and node $y$: */
**if** $y \in D_x$ **then**
  $x$ forwards the message to destination $y$ and removes $y$ from $D_x$
**end if**
/* Combine the destination sets of $x$ and $y$ */
Let $D_{xy} = D_x \cup D_y$ and $D_x = D_y = \emptyset$
/* Compare node social similarities and split the destinations in $D_{xy}$ to $D_x$ and $D_y$ */
**for** each of the destinations $d_i \in D_{xy}$ **do**
  **if** $S(x, d_i) < S(y, d_i)$ **then**
    add $d_i$ to $D_y$, and $x$ forwards the message to $y$ if $y$ does not have it
  **else**
    add $d_i$ to $D_x$
  **end if**
**end for**

Figure 1: Our multicast framework

that $y$ will be the next forwarder for the message destined for $d_i$; otherwise, $d_i$ will be put into $D_x$. After this, nodes $x$ and $y$ will become new message holders and the process will repeat until all of the destinations have received the message.

Starting from the source node $s$ and through the splits in the middle, the multicast process naturally forms a tree. It follows the cost reduction intuition that the destinations should share the paths on the tree as long as possible until a better node appears to carry over some of the destinations. This idea can be clearly presented in the example shown in Fig. 2. In the figure, the label in a solid circle represents an intermediate relay and the label in a dashed

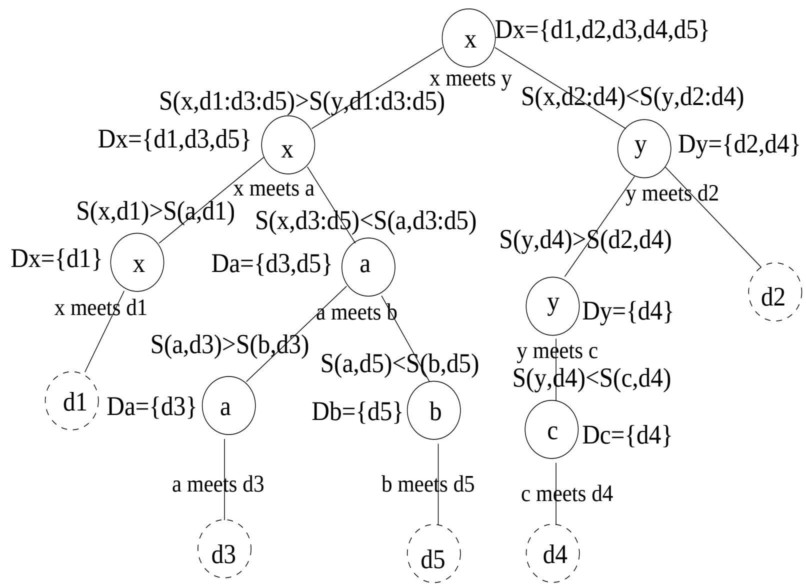



Figure 2: A tree showing the multicast process. The notation "$S(x, d_i : d_j : d_k) > S(y, d_i : d_j : d_k)$" means "$S(x, d_i) > S(y, d_i)$ and $S(x, d_j) > S(y, d_j)$ and $S(x, d_k) > S(y, d_k)$ ".

circle represents a destination. Initially, the source node or message holder $x$ has a message to deliver to the destination set $D_x = \{d_1, d_2, d_3, d_4, d_5\}$. When $x$ meets a node $y$, if destinations $d_1, d_3, d_5$ are more socially similar to $x$ than $y$, then they will be allocated to $D_x$, and $d_2, d_4$ will be allocated to $D_y$ if they are more socially similar to $y$. The notation "$S(x, d_i : d_j : d_k) > S(y, d_i : d_j : d_k)$" is a shortened form of "$S(x, d_i) > S(y, d_i)$ and $S(x, d_j) > S(y, d_j)$ and $S(x, d_k) > S(y, d_k)$". Later, when $x$ meets node $a$ and $a$ meets node $b$, they will make decisions following the same rule. The multicast tree continues expanding until all of the destinations are reached.

## 5. Analysis

In this section, we analyze the properties of our algorithms.

