

SOCIAL-SIMILARITY-BASED MULICAST ROUTING ALGORITHMS
IN IMPROMPTU MOBILE SOCIAL NETWORKS

by

Yuan Xu, B.E.

A thesis submitted to the Graduate Council of
Texas State University in partial fulfillment
of the requirements for the degree of
Master of Science
with a Major in Computer Science
December 2014

Committee Members:

Xiao Chen, Chair

Qijun Gu

Mina S. Guirguis

COPYRIGHT

By

Yuan Xu

2014

FAIR USE AND AUTHOR'S PERMISSION STATEMENT

Fair Use

This work is protected by the Copyright Laws of the United States (Public Law 94-553, section 107). Consistent with fair use as defined in the Copyright Laws, brief quotations from this material are allowed with proper acknowledgement. Use of this material for financial gain without the author's express written permission is not allowed.

Duplication Permission

As the copyright holder of this work I, Yuan Xu, authorize duplication of this work, in whole or in part, for educational or scholarly purposes only.

DEDICATION

This work is dedicated to my husband Lei, son Brandon, and daughter Claire who have been proud and supportive of my work and who have shared the uncertainties, challenges and sacrifices for completing this thesis. This work is also dedicated to my parents, who have set good examples for hard work and persistence, and who have instilled in me the inspiration to set high goals and the confidence to achieve them.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr. Xiao Chen, for her excellent guidance, encouragement, patience, and providing me with a great research environment. Without her guidance and persistent help, this thesis would not have been possible.

I wish to express sincere thanks to my committee members, Dr. Qijun Gu and Dr. Mina S. Guirguis, for their constant support and insightful comments on this thesis. I appreciate their support. I am grateful to all the members of the Computer Science department for assisting me in many different ways.

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	v
LIST OF FIGURES	viii
ABSTRACT.....	ix
CHAPTER	
1. INTRODUCTION.....	1
1.1 Delay Tolerant Networks (DTNs)	1
1.2 Mobile Social Networks (MSNs)	2
1.3 Unicast, Broadcast, and Multicast Routing Problems	3
1.4 Multicast Routing Protocols in IMSNs.....	4
1.5 Contributions	7
1.6 Organization	8
2. RELATED WORKS	9
2.1 Multicast Algorithms in DTNs	9
2.2 Social-based Multicast Algorithms in MSNs	10
3. DYNAMIC SOCIAL FEATURES AND SOCIAL SIMILARITY.....	13
3.1 Static Social Features.....	13
3.2 Dynamic Social Features	15
3.3 Enhanced Dynamic Social Features	15
3.4 Calculation of Social Similarity Metrics	16
3.4.1 Tanimoto Similarity	17
3.4.2 Cosine Similarity.....	17
3.4.3 Euclidean Similarity	18
3.4.4 Weighted Euclidean Similarity	18
4. SOCIAL-SIMILARITY-BASED MULTICAST ROUTING PROTOCOL.....	19
4.1 Social-similarity-based Multicast Routing Algorithm.....	19

LIST OF FIGURES

Figure	Page
4.1 Pseudo code of the Multi-Sosim algorithm	20
4.2 A tree structure showing the multicast process.....	21
6.1 Comparison of Tanimoto, Cosine, Euclidean, and Weighted Euclidean social similarity metrics	30
6.2 Comparison of Epidemic, SPM, and Multi-Sosim algorithms with 2 destinations	33
6.3 Comparison of Epidemic, SPM, and Multi-Sosim algorithms with 5 destinations	34
6.4 Comparison of Epidemic, SPM, and Multi-Sosim algorithms with 10 destinations	35
6.5 Comparison of Multi-Sosim, Multi-Unicast, and Multi-FwdNew with 5 destinations	37
6.6 Comparison of Multi-Sosim, Multi-Unicast, and Multi-FwdNew with 10 destinations	38
6.7 Comparison of Multi-Sosim and E-Multi-Sosim with 5 destinations	40
6.8 Comparison of Multi-Sosim and E-Multi-Sosim with 10 destinations	41

ABSTRACT

Mobile Social Networks (MSNs) where people contact each other through mobile devices have become increasingly popular. In this thesis, we study a special kind of MSNs formed impromptu (IMSNs) 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) routing 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 work, we propose the concept of dynamic social features and enhanced dynamic social features to capture people's contact behavior. Based on them, we design a novel social-similarity-based multicast algorithm (Multi-Sosim) and its enhancement (E-Multi-Sosim). Simulation results using a real conference trace representing an IMSN show that the E-Multi-Sosim algorithm performs better than the Multi-Sosim algorithm, which outperforms its variations and the existing one using static social features.

CHAPTER 1

Introduction

In this chapter, we introduce the concept of delay tolerant networks, mobile social networks, and the related routing problems.

1.1 Delay Tolerant Networks (DTNs)

Delay tolerant networks (DTNs) are a special class of wireless mobile networks which are characterized by large network delay, frequent mobility, limited cache space, lack of continuous network connectivity, etc (Cerf et al., 2007; Jain, Fall, & Patra, 2004). In such networks there is no guarantee of contemporaneous end-to-end paths, which makes the problem of routing much more complex. The routing protocols use store-carry-forward mechanism to propagate messages. When two nodes move within each other's transmission range, they communicate directly and become neighbors during that time period. When they move out of their ranges, their contact is lost. The message to be delivered needs to be stored in the local buffer until a contact occurs in the next hop.

These networks have a variety of applications. For example, a DTN could appear in connected vehicle networks (Burgess, Gallagher, Jensen, & Levine, 2006) where each vehicle is equipped with a radio transceiver that allows it to communicate with others. In this network, all vehicles will help each other forward messages. When a vehicle moves into the transmission range of a source, it receives data transmitted by the source. This vehicle can travel again and once it moves into the transmission range of the destination, it will forward data to the destination. Messages will experience significant delays due to

the intermittent connectivity of vehicles. Similarly, DTNs can also be formed in satellite communication networks (Wyatt, Burleigh, Jones, Torgerson, & Wissler, 2009), village area networks (Pentland, Fletcher, & Hasson, 2004), and mobile social networks (Scott, Crowcroft, Hui, & Diot, 2006).

1.2 Mobile Social Networks (MSNs)

As more portable, affordable, and powerful mobile devices such as laptops, smart phones, and tablets are developed and improved rapidly, mobile social networks (MSNs) are more ubiquitous in our daily lives and have become a hot research topic these days. In such networks, people move around and contact with one another through their mobile devices. MSN can be considered as a type of DTNs that involve social factors which reflect human behavior. It lacks continuous end-to-end connections between nodes, due to node mobility and limited transmission range.

In this thesis, our research focuses on 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 networks* (IMSNs). The IMSN allows people to communicate in a lightweight mechanism based on contact opportunities via local wireless bandwidth such as Bluetooth without a network infrastructure.

The links between nodes in IMSN are time-dependent, unstable, and short-term as people come and go at events. Therefore continuous network connectivity is not guaranteed. To illustrate the characteristics of IMSN, consider an IMSN formed by participants using small Bluetooth devices to record their contact with each other at a conference. They get connected during the conference and disconnected when the

conference finishes. A participant may want to send a file to one or more people attending the conference. The possible transmission path depends on his encounters with other users of the IMSN, and hence changes with space and time. It is also difficult to predict the transmission because of the unpredictable movement of the nodes. For example, two participants who have common interests may contact more frequently than others. All of these factors make the routing problems in IMSNs more challenging.

1.3 Unicast, Broadcast, and Multicast Routing Problems

In wireless network, there are three communication mechanisms: unicast, broadcast, and multicast. Unicast is the term used to describe the communication where the message is sent from a single source to a specified destination in the network. In this case there is just one source, and one destination. Broadcast is the term used to describe the communication where the message is sent from a single source to all other nodes in the network. In this case there is one source, and all other nodes are as destinations. Multicast is the term used to describe the communication where the message is sent from a single source to a set of destinations in the network. In this case there are one source and multiple destinations. From the definitions, we can see that both unicast and broadcast are special cases of multicast where the group of recipients is one node for unicast or the entire network for broadcast. Therefore, we are going to focus on multicast in this thesis.

Multicast has various and important applications in IMSNs. For example, in a conference, presentations are delivered to inform the participants about the newest technology (Hui et al., 2005); In an emergency scenario, information regarding local

conditions and hazard levels is disseminated to the rescue workers (Zhao, Ammar, & Zegura, 2005); And in campus life, school information is sent to a group of student mobile users over their wireless interfaces (Wang & Chen, 2001).

1.4 Multicast Routing Protocols in IMSNs

We already know that IMSNs are special cases of DTNs which involve social factors. Nodes in IMSNs can only communicate through a store-carry-forward fashion. The conventional ad-hoc network routing schemes, such as DSR (Johnson & Maltz, 1996), LAR (Ko & Vaidya, 2000), DSDV (Perkins & Bhagwat, 1994), AODV (Perkins, Royer, & Das, 2002), etc., would fail. Routing in IMSNs requires a new model that consists of a sequence of independent, local forwarding decisions, based on the current connectivity information and the predictions of future connectivity to suit its distributed and dynamic nature.

