# Social-Similarity-based Multicast Algorithm in Impromptu Mobile Social Networks

Yuan Xu, Xiao Chen
Department of Computer Science, Texas State University, San Marcos, TX, USA
Email: y_x3@txstate.edu, xc10@txstate.edu

***Abstract*—Mobile social networks (MSNs) where people contact each other through mobile devices have become increasingly popular. In this paper, we study a special kind of MSNs formed impromptu when people gather together at conferences, social events, etc. Multicast is an important routing service which supports the dissemination of messages to a group of users. Most of the existing related multicast algorithms are designed for general Delay Tolerant Networks (DTNs) where social factors are neglected. Recently, a social-profile-based multicast (SPM) protocol that utilizes the static social features in user profiles has been proposed. We believe that in a dynamic environment such as the IMSN, static social features may not reflect people's dynamic behavior. Therefore, in this paper, we propose a novel Social-Similarity-based Multicast Algorithm (Multi-Sosim) using nodes' dynamic social features and a compare-split scheme to improve multicast efficiency in IMSNs. Simulation results using a real trace show that our algorithm outperforms its variations and the existing one using static social features.**

***Index Terms*—mobile social networks, multicast, multicast tree, social features, social similarity**

## I. Introduction

Mobile social networks (MSNs) where people contact each other through mobile devices such as smart phones, tablets, and so on have become a hot research topic these days. In this paper, we study a special kind of MSNs formed impromptu when people gather together at conferences, social events, rescue sites, campus activities, etc. We refer to it as *Impromptu Mobile Social Network* (IMSN). In IMSNs, 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 IMSNs. 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 uncertain and time-dependent nature of IMSNs, there may not guarantee a path from a source to the destinations at any time. Therefore, IMSNs are special cases of Delay Tolerant Networks (DTNs) that involve social factors. Routing, either unicast or multicast, poses special challenges. Nodes in IMSNs can only communicate through a store-carry-forward fashion. When two nodes move within each other's transmission range, they communicate directly during that time period. When they move out of their ranges, their contact is lost. The message to be delivered needs to be stored in the local buffer until a contact occurs in the next hop.

Most of the existing multicast algorithms focus on general DTNs [5], [6], [11], [13], [14] without social factors. There are few multicast algorithms specifically designed for IMSNs. The closest we can find is Deng et al.'s multicast algorithm proposed for MSNs in [2]. The researchers found, through the study of the Infocom06 trace, that the static social features in user profiles could effectively reflect node contact behavior and developed a social profile-based multicast (SPM) scheme based on the two most important social features: *Affiliation* and *Language* they extracted from the trace. In their scheme, social features $F_i$ can refer to $Nationality, City, Language, Affiliation$, and so on and these social features can take different values $f_i$. For example, a social feature $F_i$ can be $Language$ and its value $f_i$ can be $English$. The intuition is that nodes sharing more common social features tend to meet more often. Thus the nodes that have more common social features with the destination are better forwarders to deliver the message to it. We believe, in a dynamic environment such as the IMSN, the multicast algorithm can be further improved because the static social features may not always capture node 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 user profile can not reflect his behavior in Texas. The information that is helpful in making multicast decisions can only be gathered from the node contact behavior at the conference. Therefore, in this paper, we extend static social features to *dynamic social features* to better reflect node contact behavior and develop a new multicast algorithm specifically for IMSNs based on the dynamic social features.

In multicast, a message holder is expected to forward a message to multiple destinations. To reduce the overhead and forwarding cost, the destinations will share the routing path until the point that they have to be separated, which usually results in a tree structure. In our multicast, we use a compare-split scheme to determine the separation point. When a message holder meets another node, we compare the social similarity of each of the destinations with the message holder and with the meeting node based on the dynamic social features, and then split the destinations according to the comparison results. That is, whichever, either the message holder or the meeting node, is more socially close to the destination will be responsible for relaying the message to that destination due to its higher delivery probability.