### *5.1. Property of Dynamic Social Feature Definition (2)*

**Theorem 1.** *Suppose node $x$ has met $M_{xi}$ nodes with a certain social feature out of $M_{xtotal}$ nodes it has met so far and node $y$ has met $M_{yi}$ nodes with the* *same social feature out of $M_{ytotal}$ nodes it has met so far. 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 $x_i \leq y_i$. That is, definition (2) breaks the tie of the same frequency by favoring the more active node.*

*Proof.* To prove $x_i \leq y_i$, it is equivalent to proving 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. Therefore the left side is less or equal to the right side. This proves the theorem. □

### 5.2. The Number of Forwardings

**Theorem 2.** *In our routing framework, if there is only one destination $d$ in the destination set $D$, the expected number of forwardings to reach the destination from source $s$ 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 be 1 and define the gap within which to reach $d$ in one hop to be $\beta$ as shown in Fig. 3(a). In other words, if the message holder is within gap $\beta$ to

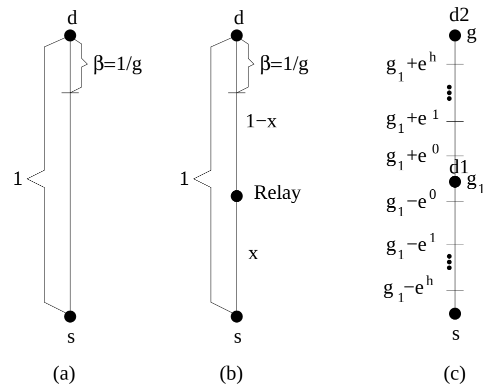



Figure 3: (a) One destination $d$, whose gap to source $s$ is 1. The range to reach $d$ in one hop is $\beta = 1/g$. (b) Reaching $d$ in 2 hops via the relay node. The gap from $s$ to the relay is $x$ and the gap from the relay to $d$ is $1 - x$. (c) Two destinations $d_1$ and $d_2$, whose gaps to $s$ 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.

$d$, that node can deliver the message to $d$ in one hop. Since the gap length is $\beta$ and the gap from $s$ to $d$ is 1, the probability of a node falling in such a gap is $\beta$. So $\beta$ is also the probability to reach $d$ in one hop. Relative to the original gap $g$ between $s$ and $d$, gap $\beta$ is equal to $\frac{1}{g}$.

Now let us calculate the probability to reach $d$ in $h$ hops from $s$. If $h = 1$, that means $d$ can be reached from $s$ in one hop. That probability is $\beta$ according to the above explanation. If $h = 2$, that means $d$ can be reached from $s$ in two hops. Then there should be a relay lying between $s$ and $d$ as shown in Fig. 3(b). Assume the gap from $s$ to the relay is $x$ and the gap from the relay to $d$ is $1-x$. Now the probability to reach $d$ from $s$ in two hops becomes $\frac{\beta}{1-x}$. Since $x$ is in the range of $[0, 1-\beta]$, the overall probability to reach $d$ in two hops should be $\int_0^{1-\beta} \frac{\beta}{1-x} dx$. The same reasoning can be extended to calculate the probability to reach $d$ in $h$ hops.

Therefore, the probability to reach $d$ in

$$1 \text{ hop from s is: } \beta,$$
$$2 \text{ hops from s is: } \int_0^{1-\beta} \frac{\beta}{1-x} dx = \beta \ln \frac{1}{\beta},$$
$$3 \text{ hops is: } \int_0^{1-\beta} \int_{x_1}^{1-\beta} \frac{\beta}{(1-x_1)(1-x_2)} dx_2 dx_1 = \frac{\beta}{2!} (\ln \frac{1}{\beta})^2,$$
$$\cdots,$$
$$h \text{ hops is: } \int_0^{1-\beta} \int_{x_1}^{1-\beta} \cdots \int_{x_{h-1}}^{1-\beta} \frac{\beta}{(1-x_1)(1-x_2)\cdots(1-x_{h-1})} dx_{h-1} \cdots d_{x_1}$$
$$= \frac{\beta}{h!} (\ln \frac{1}{\beta})^h, \text{and so on.}$$