Most of the existing multicast algorithms focus on general DTNs (Lee, Oh, Lee, & Gerla, 2008; Mongiovi, Singh, Yan, Zong, & Psounis, 2012; Wang & Wu, 2012; Xi & Chuah, 2009; Zhao et al., 2005) without considering social factors. There are few multicast algorithms specifically designed for IMSNs where people play an important role. The closest we can find is the multicast algorithm proposed for MSNs by Deng, Chang, Tao, Pan and Wang (2013). 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 can be *language* and its value could be English. The intuition is that nodes have more common social features tend to meet more often. So the nodes having 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 nodes' dynamic contact behavior. For example, a student who put New York as his state in his social profile may actually attend a conference in Texas. In that case, the static information in his user profile cannot 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 thesis, we extend static social features to dynamic social features and enhanced dynamic social features to better reflect nodes' contact behavior and then develop new multicast algorithms based on these new concepts.

Though not formally defined as dynamic social features, the idea of this concept was first put forward in our unicast routing algorithm (Rothfus, Dunning, & Chen, 2013). In dynamic social features, we not only consider if a node shares the common social features with a destination, but also record the frequency this node has met other nodes which have the same social feature values as the destination during the time interval we observe. For example, if node A wants to send a message to a person from New York at a conference, we not only consider if node A, same as the destination, is a New Yorker, but also record that it has met New Yorker 90% of the time during the observation

interval at the conference. Unlike the static social features from user profiles, dynamic social features are time-related. So they change as user contact behavior changes over time. Thus we can have a more accurate way to choose the best forwarders in multicast. Take another example in which 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 the case of the unicast algorithm (Rothfus et al., 2013), both the theoretical analysis and simulation results indicate that our algorithm based on the idea of dynamic social features performs better than the existing unicast ones using static social features. Inspired by these preliminary results in unicast, in this thesis, we apply dynamic social features to multicast to further improve its performance.

The idea of the enhanced dynamic social features is inspired by the fact that we need to make a further decision to choose the better forwarder when two nodes have the same contact frequency in the above dynamic social features. In this thesis, we propose an enhanced way to calculate the dynamic social features of nodes and also apply them to multicast.

In multicast, a message holder is expected to forward a message to multiple destinations. To reduce the overhead and forwarding cost, a multicast process usually results in a tree structure where the destinations share the routing path until the point that they have to be separated by some compare-split scheme. In our multicast, if a message

holder meets another node, the compare-split scheme is based on the comparison of the social similarity of each of the destinations with the message holder and with the meeting node using dynamic social features. That is, whichever, either the message holder or the meeting node, is more socially close to the specific destination will have a higher chance to deliver the message to it and thus should relay the message to that destination.

1.5 Contributions

The main contributions of this thesis are summarized as follows:

- 1) We introduce the new concepts of dynamic social features and enhanced dynamic social features.
- 2) We propose a novel social-similarity-based multicast (Multi-Sosim) routing algorithm using dynamic social features and an enhanced multicast (E-Multi-Sosim) using enhanced dynamic social features for IMSNs.
- 3) We discuss two variations of Mutli-Sosim algorithm: 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 Multi-Unicast where multicast is implemented by multiple unicasts with each unicast conducted using dynamic social features.
- 4) To evaluate the performance of the Multi-Sosim algorithm, we compare it with SPM, Multi-FwdNew, and Multi-Unicast algorithms. The epidemic algorithm is included as a benchmark in the comparison. Simulation results show that Multi-Sosim outperforms SPM with a higher delivery ratio and lower latency with 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 over Multi-Unicast and Multi-FwdNew demonstrates that it is better to let the destinations share the paths and it is wise to reconsider a better forwarder for each destination whenever a message holder meets another node, respectively.

5) We also compare the Multi-Sosim algorithm with the E-Multi-Sosim algorithm. Simulation results show that the latter performs better than the former, which verifies that using enhanced dynamic social features can further improve the multicast performance.

1.6 Organization

The rest of the thesis is organized as follows: Chapter 2 references the related works; Chapter 3 presents the concepts of dynamic social features and enhanced dynamic social features, and different ways to calculate social similarity; Chapter 4 proposes a new multicast algorithm and discusses its possible variations; Chapter 5 gives the analysis of the Multi-Sosim algorithm; Chapter 6 shows the simulation results; and the conclusion is in Chapter 7.

CHAPTER 2

Related Works

In this chapter, we introduce multicast algorithms in the literature proposed for DTNs and MSNs.

2.1 Multicast Algorithms in DTNs

One efficient yet costly routing approach in DTNs is epidemic routing (Vahdat & Becker, 2000) where a message holder will forward a message to all of the nodes it comes into contact with, so that the message is spread epidemically throughout the network until it reaches all of the destinations. This approach is relatively simple because it requires no knowledge about the network. It provides a large amount of redundancy since all nodes receive the message making it achieve high delivery ratio and robustness. Additionally, since it tries every path, it delivers the message in the minimum amount of time so that the latency is very low. However, it has inevitable high forwarding cost because it uses all available paths instead of just a limited number. The epidemic algorithm will be used as a benchmark in our simulations.

Another basic algorithm in DTNs is wait (or direct delivery) (Jones & Ward, 2004), where the source does not forward copies to any intermediate nodes at all. It just waits and sends the message to the destinations when it meets the destination. It does not require any information about the network either. Due to the simplicity, there is only one message generated to each of the destinations in this approach, so the forwarding cost is

low. However, it only works if the source meets the destinations and the latency can be very high if it takes the source a long time to meet the destinations.

Most of other existing related multicast algorithms are designed for DTNs where social factors are not considered. In recent years, Zhao et al. (2005) introduce some new semantic models for multicast and conclude that the group-based strategy is suitable for multicast in DTNs. Lee et al. (2008) 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 and Chuah (2009) present an encounter-based multicast routing, and Chuah and Yang (2009) develop a context-aware adaptive multicast routing scheme. Mongiovi et al. (2012) use graph indexing to minimize the remote communication cost of multicast. Wang and Wu (2012) exploit the contact state information and use a compare-split scheme to construct a multicast tree with a small number of relay nodes.

2.2 Social-based Multicast Algorithms in MSNs

As social network applications explode in recent years, analysis of these network graphs shows that some nodes are the common acquaintances of other nodes and act as communication hubs (Motani, Srinivasan, & Nuggehalli, 2005; Srinivasan, Motani, & Ooi, 2006). Therefore, one promising way of predicting future contact probability is to use metrics such as *centrality* and *similarity* in network analysis to assess the message delivery probability of a node based on the connections in the graphs (Hui, Crowcroft, & Yoneki, 2008; Pietilainen & Diot 2012). Nevertheless, in these network graphs, past node contacts have been aggregated into a “static” social graph. As pointed out by

Hossmann, Spyropoulos, and Legendre (2009), Yang and Wu (2013), the “static” social graph has the tradeoff between time-related information lost and predictive capability.

Some other MSN routing algorithms use social features in user profiles to guide routing. Mei, Morabito, Santi, and Stefa (2011) find that individuals with similar social features tend to meet more often in MSNs. The individuals are characterized by high dimensional feature profiles, though usually only a small subset of important features are extracted and used in routing. Wu and Wang (2012) provide a systematic multicast routing approach by taking advantage of the structural property of hypercubes to resolve social feature differences between a source and destinations. Gao, Li, Zhao and Cao (2009) propose a community-based multicast routing scheme by exploiting node centrality and social community structures. Most recently, Deng, et al. (2013) 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.

The advantage of these social-feature-based approaches is that they do not need to record nodes’ contact history. They work well in social networks where the activities of individuals follow the information in their social profiles because the social relations among mobile users are more likely to be long-term and less volatile. But they may not be suitable for IMSNs where user activities are time-dependent and may deviate from their social features in their profiles. Therefore, in order to capture nodes’ dynamic behavior to steer the routing in the right direction, the multicast algorithm for IMSNs needs to be rethought about.

In this thesis, we propose novel routing algorithms using dynamic social features to capture nodes' contact behavior and a compare-split scheme to decide better routing paths for the destinations so as to improve multicast efficiency in IMSNs. To the best of our knowledge, this work is the first one that utilizes dynamic social features to guide multicast routing in IMSNs.

CHAPTER 3

Dynamic Social Features and Social Similarity

In this chapter, we first explain static social features as the preliminary for the dynamic social features, and then give the definitions of dynamic social features and enhanced dynamic social features. After those, we present the formulas to calculate the social similarity of two nodes based on these dynamic social features.