Based on the dynamic social features and the compare-split scheme, we propose a novel *social-similarity-based multicast* (Multi-Sosim) routing algorithm for IMSNs. In addition, we discuss its two variations: (1) *Multi-FwdNew* which is similar to Multi-Sosim but the message holder only considers forwarding the message to a newly met node so that destinations can share the paths longer. And (2) *Multi-Unicast* where multicast is implemented by multiple unicasts with each unicast conducted using dynamic social features. To evaluate the performance of the Multi-Sosim algorithm, we compare it with Multi-FwdNew and Multi-Unicast, the existing SPM algorithm, and the Epidemic algorithm as a benchmark. Simulation results show that Multi-Sosim outperforms SPM by having a higher delivery ratio, a lower latency, and a little increase in the number of forwardings, which confirms that using dynamic social features can make better multicast routing decisions than using static social features in IMSNs. The better performance of Multi-Sosim compared to Multi-FwdNew and Multi-Unicast concludes that letting the destinations share the paths longer can reduce the cost and separating the destinations to better forwarders can reduce latency.

The rest of the paper is organized as follows: Section II references the related works; Section III presents our multicast algorithm; Section IV shows the simulation results; and the conclusion is in Section V.

## II. Related works

The multicast algorithm in MSNs can be implemented using rudimentary approaches such as Epidemic routing [10], but it has inevitable high forwarding cost. Most of the existing related multicast algorithms are designed for DTNs where social factors are not considered. Zhao et al. [14] introduce some new semantic models for multicast and conclude that the group-based strategy is suitable for multicast in DTNs. Lee et al. [5] 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. [13] present an encounter-based multicast routing, and Chuah et al. [1] develop a context-aware adaptive multicast routing scheme. Mongiovi et al. [6] use graph indexing to minimize the remote communication cost of multicast. 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 papers that study multicast in MSNs. Gao et al. [3] 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 and less volatile than node mobility in MSNs. But it may not be suitable for IMSNs where social relations are short-term and unstable. Recently, Deng et al. [2] propose a social-profile-based multicast (SPM) algorithm that uses social features in user profiles to guide the multicast routing in MSNs. More specifically, the algorithm selects relay nodes with a small average affiliation distance or high common language ratio to the destinations. 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 IMSNs. So the multicast algorithm for IMSNs needs to be rethought about.

## III. Multicast Routing Protocol

In this section, we propose a multicast routing algorithm called Multi-Sosim for IMSNs that selects the best forwarding nodes based on social similarity of nodes using dynamic social features and a compare-split scheme.

### A. Social-Similarity-based Multicast Algorithm (Multi-Sosim)

Our multicast routing algorithm Multi-Sosim is shown in Fig. 1. In the beginning, a source node $s$ has a message to be delivered to a set of destinations $D_s = \{d_1, d_2, \cdots, d_n\}$. We refer to $D_s$ as the destination set of $s$. We initialize the destination sets of all of the other nodes empty. We start the routing process in a while loop. As long as not all of the $n$ destinations have received the message, we repeat the following steps to choose the next best forwarding node for each of these destinations.

When a 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 $y$. Next, we will combine the destination sets of $x$ and $y$ into $D_{xy}$ and make the destination sets $D_x$ and $D_y$ to be empty. Then we use a compare-split scheme to split the destinations in $D_{xy}$ and put them into $D_x$ and $D_y$ by comparing the social similarity of each destination $d_i$ with $x$ and $y$. The social similarity $S(x, y)$ of two nodes $x$ and $y$ is calculated based on the dynamic social features of nodes, which will be explained in detail in Section III-B. If $y$ is more socially similar to $d_i$, then $d_i$ should be placed into $D_y$, meaning $y$ will be the next forwarder for the message destined for $d_i$; otherwise, $d_i$ should be placed into $D_x$ and $x$ will be the next forwarder for the message to $d_i$. After $x$ and $y$ regain their destination sets, they become new message holders and will repeat the routing process 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, then the destinations split. This idea can be clearly presented in the example shown in Fig. 2. In the figure, the label in a solid circle represents a node and the label in a dashed circle represents a destination. Initially, the source node $x$ has a message 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