These probabilities form a distribution as their summation $\sum_{h=0}^{\infty} \frac{\beta}{h!} (\ln \frac{1}{\beta})^h$ 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 $1 + \ln \frac{1}{\beta} \cdot \beta \cdot e^{\ln \frac{1}{\beta}} = 1 + \ln \frac{1}{\beta} = \ln g + 1$. $\square$

**Theorem 3.** *The expected number of forwardings in our routing framework 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 routing framework, 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 node that has a smaller social similarity gap to that destination. Let us first look at the 2-destination case as shown in Fig. 3(c). 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 let us 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 the 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). $g_1 \le \frac{g}{2}$ and (2). $g_1 > \frac{g}{2}$.

In case (1), if the two destinations split at the $h+1$ ($h \ge 0$) hop, the expected number of extra forwardings to reach $d_1$ is

$$1 \cdot \tfrac{2e^0}{g-g_1+e^0} + 2 \cdot (\tfrac{2e^1}{g-g_1+e^1} - \tfrac{2e^0}{g-g_1+e^0}) + 3 \cdot (\tfrac{2e^2}{g-g_1+e^2} - \tfrac{2e^1}{g-g_1+e^1})$$
$$+ \cdots + \lceil \ln g_1 \rceil (1 - \tfrac{2e^{\lfloor \ln g_1 \rfloor -1}}{g-g_1+e^{\lfloor \ln g_1 \rfloor -1}}) = \lceil \ln g_1 \rceil - \textstyle\sum_{h=0}^{\lfloor \ln g_1 \rfloor -1} \tfrac{2e^h}{g-g_1+e^h}.$$

From $g_1 \le \frac{g}{2}$ and $e^{\lfloor \ln g_1 \rfloor} \le g_1$, we have

$$\textstyle\sum_{h=0}^{\lfloor \ln g_1 \rfloor -1} \tfrac{2e^h}{2(g-g_1)} \le \sum_{h=0}^{\lfloor \ln g_1 \rfloor -1} \tfrac{2e^h}{g-g_1+e^h} \le \sum_{h=0}^{\lfloor \ln g_1 \rfloor -1} \tfrac{2e^h}{g-g_1}.$$

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

Again from $g_1 \le \frac{g}{2}$,

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

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 $h+1$ ($h \ge 0$) 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 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 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 our routing framework 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 our routing framework is $k$. □

## 6. Simulations

In this section, we evaluate the performance of our multicast algorithms by comparing them with their variations and the existing ones using a custom simulator written in Java. The simulations were conducted using a real conference trace [26] representing an OMSN created at INFOCOM 2006. The trace dataset consists of two parts: *contacts* between the iMote devices that were carried by conference participants and the self-reported *social features* of the participants, which were collected using a questionnaire form. The six social features considered were *Affiliation*, *City*, *Nationality*, *Language*, *Country*, and *Position*.

### *6.1. Algorithms Compared*

We compared the following related multicast protocols.

1. *The Flooding Algorithm* (Flooding) [16]: The message is spread epidemically throughout the network until it reaches all of the destinations.
2. *The Social-Profile-based Multicast Routing Algorithm* (SPM) [14]: The multicast algorithm based on static social features in user profiles.
3. *The Multi-Sosim Algorithm* (Multi-Sosim): Our multicast algorithm based on dynamic social feature definition (1) and using the Euclidean social similarity metric.

4. *The E-Multi-Sosim Algorithm* (E-Multi-Sosim): Our multicast algorithm based on dynamic social feature definition (2) and using the Euclidean social similarity metric.
5. *Variation* 1 *of the Multi-Sosim Algorithm* (Multi-FwdNew): This algorithm is similar to Multi-Sosim but a message holder only forwards the message to a newly met node whose destination set is empty.
6. *Variation* 2 *of the Multi-Sosim Algorithm* (Multi-Unicast): The message to multiple destinations is delivered by multiple independent unicasts (from the source to each of these destinations), where each unicast is conducted using dynamic social features.