3.1 Static Social Features

To make the definition of dynamic social features more clear and distinguish it from static social features (Deng et al., 2013), we look at the latter first and give an example to explain how it is used in routing. Just as the name implies, static social features don't change with the time or space. For example, assume we consider four social features: $\langle F_1, F_2, F_3, F_4 \rangle$, which refer to $\langle \text{nationality, city, affiliation, language} \rangle$. Suppose destination D 's values in these four social features are: $\langle \text{USA, New York, student, English} \rangle$. These are the target social features that a source wants to reach, so we set the vector of D to $\langle 1, 1, 1, 1 \rangle$. Suppose there is a source that wants to send a message to D . If it has the same value as D for feature F_i , then the value in its F_i dimension is set to 1, otherwise it is set to 0. Suppose a source has nothing in common with the destination, then its vector is set to $\langle 0, 0, 0, 0 \rangle$. The routing process then attempts to resolve the differences between $\langle 0, 0, 0, 0 \rangle$ and $\langle 1, 1, 1, 1 \rangle$ via intermediate nodes. A possible path, represented by nodes' static social feature vectors, would be $\langle 0, 0, 0, 0 \rangle \rightarrow \langle 1, 0, 0, 0 \rangle \rightarrow \langle 1, 0, 0, 1 \rangle \rightarrow \langle 1, 0, 1, 1 \rangle \rightarrow \langle 1, 1, 1, 1 \rangle$. In each hop, the message is

passed on to a node that shares more common social features with the destination since it is expected to have a higher probability to deliver the message to the destination.

However, in the applications of IMSNs, people's static social features do not always reflect their dynamic behavior in reality. For example, consider a student from New York attends a conference in Texas. A simple feature value in his social profile will not be sufficient to reflect his dynamic behavior. Another situation is that in real life, a person mostly communicates with others who have more than one social features in common. So if we just look at 1 or 0 difference in their social features, then it is hard to tell which one is more socially similar to that person. Take an example where a message needs to be sent to multiple destinations at a conference and we just consider two social features $\langle \text{city}, \text{affiliation} \rangle$. Suppose there is a destination D whose social feature values in these two dimensions are $\langle \text{New York}, \text{student} \rangle$. If two candidate forwarders A and B both have the same social feature values as D , then their social vectors will both be set to $\langle 1, 1 \rangle$, which makes them indistinguishable.

Therefore, we propose a more accurate way to evaluate nodes' delivery probabilities by taking their dynamic contact behavior into account. We look at the nodes' past meeting ratios. For the above example, if node A meets New Yorkers 90% of the time and students 80% of the time, denoted by the frequency vector $\langle 90\%, 80\% \rangle$, while node B 's frequency vector for the same features are $\langle 60\%, 40\% \rangle$ during the time we observe, then we can tell node A is a better message forwarder than B . We refer to the frequency vector representing a node meeting with other nodes that have the same social feature values as the destination as the *dynamic social features* of that node. Its definition is as follows.

3.2 Dynamic Social Features

Suppose we consider m social features $\langle F_1, F_2, \dots, F_m \rangle$ of nodes in IMSNs. We associate each individual node with a vector of its dynamic social features. For convenience, we use a node's label as its vector's label. Thus, a node x has a vector x of length m : $\langle x_1, x_2, \dots, x_m \rangle$ and a node y has a vector y of length m : $\langle y_1, y_2, \dots, y_m \rangle$. A node x 's dynamic social features are contained in its vector, which is:

$$\langle x_1, x_2, \dots, x_m \rangle = \langle \frac{M_1}{M_{total}}, \frac{M_2}{M_{total}}, \frac{M_3}{M_{total}}, \dots, \frac{M_m}{M_{total}} \rangle \quad (3.1)$$

where M_i is the number of meetings of node x with nodes whose value f_i of feature F_i is the same as that of destination d , and M_{total} is the total number of meetings of node x with any other node in the history we observe. Thus $0 \leq x_i \leq 1$ for all $1 \leq i \leq m$.

Dynamic social features, as can be seen in the definition, not only record if a node has the same social feature values as the destination, but also record the frequency this node has met other nodes which have the same social feature values as the destination. Unlike static social features from user profiles, dynamic social features are time-related, so they change as user activities change over time. And thus we can have more accurate information to make routing decisions.

3.3 Enhanced Dynamic Social Features

The above definition of dynamic social features still have a little problem. For example, if node A has met 10 people in total and 6 of them are New Yorkers while B has met 100 people and 60 of them are New Yorkers. Based on the above definition,

they have the same frequency to meet New Yorkers, which makes them indistinguishable in the likelihood to deliver messages to New Yorkers.

Therefore, we propose an enhanced way to evaluate the social closeness between two nodes. Similarly, suppose we consider m social features $\langle F_1, F_2, \dots, F_m \rangle$ of nodes in IMSNs. A node x has a vector x of length m : $\langle x_1, x_2, \dots, x_m \rangle$, and for all $1 \leq i \leq m$, we calculate x_i as:

$$x_i = \left(\frac{M_{i+1}}{M_{total}+1} \right)^{p_i} * \left(\frac{M_i}{M_{total}+1} \right)^{1-p_i} \quad (3.2)$$

where $p_i = \frac{M_i}{M_{total}}$. M_i and M_{total} have the same meaning as before. This definition predicts x_i by looking at the next meeting of node x with another node. So the total meeting times will be $M_{total} + 1$. The first part $\left(\frac{M_{i+1}}{M_{total}+1} \right)^{p_i}$ means that there is p_i probability that x will have a “good” meeting with another node with the same social feature f_i next time. In this case, M_i will also be incremented by 1. The second part $\left(\frac{M_i}{M_{total}+1} \right)^{1-p_i}$ means that there is $1 - p_i$ probability for x not to meet a node with the same social feature f_i next time. In that case, M_i will remain the same. The definition for x_i then takes the geometric mean of the two parts. It is easy to see that each x_i satisfies $0 \leq x_i \leq 1$.

3.4 Calculation of Social Similarity Metrics

With the node’s dynamic social features defined, the next task is to use some similarity metric to compare the social similarity of two vectors.

To compare the social similarity $S(x, y)$ between nodes x and y , we can use the

following similarity metrics derived from data mining (Han, Kamber, & Pei, 2012). In our metrics, 1 means 100% identical and 0 means not similar at all. To deal with various values of social data, we normalize the outputs of all metrics to the range of [0, 1].

3.4.1 Tanimoto Similarity

The Tanimoto coefficient to measure the similarity of node x and node y is:

$$S(x, y) = \frac{x \cdot y}{x \cdot x + y \cdot y - x \cdot y} \quad (3.3)$$

where the notation $x \cdot y$ is the product of the two vectors.

For example, suppose we look at three social features: city, language, and position in the network. If the values of social features of destination D are: <New York, English, student>. 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 history we observe, then node x has a vector of $x = \langle 0.7, 0.93, 0.41 \rangle$. And node y is the destination who has a vector of $y = \langle 1, 1, 1 \rangle$. Using the Tanimoto metric in equation (3.2), we can get $S(x, y) = 0.82$.

3.4.2 Cosine Similarity

It measures the similarity of node x and node y as:

$$S(x, y) = \frac{x \cdot y}{\sqrt{(x \cdot x)(y \cdot y)}} \quad (3.4)$$

3.4.3 Euclidean Similarity

We can also use the Euclidean distance to measure a node's social similarity to another node. To make the similarity definition consistent, we normalize the original definition of Euclidean similarity to the range of $[0, 1]$ and subtract it from 1. Now the Euclidean similarity of x and y is defined as:

$$S(x, y) = 1 - \frac{\sqrt{\sum_{i=1}^m (y_i - x_i)^2}}{\sqrt{m}} \quad (3.5)$$

3.4.4 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 message. To determine the weight of a social feature, we use the Shannon entropy (Shannon, Petigara, & Seshasai, 1948) which quantifies the expected value of the information contained in the social feature (Wu & Wang, 2012). The Shannon entropy for a given social feature is calculated as:

$$\omega_i = - \sum_{i=1}^k p(f_i) \cdot \log_2(f_i) \quad (3.6)$$

where ω_i is the Shannon entropy for feature F_i , vector $\langle f_1, f_2, \dots, f_k \rangle$ 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 as follows:

$$S(x, y) = 1 - \frac{\sqrt{\sum_{i=1}^m \omega_i (y_i - x_i)^2}}{\sqrt{\sum_{i=1}^m \omega_i}} \quad (3.7)$$

CHAPTER 4

Social-similarity-based Multicast Routing Protocol

In this chapter, we propose the social-similarity-based multicast (Multi-Sosim) routing algorithm for IMSNs and discuss its two variations.

4.1 Social-similarity-based Multicast Routing Algorithm

The pseudo code of Multi-Sosim is shown in Figure 4.1. In the beginning, a source node s , also the initial message holder x , has a message to be delivered to a set of destinations $D_s = \{d_1, d_2, \dots, d_n\}$. We refer to D_s as the destination set of s . We initialize the destination sets of all of the other nodes to be empty. The routing process is started 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 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 directly. Next, we will 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} and put them into D_x and D_y by comparing the social similarity of each destination d_i with x and with y . The social similarity $S(x, y)$ of two nodes x and y is calculated based on the dynamic social features of nodes. 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.