**Algorithm Multi-Sosim: social-similarity-based multicast routing algorithm**

**Require:** The source node $s$ and its destination set $D_s = \{d_1, d_2, \cdots, d_n\}$

Initialize the destination sets of all of the nodes except $s$ to be empty
**while** not all of the destinations receive the message **do**
  On contact between a message holder $x$ and node $y$:
  **if** $y \in D_x$ **then**
    /* Found the destination $y$ */
    $x$ forwards a copy of the message to $y$ 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 destination $d_i \in D_{xy}$ **do**
    /* Calculate the social similarity $S(x, d_i)$ and $S(y, d_i)$, respectively */
    **if** $S(x, d_i) < S(y, d_i)$ **then**
      add $d_i$ to $D_y$, and $x$ forwards a copy of the message to $y$ if $y$ does not have it
    **else**
      add $d_i$ to $D_x$
    **end if**
  **end for**
**end while**

Fig. 1. Our multicast algorithm Multi-Sosim

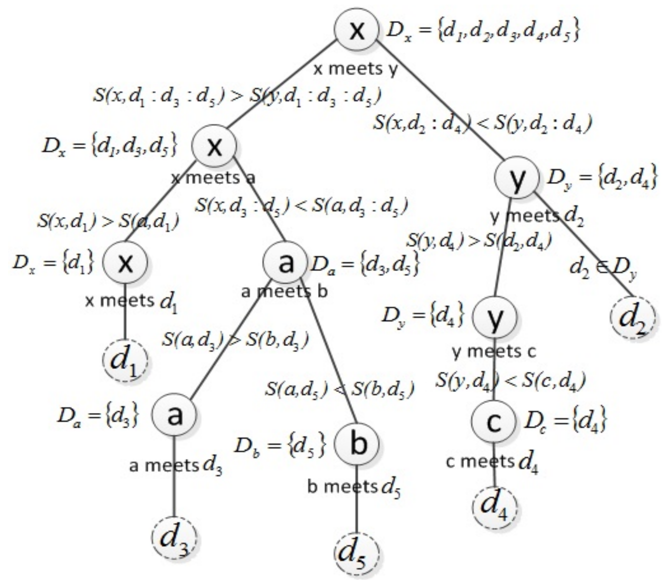



Fig. 2. A tree showing the multicast process. The notation "$S(x, d_i : d_j : d_k) > S(y, d_i : d_j : d_k)$" is a shortened form of "$S(x, d_i) > S(y, d_i)$ and $S(x, d_j) > S(y, d_j)$ and $S(x, d_k) > S(y, d_k)$ ".

same rule. The multicast tree will grow like this until all of the destinations are reached.

In the Multi-Sosim algorithm, the destinations share the path until the message holder meets another node. Regardless of whether that node is a newly met node or not, the destinations will be split into the two meeting nodes. One alternative is that the message holder can only consider splitting the destinations if it meets a new node whose destination set is empty. In that case, the destinations can share the paths longer. We refer to this variation as the *Multi-FwdNew* algorithm. Another opposite alternative is not to let the destinations share any path at all. That is, the mutlicast is implemented by multiple unicasts where the message to each destination is delivered individually without considering path sharing. We refer to this variation as the *Multi-Unicast* algorithm.

### *B. Social Similarity based on Dynamic Social Features*

In this section, we first introduce dynamic social features and then present the formulas to calculate the social similarity of two nodes based on dynamic social features.