### *6.2. Evaluation Metrics*

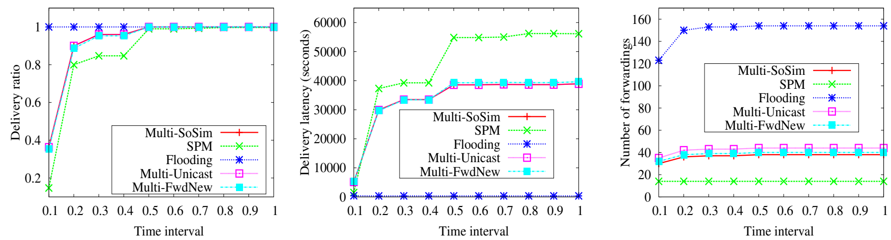
We used three important metrics to evaluate the performance of the multicast algorithms. Since a multicast involves multiple destinations, we define a *successful multicast* as the one that successfully delivers the message to all of the destinations. The three metrics are: (1) *Delivery ratio*: The ratio of the number of successful multicasts to the number of total multicasts generated. (2) *Delivery latency*: The time between when the source starts to deliver the message to when all of the destinations have received the message. (3) *Number of forwardings*: The number of forwardings needed to deliver the message to all of the destinations.

### *6.3. Simulation Setup*

In our simulations, we divided the whole trace time into 10 intervals. Thus, 1 time interval is 0.1 of the total time length and 10 time intervals make up the length of the whole trace. For each of algorithms compared, we tried the sizes of the destination sets to be 5 and 10. In each experiment, we randomly generated a source and its destination set. We ran each algorithm 300 times and averaged the results of the evaluation metrics.

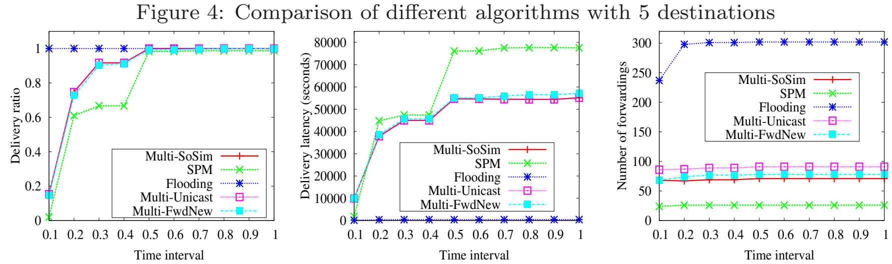
### *6.4. Simulation Results*

The simulation results with 5 and 10 destinations are shown in Figs. 4 and 5, respectively. For the flooding algorithm, as expected, it achieves the highest