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

- 1: Initialize the destination sets of all of the nodes except s to be empty
- 2: **while** not all of the destinations receive the message **do**
- 3: On contact between a message holder x and node y :
- 4: **if** $y \in D_x$ **then**
- 5: $/*$ Found the destination y $*/$
- 6: x forwards the message to y and removes y from D_x
- 7: **end if**
- 8: $/*$ Combine the destination sets of x and y $*/$
- 9: Let $D_{xy} = D_x \cup D_y$ and $D_x = D_y = \emptyset$
- 10: $/*$ Compare node social similarities and split the destinations in D_{xy} to D_x and D_y $*/$
- 11: **for** each destination $d_i \in D_{xy}$ **do**
- 12: $/*$ Calculate the social similarity $S(x, d_i)$ and $S(y, d_i)$, respectively $*/$
- 13: **if** $S(x, d_i) < S(y, d_i)$ **then**
- 14: add d_i to D_y , and x forwards the message to y if y does not have it
- 15: **else**
- 16: add d_i to D_x
- 17: **end if**
- 18: **end for**
- 19: **end while**

Figure 4.1: Pseudo code of the Multi-Sosim algorithm

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 Figure 4.2. 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 send 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 based on the dynamic social features, 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.

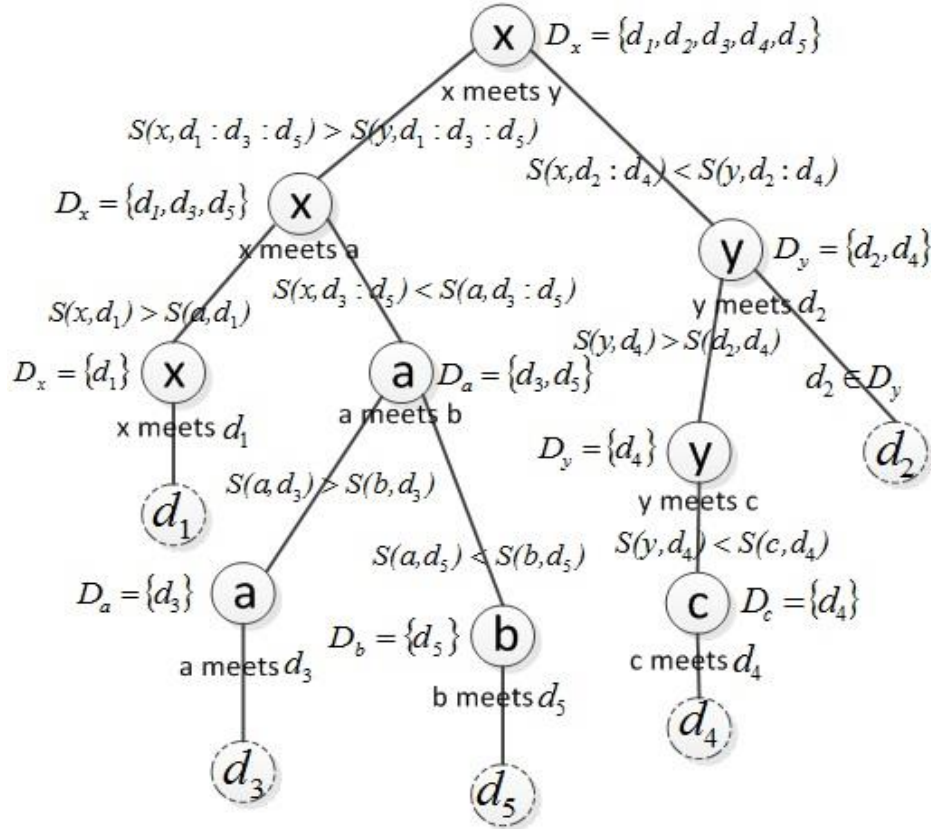


Figure 4.2: A tree structure showing the multicast process

4.2 Two Variations

In the above 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 a node met before, the destinations will be split. 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. That is, the multicast is implemented by multiple unicasts where each destination is reached individually. We refer to this variation as the Multi-Unicast algorithm.

CHAPTER 5

Analysis

In this chapter, we analyze the Multi-Sosim algorithm in terms of the number of forwardings and the number of copies.

5.1 The Number of Forwardings

Lemma 1. In the Multi-Sosim algorithm, if there is only one destination in the destination set D , it takes at most $\log n$ forwardings to reach that destination on average, where n is the total number of nodes in the network.

Proof. Consider a source node s which has a social similarity gap g to the destination. To reach the destination, the message will be delivered to a node with a smaller gap to the destination in each forwarding. Suppose the gap is updated l times before the message reaches the destination, and suppose the gap at the l th update is denoted as the random variable G_l . Assume the contact rate of nodes is independent of node similarity, a node is equally likely to meet another node with any particular similarity value. The next update of the gap occurs when it meets a node with a smaller gap than G_l , and all values above this level are equally likely.

Hence, we can write

$$G_{l+1} = G_l \times U \quad (4.1)$$

where U is independent of G_l and follows a uniform distribution on $(0, 1]$. By induction we then find the conditional expected value of G_{l+1} given G_l is:

$$E[G_{l+1}|G_l] = \frac{G_l}{2} \quad (4.2)$$

Hence, the expected value of G_l is:

$$E[G_l] = \frac{g}{2^l} \quad (4.3)$$

This process is like an expanding binary tree. The total number of nodes in the network is n and the depth of the binary tree is $l = \log n$. Thus, it needs on average $O(\log n)$ forwardings to reach a destination.

In IMSNs, node contact rate is related to social similarity. Two nodes contact more frequently if they are more socially similar. The gap G_l will be reduced more quickly than the above assumption case where node contact rate is independent of node similarity. Therefore, the number of forwardings in the Multi-Sosim algorithm will not exceed $O(\log n)$.

Theorem 1. The complexity of number of forwardings in the Multi-Sosim algorithm is $O(kn + 2k - 1)$ in the worst case and $O(k \log n + 2k - 1)$ in the average case, where n is the number of nodes in the network and k is the number of destinations in the multicast set.

Proof. In the Multi-Sosim algorithm, if node x multicasts to a destination set $D = \{d_1, d_2, \dots, d_k\}$, denoted by $x \rightarrow D$, there are three cases that $x \rightarrow D$ will change when x meets y .

1) Transmission: y is in the destination set. In this case, the message is delivered to y directly.

2) Split: y is more socially similar to some of the destinations in D . In this case, $x \rightarrow D_1, y \rightarrow D_2$, where $D_1 \cup D_2 = D$ and $D_1 \cap D_2 = \emptyset$.

3) Handover: y is more socially similar to all of the destinations in D than x . In this case, node x hands over the message and the destination set D to y and then deletes

its own copy.

A multicast process is composed of the forwardings caused by the above three cases which are calculated separately below.

1) Transmission: The number of transmissions in the multicast process is k because each of the k destinations will be reached once. Each transmission is counted as one forwarding. So the number of forwardings in transmissions is k .

2) Split: The number of splits in the multicast is $k - 1$. This can be proved by induction. In the base case, let $k = 2$ and D is split into two sets with only one element each, the number of splits is 1, which is $k - 1$. Assume the claim is correct for any $r < k$. Now if $r = k$ and set D is split into two sets D_1 and D_2 , where $D_1 \cup D_2 = D$ and $D_1 \cap D_2 = \emptyset$. Suppose the size of D_1 is r_1 and the size of D_2 is r_2 . Both r_1 and r_2 are less than k and $r_1 + r_2 = r = k$. According to the assumption, the number of splits in set D_1 is $r_1 - 1$ and the number of splits in set D_2 is $r_2 - 1$. Then the number of splits when D splits into D_1 and D_2 is: $r_1 - 1 + r_2 - 1 + 1 = r - 1 = k - 1$. That proves the claim. Each split is counted as a forwarding, so the number of forwardings in splits is $k - 1$.

3) Handover: The number of handovers in the multicast is about k for k destinations. Eventually there will be a set for each of the k destinations. Then each destination will be reached independently. For a particular destination d_i , in the worst case, the number of forwardings needed for a message holder to reach it is $O(n)$ after using all of the nodes in the network as relays. The average number of forwardings to reach it is $O(\log n)$ which is explained in Lemma 1. There are altogether k destinations, so the total number of forwardings needed in handover is $O(kn)$ for the

worst case and $O(k \log n)$ for the average case.

In total, the maximum number of forwardings in the multicast process is the summation of all of the above three cases, which is: $O(kn + 2k - 1)$ in the worst case and $O(k \log n + 2k - 1)$ in the average case.

5.2 The Number of Copies

Theorem 2. The number of extra copies produced in the Multi-Sosim algorithm is $2k - 1$, where k is the number of destinations in the multicast set.

Proof. A multicast process is composed of the copies produced by the above three cases which are calculated separately below.

1) Transmission: The number of transmissions in the multicast process is k because of k destinations. Each destination will eventually get a copy of the message. So the number of copies produced by transmissions is k .

2) Split: The number of splits in the multicast is $k - 1$. Each split produces an extra copy of the message. So the number of copies produced by splits is $k - 1$.

3) Handover: The handover does not produce any number of extra copies since the message holder will send a copy to the new message holder and delete its own copy.

In total, the number of extra copies produced in the multicast process is the summation of all of the above three cases, which is $2k - 1$.

CHAPTER 6

Simulations

In this chapter, we evaluate the performance of the Multi-Sosim algorithm. We first compare different similarity metrics to decide the metric we will use in our simulations. Then we compare the Multi-Sosim algorithm with its two variations and the existing algorithms. Finally we compare the Multi-Sosim algorithm with its enhancement.