*1) Definition of dynamic social features:* Suppose we consider $m$ social features $\langle F_1, F_2, \cdots, F_m \rangle$ of nodes in IMSNs. We associate each individual node with a vector of its social features. For convenience, we use a node's label as its vector's label. Thus, a node $x$ has a vector $x$ of $\langle x_1, x_2, \cdots, x_m \rangle$ and a node $y$ has a vector $y$ of $\langle y_1, y_2, \cdots, y_m \rangle$. A node $x$'s dynamic social features are contained in its vector, which is $< x_1, x_2, \cdots, x_m >= \left\langle \frac{M_1}{M_{total}}, \frac{M_2}{M_{total}}, \frac{M_3}{M_{total}}, \cdots \frac{M_m}{M_{total}} \right\rangle$, where $M_i$ is the number of meetings of node $x$ with nodes whose value $f_i$ of feature $F_i$ is the same as that of destination $d$, and $M_{total}$ is the total number of meetings of node $x$ with any other node in the recent history we observe. Thus, each $x_i$ $(1 \leq i \leq m)$ is in the range of $[0, 1]$. The length of the recent history can be set according to applications. We can look at a history starting from the beginning until the time we need to make a routing decision or we can adopt a sliding window to purge old contacts periodically after new ones come in over time.

Though not formally defined as dynamic social features, the idea of this concept was first put forward in our unicast routing algorithm in [7]. In dynamic social features, we not only record if a node has the same social feature values with the destination, but also record the frequency this node has met other nodes which have the same social feature values during the recent time interval we observe. For example, we not only record that 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 in the observed time window. Thus we can have a more accurate way to choose the best forwarders in multicast. For example, suppose the destination has social feature values *New Yorker* and *Student* and we have two candidate nodes $A$ and $B$, both of which are *New Yorkers* and *Students*. Nodes $A$ and $B$ are equally good forwarders if we just look at their static social feature values.

However, if we know that $A$ has met *New Yorkers* $90\%$ of the time and *Students* $80\%$ of the time and $B$ has met *New Yorkers* $60\%$ of the time and *Students* $40\%$ of the time during the time interval we observe, then obviously $A$ is a better forwarder. In our unicast paper [7], both the theoretical analysis and simulation results indicate that our unicast algorithm using dynamic social features performs better than the unicast ones using static social features. Inspired by the preliminary results in unicast, in this paper, we apply dynamic social features to multicast to further improve its performance.

*2) Calculation of social similarity:* With the node's dynamic social features defined, we can use the following similarity metrics derived from data mining [4] to compare the social similarity $S(x, y)$ of nodes $x$ and $y$. All of these metrics are normalized to the range of $[0, 1]$.

- **Tanimoto similarity**

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

- **Cosine similarity**

  It measures the similarity of $x$ and $y$ as: $S(x, y) = \frac{x \cdot y}{\sqrt{(x \cdot x)(y \cdot y)}}$.

- **Euclidean similarity**

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

- **Weighted Euclidean similarity**

  In addition to the basic Euclidean similarity mentioned above, we also employ the weighted Euclidean similarity to favor the social features that are more influential to the delivery of the packet. To determine the weight of a social feature, we use the Shannon entropy [9] which quantifies the expected value of the information contained in the feature [12]. The Shannon entropy for a given social feature is calculated as: $w_i = -\sum_{i=1}^{k} p(f_i) \cdot log_2(f_i)$, where $w_i$ is the Shannon entropy for feature $F_i$, vector $\langle f_1, f_2, \cdots f_k\rangle$ contains the possible values of feature $F_i$, and $p$ denotes the probability mass function of $F_i$. The weighted Euclidean similarity normalized to the range of $[0, 1]$ is: $S(x, y) = 1 - \frac{\sqrt{\sum_{i=1}^{m} w_i \cdot (y_i - x_i)^2}}{\sqrt{\sum_{i=1}^{m} w_i}}$.

To find the best fit for our simulated context, we compared Tanimoto, Cosine, Euclidean, and Weighted Euclidean similarity metrics by performing delegation forwarding routing algorithm [7]. Results show that all of the metrics performed similarly in delivery ratio, latency, and forwardings. We therefore decide to use the Euclidean metric in our multicast algorithm since it does not require the calculation of additional weighting values and performs slightly better than Tanimoto and Cosine in latency.