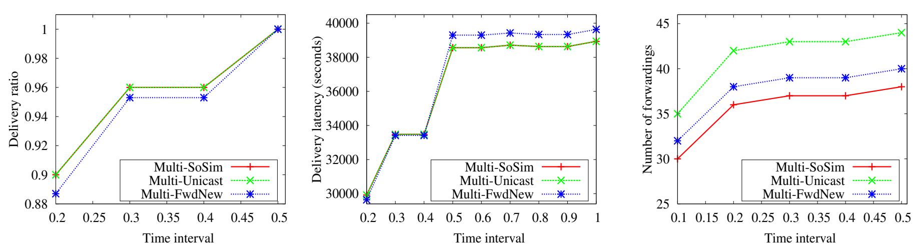

(a) Delivery ratio (b) Delivery latency (c) # of forwardings

Figure 4: Comparison of different algorithms with 5 destinations



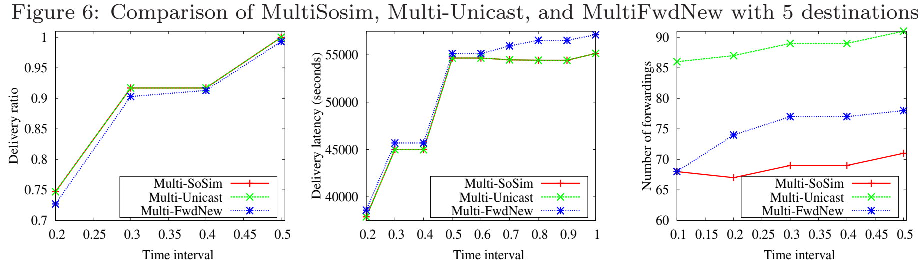

(a) Delivery ratio (b) Delivery latency (c) # of forwardings

Figure 5: Comparison of different algorithms with 10 destinations



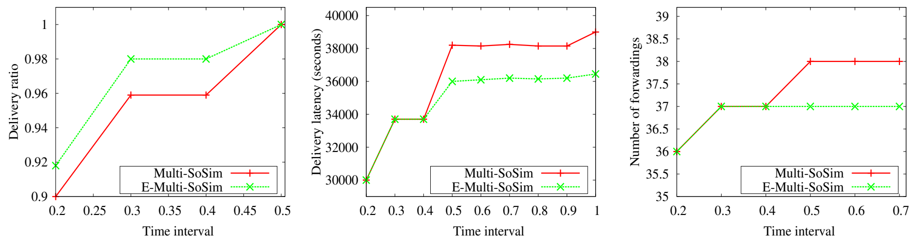

(a) Delivery ratio (b) Delivery latency (c) # of forwardings

Figure 6: Comparison of MultiSosim, Multi-Unicast, and MultiFwdNew with 5 destinations



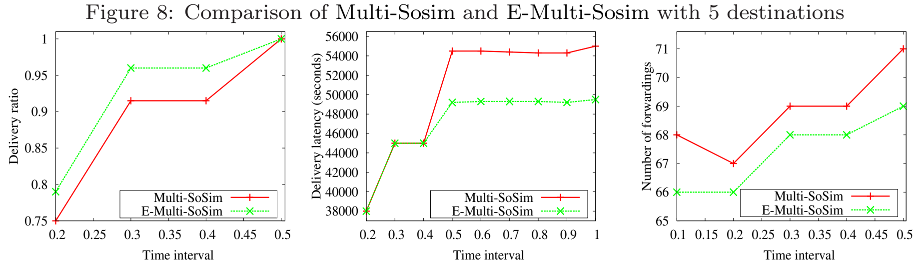

(a) Delivery ratio (b) Delivery latency (c) # of forwardings

Figure 7: Comparison of MultiSosim, Multi-Unicast, and MultiFwdNew with 10 destinations




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

Figure 8: Comparison of Multi-Sosim and E-Multi-Sosim with 5 destinations




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

Figure 9: Comparison of Multi-Sosim and E-Multi-Sosim with 10 destinations

delivery ratio and lowest delivery latency (almost close to 0 compared with others in the figures) at the cost of sending a copy to any newly met node. Thus it has the highest number of forwardings. The Multi-Sosim algorithm outperforms SPM in having a higher delivery ratio and lower latency with a little increase in the number of forwardings. This is because the dynamic social features in Multi-Sosim can more accurately capture node encounter behavior than the static social features in SPM so that multicast efficiency can be improved. The little increase in the forwardings indicates that Multi-Sosim is more actively delivering the message to the destinations.

Figs. 6 and 7 show the zoom-in simulation results of Multi-Sosim, Multi-Unicast, and Multi-FwdNew algorithms with 5 and 10 destinations. There is not much difference in delivery ratio and latency between Multi-Sosim and Multi-Unicast in this simulation as their curves are overlapped in the figures. But Multi-Sosim decreases the number of forwardings in Multi-Unicast by 16.7% and 29.9% with 5 and 10 destinations, respectively. This is because letting the destinations share the path in Multi-Sosim can reduce the number of forwarding nodes, especially when the number of destinations is increased. Multi-Sosim outperforms Multi-FwdNew in delivery ratio, latency, and the number of forwardings. With 5 destinations, the Multi-Sosim algorithm increases the delivery ratio by 1.5%, decreases latency by 2.0%, and decreases the number of forwardings by 6.7% comparing with Multi-FwdNew. With 10 destinations, the Multi-Sosim algorithm increases the delivery ratio by 2.8%, decreases latency by 3.9%, and decreases the number of forwardings by 11.6%. This is because Multi-Sosim selects a better forwarder for each of the destinations whenever a message holder meets another node while Multi-FwdNew does that only when a newly met node is encountered.