6.1 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 destinations and when all of the destinations receive the message.
- 3) Number of forwardings: The number of forwardings needed to deliver the message to all of the destinations.

6.2 The Real Trace

The simulations were conducted using a real conference trace (Scott, Gass, Crowcroft, Hui, Diot, & Chaintreau, 2009) reflecting an IMSN created at Infocom 2006.

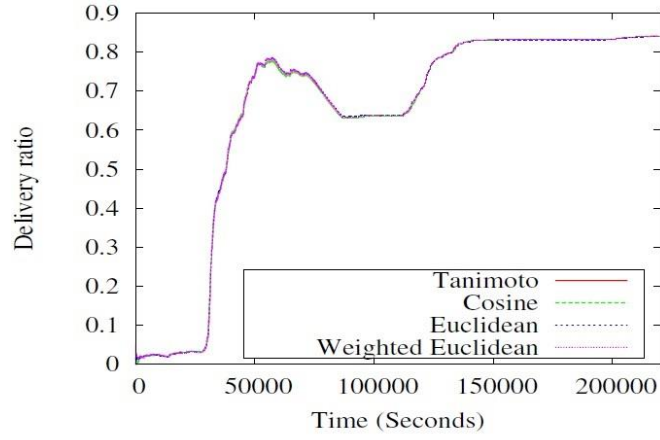
In total, 78 students and researchers carrying the Bluetooth small devices (iMotes) communicated at the IEEE Infocom 2006 conference in Miami for four days. The trace dataset consists of two parts: *contacts* between the iMote devices and the self-reported *social features* of the participants which were collected using a questionnaire form. The six social features extracted from the dataset were affiliation, city, nationality, language, country, and position. In total, 61 participants provided full social features, while others provided incomplete information. There were 128,979 contacts between the 61 participants over a period of 340,808 time slots in seconds.

6.3 Comparison of Social Similarity Metrics

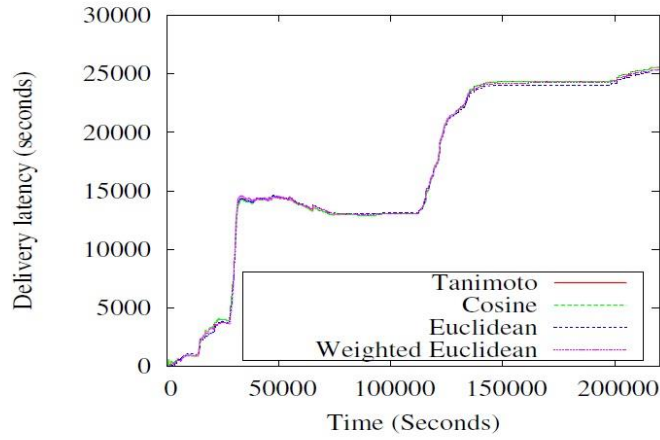
To find the best fit for our simulated context, we compared Tanimoto, Cosine, Euclidean, and Weighted Euclidean similarity metrics in unicast scenario (Rothfus et al., 2013). In the routing process, we apply the idea of delegation forwarding proposed by Erramilli, Crovella, Chaintreau, and Diot (2008) because it can bring down the expected cost of delivering messages. The main idea of delegation forwarding is that it assigns a quality and a level value to each node. The quality value of a node here is $S(x, y)$, and the level value is τ . Initially, the level value of each node is equal to its quality value. During the routing process, a message holder compares the quality of the node it meets with its own level. It only forwards the message to a node with a higher quality than its own level. In addition, the message holder raises its own level to the quality of the higher quality node. The result of delegation forwarding is that a node will forward a message only if it encounters another node whose quality metric is greater than any seen by the node so far.

We utilized the first two days of the data as the initial history and performed our simulations on the remaining days. We generated messages from a randomly chosen source to a randomly chosen destination every two seconds in the first 24 hours of the simulation. We then averaged five separate simulations of each algorithm with identical setups to mitigate the effect of any outliers in the performance. To perform a fair comparison of the algorithms, we set time-to-live of all of the packets to 9, meaning that a given packet can be transferred at most nine times so that the delivery ratio will not always be 100% during the whole time frame of the trace.

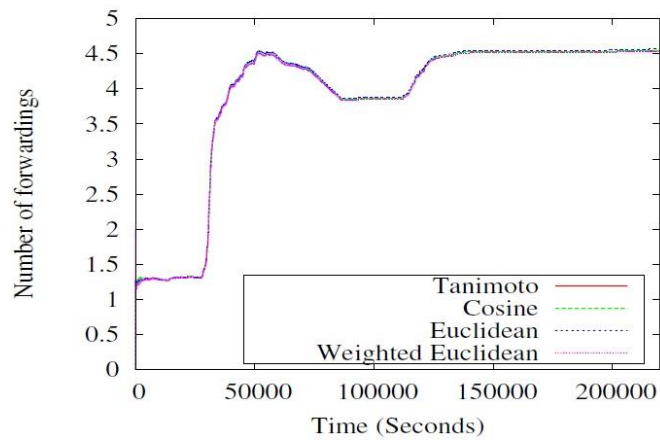
The simulation results in Figure 6.1 show that all of the similarity metrics performed similarly in delivery ratio, latency, and number of forwardings. We therefore decided to use the Euclidean metric since it did not require the calculation of additional weighting values and performed slightly better than Tanimoto and Cosine metrics in latency.



(a) Delivery ratio



(b) Delivery latency



(c) Number of forwardings

Figure 6.1: Comparison of Tanimoto, Cosine, Euclidean, and Weighted Euclidean social similarity metrics

6.4 Comparison of Multicast Algorithms

We compared our algorithm with the following related multicast protocols.

1) The epidemic algorithm (epidemic) (Vahdat & Becker, 2000): The message is spread epidemically throughout the network until it reaches all of the multicast destinations.

2) The social-profile-based multicast routing algorithm (SPM) (Deng et al., 2013): 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.

6) The enhanced Multi-Sosim algorithm (E-Multi-Sosim): This algorithm is similar to Multi-Sosim but it is based on the enhanced dynamic social features.

6.5 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 the algorithms compared, we tried the sizes of the destination sets to be 2, 5, and 10.

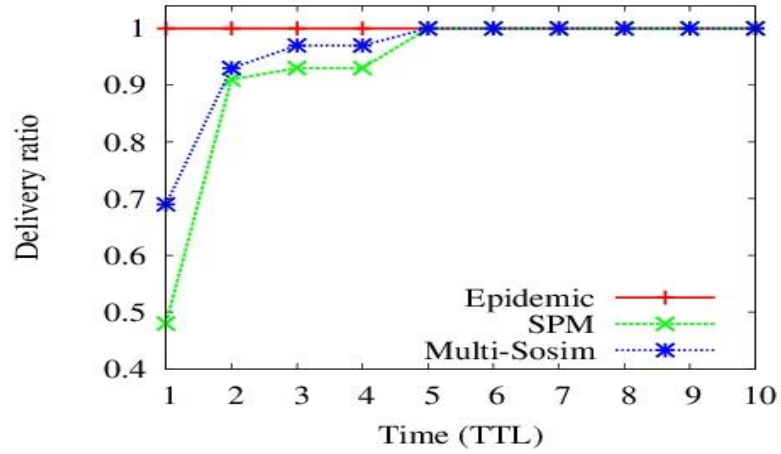
In each experiment, we randomly generated a source and its destination set. We ran each algorithm 300 times and averaged the results of the evaluation metrics.

6.6 Simulation Results

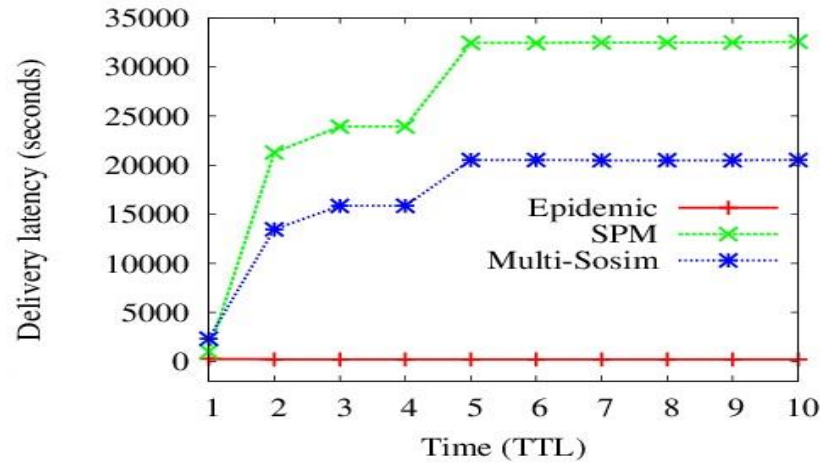
6.6.1 Comparison Results of Multi-Sosim and the Existing Algorithms

Figure 6.2, 6.3 and 6.4 show the simulation results of epidemic, SPM, and Multi-Sosim algorithms with 2, 5, and 10 destinations, 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.