## IV. SIMULATIONS

In this section, we evaluate the performance of our multicast algorithm by comparing it with the existing ones using a custom simulator written in Java. The simulations were conducted using a real conference trace [8] reflecting an IMSN created at IEEE Infocom 2006. The trace recorded conference attenders' encounter history using Bluetooth small devices (iMotes) for four days at the conference in Miami. The trace dataset consists of two parts: *contacts* between the iMote devices that were carried by participants and self-reported *social features* of the participants which were collected using a questionnaire form. The six social features extracted from the dataset were *Affiliation, City, Nationality, Language, Country,* and *Position*.

### A. Comparison with Existing Algorithms

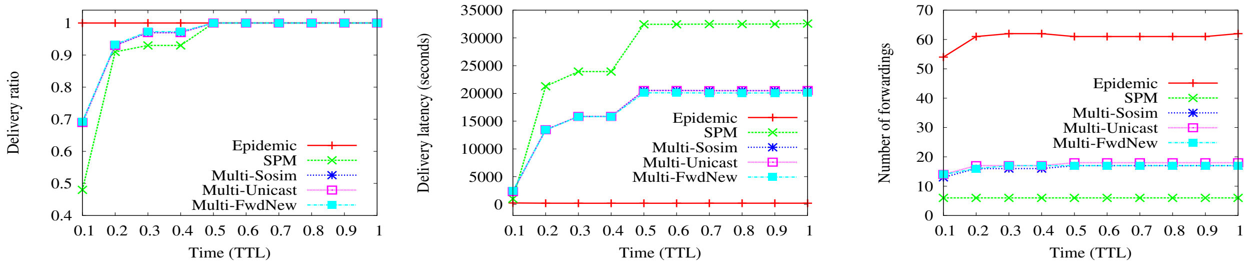
We compared our algorithm with the following related multicast protocols.

1) *The Epidemic Routing Algorithm* (Epidemic) [10]: The message is spread epidemically throughout the network until it reaches all of the destinations.
2) *The Social-Profile-based Multicast Routing Algorithm* (SPM) [2]: 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 features.
4) *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.
5) *Variation* 2 *of the Multi-Sosim Algorithm* (Multi-Unicast): The message to multiple destinations is delivered by multiple unicasts, where each unicast is conducted using dynamic social features.

### B. Evaluation Metrics

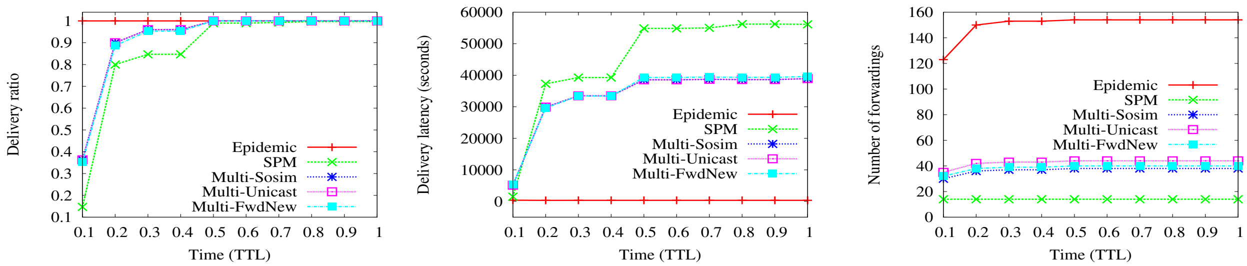
We use three important metrics to evaluate the performance of the multicast algorithms. Since a multicast involves multiple destinations, we define a *successful multicast* as the one that successfully delivers the message to all of the destinations.

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 all of the multicast destinations and when all of the multicast destinations receive the message.