Figs. 8 and 9 present the comparison of Multi-Sosim and E-Multi-Sosim algorithms with 5 and 10 destinations. With 5 destinations, the E-Multi-Sosim algorithm increases the delivery ratio by 2.1%, decreases latency by 6.4%, and decreases the number of forwardings by 2.7% comparing with Multi-Sosim. With 10 destinations, the E-Multi-Sosim algorithm increases the delivery ratio by 4.3%, decreases latency by 2.9%, and decreases the number of forwardings by 10.6%. This is because the enhanced dynamic social features in E-Multi-Sosim can more accurately capture nodes' dynamic contact behavior to improve multicast efficiency.

## 7. Conclusion

In this paper, we have proposed a novel social-similarity-based multicast framework for OMSNs where node connections are established opportunistically. We have instantiated this framework with two algorithms Multi-Sosim and E-Multi-Sosim based on a compare-split scheme to select the best relay node for each of the destinations in each hop to improve multicast efficiency and dynamic and enhanced dynamic social features to capture nodes' contact behavior. We have conducted a theoretical analysis of our proposed algorithms and evaluated their performance by comparing them with other related algorithms through simulations using a real trace representing an OMSN. The simulation results have verified the advantages of the dynamic social features over the static ones and the appropriateness of the compare-split scheme adopted in our multicast algorithms. In our future work, we plan to test our algorithms using more traces in OMSNs as they become available.


### Acknowledgements

This research was supported in part by NSF CNS grant 1305302 and NSF ACI grant 1440637.


### References


[1] 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. on Parallel and Distributed Systems 24 (4) (2013) 734–743.

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

[3] M. Xiao, J. Wu, L. Huang, Community-Aware Opportunistic Routing in Mobile Social Networks, IEEE Trans. on Computers 63 (7) (2014) 1682–1695.

[4] H. Zhou, J. Chen, J. Fan, Y. Du, S. K. Das, ConSub: Incentive-Based Content Subscribing in Selfish Opportunistic Mobile Networks, IEEE Jnl. on Selected Areas in Communications 31 (9) (2013) 669–679.

[5] 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. on Veh. Tech. 62 (9) (2013) 4629–4642.

[6] DTN Research Group, `http://w5,www.dtnrg.org/`.

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

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

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

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

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

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

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

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

[15] A. Mei, G. Morabito, P. Santi, J. Stefa, Social-aware stateless forwarding in pocket switched networks, in: Proc. of IEEE INFOCOM, 2011, pp. 251–255.

[16] A. Vahdat, D. Becker, Epidemic routing for partially connected ad hoc networks, Tech. rep., CS-200006, Duke University (2000).

[17] M. Chuah, P. Yang, Context-aware multicast routing scheme for disruption tolerant networks, Journal of Ad Hoc and Ubiquitous Computing 4 (5) (2009) 269–281.

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

[19] J. Hu, L. L. Yang, L. Hanzo, Distributed Cooperative Social Multicast Aided Content Dissemination in Random Mobile Networks, IEEE Transactions on Vehicular Technology 64 (7) (2014) 3075–2229.

[20] J. Hu, L. L. Yang, H. V. Poor, L. Hanzo, Bridging the Social and Wireless Networking Divide: Information Dissemination in Integrated Cellular and Opportunistic Networks, IEEE Access 3 (2015) 1809–1848.

[21] Tanimoto Coefficient, `https://docs.tibco.com/pub/spotfire/7.0.1/doc/html//hc/hc_tanimoto_coefficient.htm`.

[22] Cosine Correlation, https://docs.tibco.com/pub/spotfire/7.0.1/doc/html//hc/hc_cosine_correlation.htm.

[23] Euclidean Distance, https://docs.tibco.com/pub/spotfire/7.0.1/doc/html//hc/hc_euclidean_distance.htm.

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

[25] J. W. Han, M. Kamber, J. Pei, Data Mining: concepts and techniques, Morgan Kaufmann, MA, USA, 2012.

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