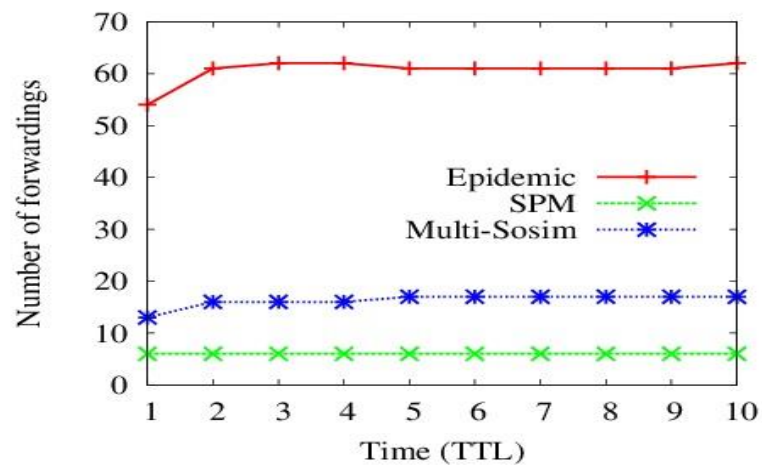
The Multi-Sosim algorithm outperforms SPM in having a higher delivery ratio and lower latency with a little increase in the number of forwardings. The little increase in the number of forwardings indicates that Multi-Sosim is more active in delivering the message to the destinations. This confirms that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



(a) Delivery ratio

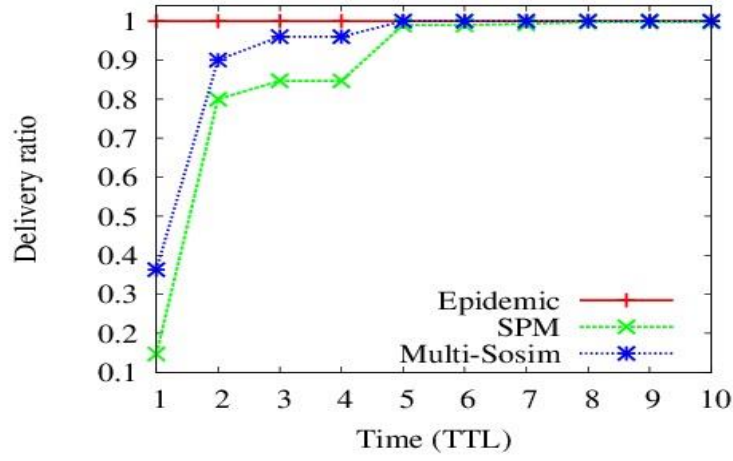


(b) Delivery latency

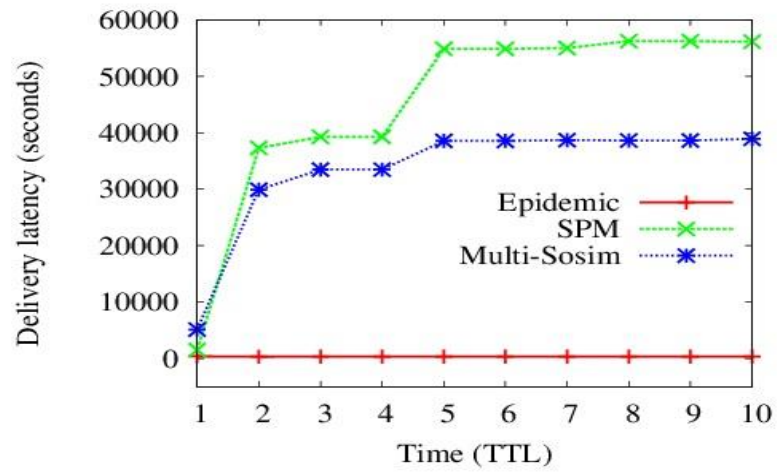


(c) Number of forwardings

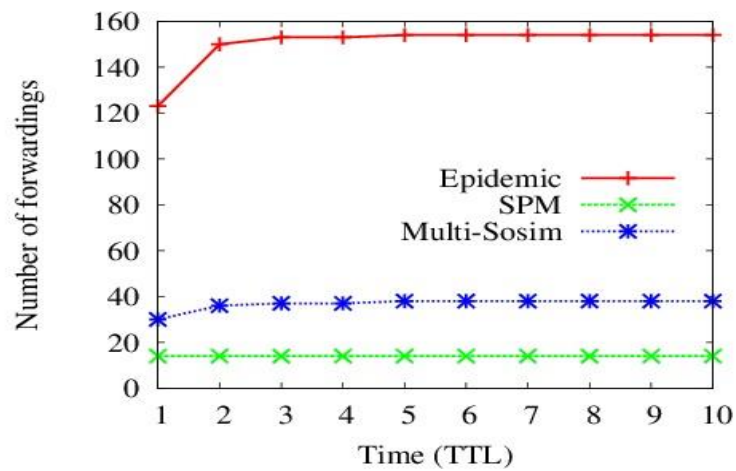
Figure 6.2: Comparison of Epidemic, SPM, and Multi-Sosim algorithms with 2 destinations



(a) Delivery ratio

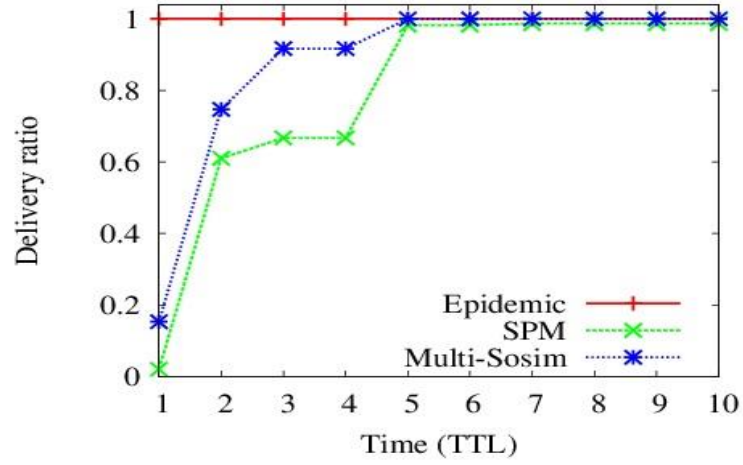


(b) Delivery latency

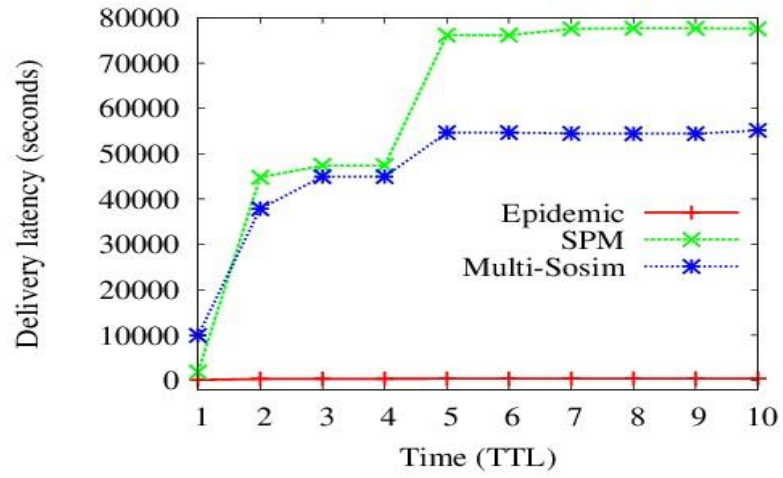


(c) Number of forwardings

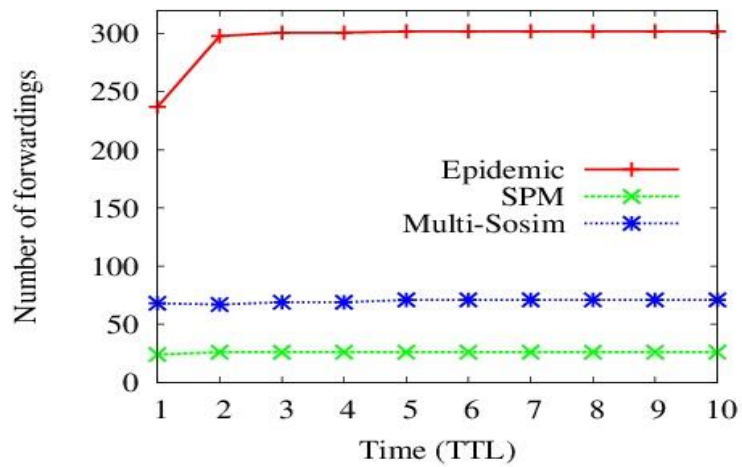
Figure 6.3: Comparison of Epidemic, SPM, and Multi-Sosim algorithms with 5 destinations