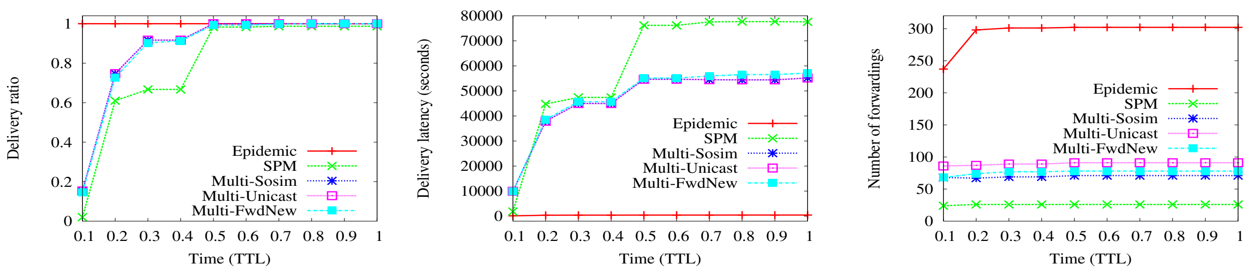

(a) Delivery ratio (b) Delivery latency (c) Number of forwardings

Fig. 3. Comparison of different algorithms with 2 destinations




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

Fig. 4. Comparison of different algorithms with 5 destinations




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

Fig. 5. Comparison of different algorithms with 10 destinations

3) *Number of forwardings*: The number of forwardings needed to deliver the message to all of the multicast destinations.

## C. Simulation Setup

In our simulations, we divided the whole trace time into $10$ intervals. Thus, $1$ TTL is $0.1$ of the total time length and $10$ TTLs is the length of the whole trace. For each of algorithms compared, we tried the size of the destination sets to be $2$, $5$, and $10$. In each experiment, we randomly generated a source and its destination set. Since the whole trace only contains node contact history of four days, the time interval we observed to calculate the dynamic social features was counted from the beginning of the trace up until the time we needed to make a routing decision. We ran each algorithm $300$ times and averaged the results of the evaluation metrics.

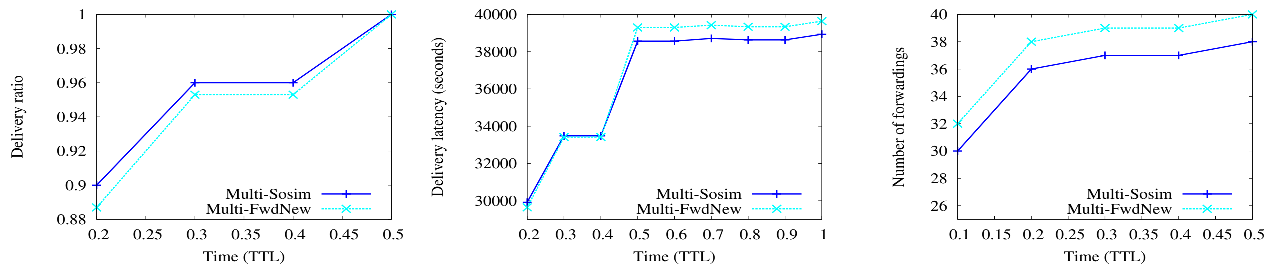
## D. Simulation Results

The simulation results with $2$, $5$, and $10$ destinations are shown in Figs. 3, 4, and 5, respectively. For the Epidemic algorithm, the results in all of the three figures show that, as expected, it has the highest delivery ratio and lowest delivery latency (almost close to $0$ compared with others in the figures) but highest number of forwardings.

The Multi-Sosim algorithm outperforms SPM by having a higher delivery ratio, a lower latency, and a little increase in the number of forwardings. The little increase in the forwardings indicates that Multi-Sosim is more active in delivering the message to the destinations. This confirms that using dynamic social features can more accurately capture node encounter behavior than using static social features in IMSNs.