(a) Delivery ratio



(b) Delivery latency



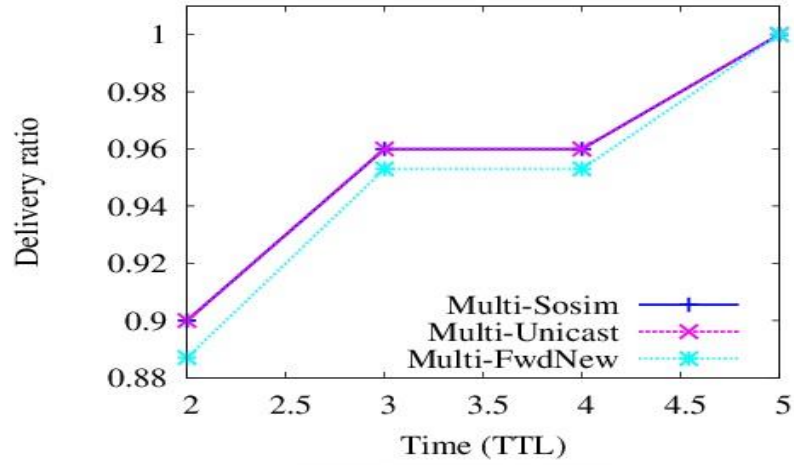
(c) Number of forwardings

Figure 6.4: Comparison of Epidemic, SPM, and Multi-Sosim algorithms with 10 destinations

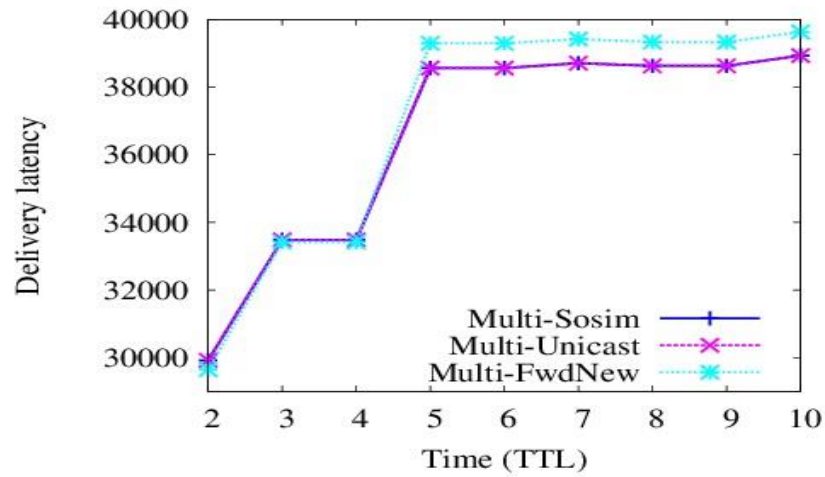
6.6.2 Comparison Results of Multi-Sosim and its Variations

Figure 6.5 and 6.6 show the zoom-in simulation results of Multi-Sosim, Multi-Unicast, and Multi-FwdNew algorithms with 5 and 10 destinations. 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 16.7% and 29.9% with 5 and 10 destinations, respectively. This verifies that letting the destinations share the path can reduce the forwarding cost, especially when the number of destinations goes up.

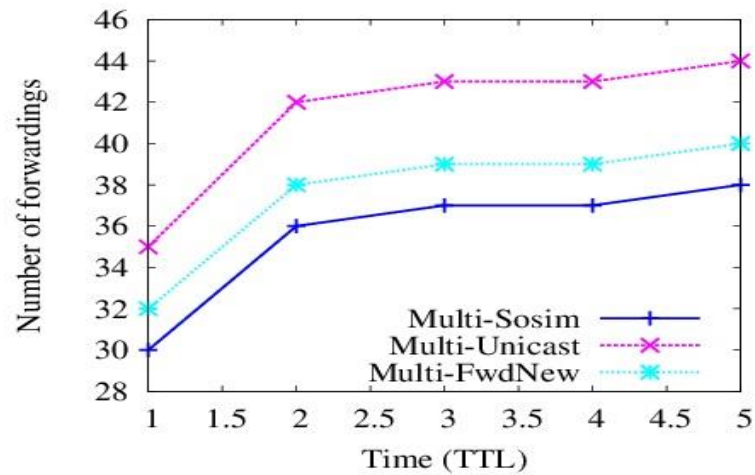
Multi-Sosim outperforms Multi-FwdNew in delivery ratio, latency, and the number of forwardings. With 5 destinations, the Multi-Sosim algorithm increases the delivery ratio by 1.5%, decreases latency by 2.0%, and decreases the number of forwardings by 6.7% comparing with Multi-FwdNew. With 10 destinations, 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.



(a) Delivery ratio

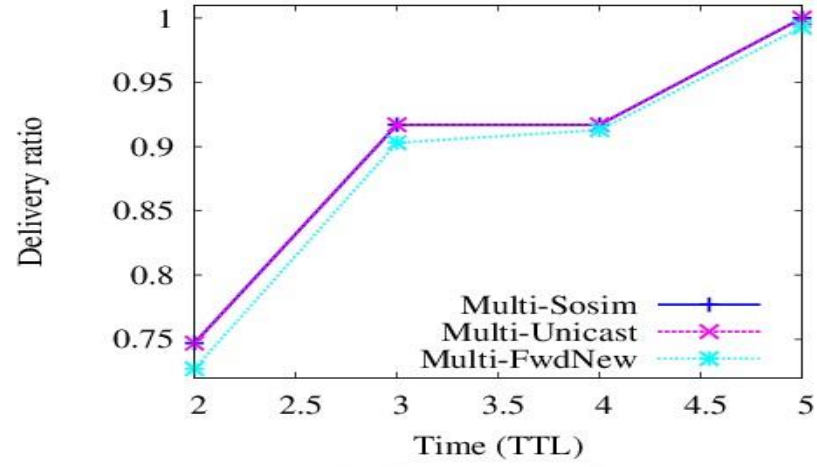


(b) Delivery latency

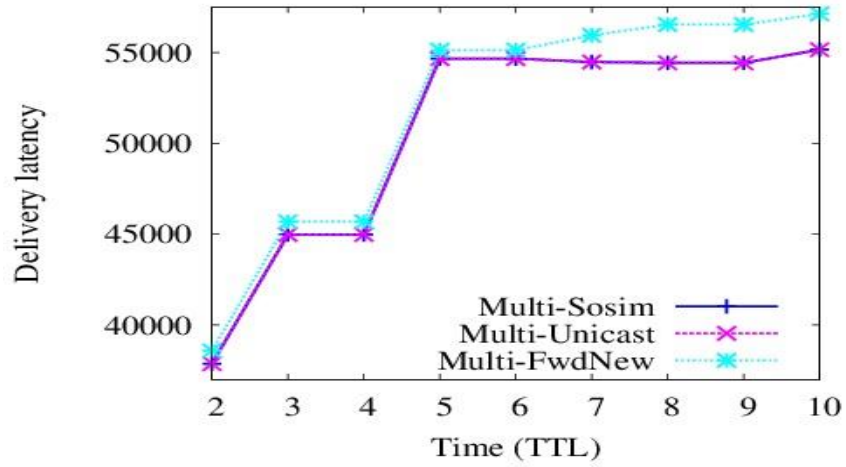


(c) Number of forwardings

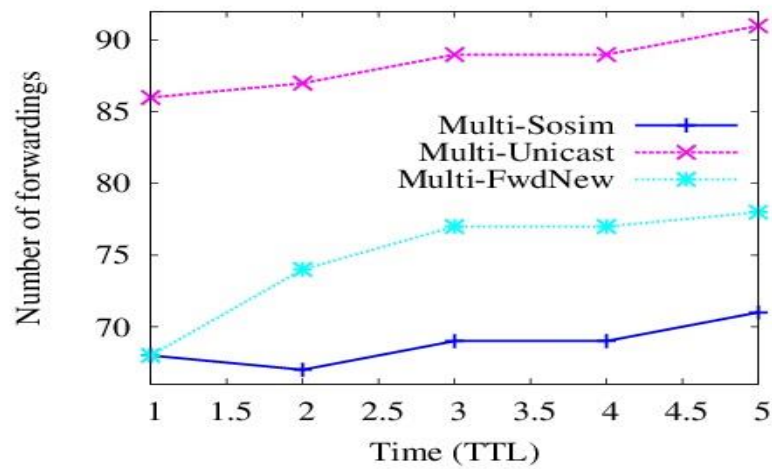
Figure 6.5: Comparison of Multi-Sosim, Multi-Unicast, and Multi-FwdNew with 5 destinations