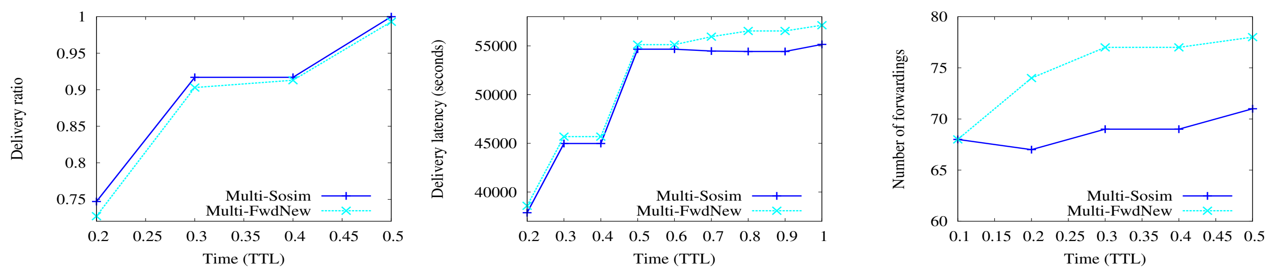
Multi-Sosim has similar delivery ratio and latency as Multi-Unicast as their curves are overlapped in the figures. But Multi-Sosim decreases the number of forwardings in Multi-Unicast by $7.7\%$, $16.7\%$, and $29.9\%$ in the cases of $2$, $5$, and $10$ destinations, respectively. This verifies that letting the destinations share the paths can reduce the forwarding cost, especially when the number of destinations goes up.

Multi-Sosim outperforms Multi-FwdNew in delivery ratio, latency, and the number of forwardings as can be seen in




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

Fig. 6. Comparison of MultiSosim and MultiFwdNew with 5 destinations




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

Fig. 7. Comparison of MultiSosim and MultiFwdNew with 10 destinations

Figs. 6 and 7 that zoom in the comparison results of the two algorithms with 5 and 10 destinations. 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 demonstrates that it is wise to reconsider the better forwarder for each destination whenever a message holder meets another node.

## V. Conclusion

In this paper, we proposed a novel multicast algorithm named Multi-Sosim and its variations for IMSNs where node connections are established impromptu and usually time-dependent, short-term, and dynamic. In Multi-Sosim, we used 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 in IMSNs. Simulation results based on a real trace representing an IMSN showed that our algorithm outperformed its variations and the existing SPM algorithm, which verified the appropriateness of utilizing the compare-split scheme in our multicast algorithm and the advantages of adopting dynamic social features over static ones. In our future work, we plan to test our algorithm using more traces in IMSNs as they become available.

## VI. Acknowledgement

This paper is supported in part by NSF CNS grants 1156712 and 1305302.

## References

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

[2] X. Deng, L. Chang, J. Tao, J. Pan, and J. Wang. Social profile-based multicast routing scheme for delay-tolerant networks. In *Proceedings of IEEE ICC*, pages 1857–1861, 2013.

[3] W. Gao, Q. Li, B. Zhao, and G. Cao. Multicasting in delay tolerant networks: a social network perspective. In *Proceedings of ACM MobiHoc*, 2009.

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

[5] U. Lee, S. Y. Oh, Lee K.-W., and M. Gerla. Relaycast: scalable multicast routing in delay tolerant networks. In *Proceedings of IEEE ICNP*, pages 218–227, 2008.

[6] M. Mongiovi, A. K. Singh, X. Yan, B. Zong, and K. Psounis. Efficient multicasting for delay tolerant networks using graph indexing. In *Proceedings of IEEE INFOCOM*, 2012.

[7] D. Rothfus, C. Dunning, and X. Chen. Social-similarity-based routing algorithm in delay tolerant networks. In *Proceedings of IEEE ICC*, pages 1862–1866, 2013.

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

[9] C. Shannon, N. Petigara, and S. Seshasai. A mathematical theory of communication. *Bell System Technical Journal*, 27(1):379–423, 1948.

[10] A. Vahdat, D. Becker, et al. Epidemic routing for partially connected ad hoc networks. Technical report, CS-200006, Duke University, 2000.

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

[12] J. Wu and Y. Wang. Social feature-based multi-path routing in delay tolerant networks. In *Proceedings of IEEE INFOCOM*, pages 1368–1376, 2012.

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

[14] W. Zhao, M. Ammar, and E. Zegura. Muticasting in delay tolerant networks: semantic models and routing algorithms. In *Proceedings of ACM WDTN*, pages 268–275, 2005.