(a) Delivery ratio



(b) Delivery latency

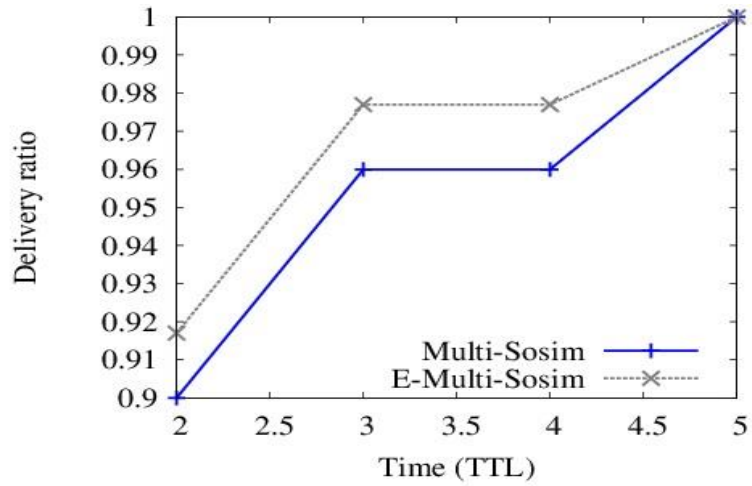


(c) Number of forwardings

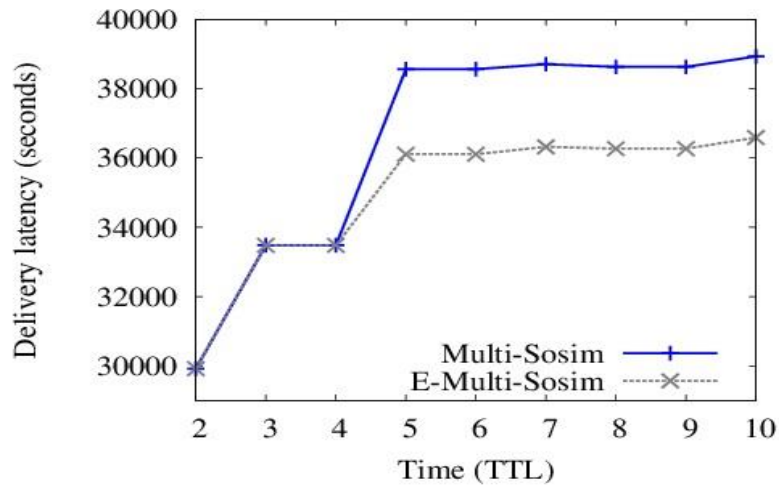
Figure 6.6: Comparison of Multi-Sosim, Multi-Unicast, and Multi-FwdNew with 10 destinations

6.6.3 Comparison Results of Multi-Sosim and its Enhancement

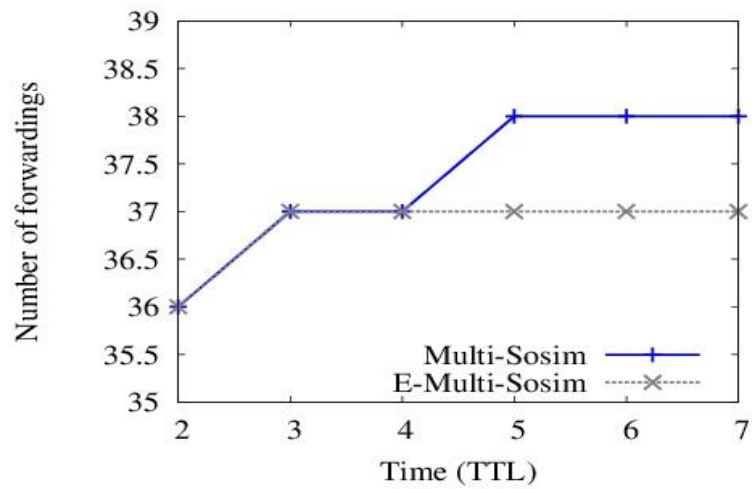
Figure 6.7 and 6.8 zoom in the comparison of Multi-Sosim and E-Multi-Sosim algorithms with 5 and 10 destinations. We can see that E-Multi-Sosim outperforms Multi-Sosim in delivery ratio, latency, and the number of forwardings. With 5 destinations, the E-Multi-Sosim algorithm increases the delivery ratio 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 demonstrates that we can capture nodes' dynamic contact behavior more accurately using enhanced dynamic social features.



(a) Delivery ratio

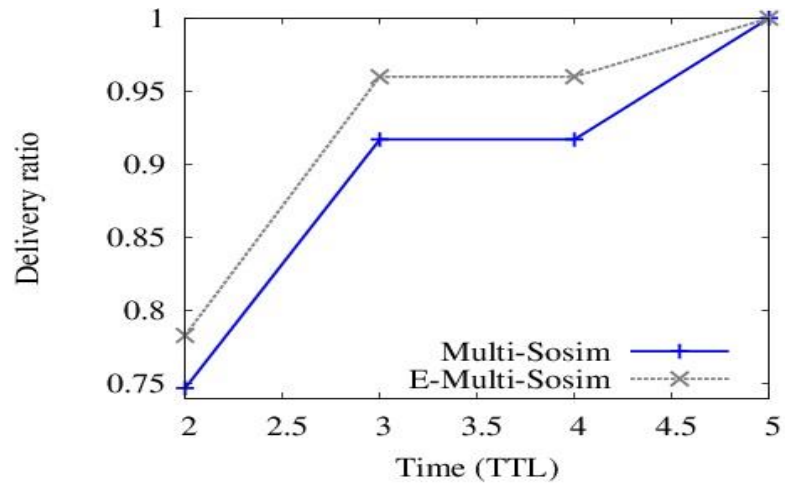


(b) Delivery latency

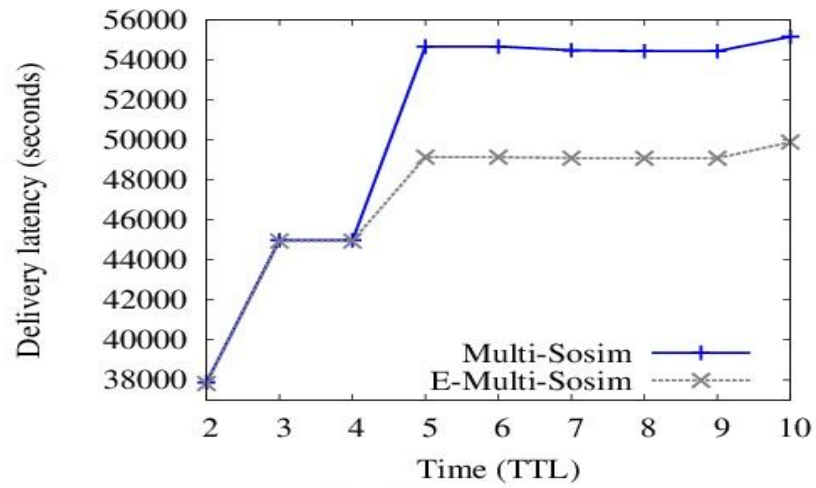


(c) Number of forwardings

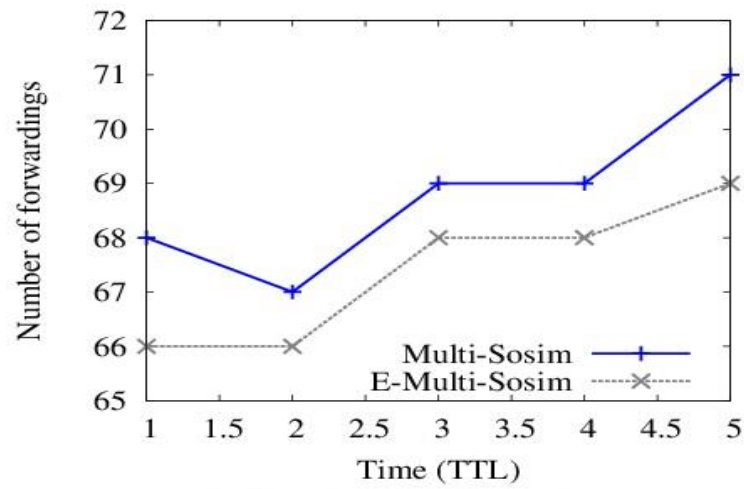
Figure 6.7: Comparison of Multi-Sosim and E-Multi-Sosim with 5 destinations



(a) Delivery ratio



(b) Delivery latency



(c) Number of forwardings

Figure 6.8: Comparison of Multi-Sosim and E-Multi-Sosim with 10 destinations

CHAPTER 7

Conclusion

In this thesis, we designed efficient multicast routing protocols for IMSNs where node connections are established impromptu and usually time-dependent, short-term, and dynamic. We introduced the concept of dynamic social features to capture nodes' contact behavior more accurately than the static social features. Then we proposed enhanced dynamic social features to further improve the accuracy of capturing nodes' contact behavior. Based on the dynamic social features and the enhanced one, we designed a novel multicast algorithm named Multi-Sosim and its enhancement E-Multi-Sosim for IMSNs. In both algorithms, a compare-split scheme was used to select the best relay node for each destination in each hop to improve multicast efficiency in IMSNs. We also studied the two variations of the Multi-Sosim algorithms: Multi-Unicast and Multi-FwdNew.

Simulation results using a real trace representing an IMSN showed that the Multi-Sosim algorithm outperformed the existing SPM algorithm and its variations, which verified the advantages of the dynamic social features over the static ones and the appropriateness of the compare-split scheme in our multicast algorithm. The E-Multi-Sosim algorithm performs better than the Multi-Sosim algorithm, which confirms that we can further capture nodes' dynamic contact behavior more accurately using enhanced dynamic social features.

In our future work, we plan to test our algorithm using more traces in IMSNs as they become available.

REFERENCES

- Burgess, J., Gallagher, B., Jensen, D., & Levine, B. N. (2006). Maxprop: routing for vehicle-based disruption-tolerant networks. In *Proceedings of IEEE INFOCOM* (pp. 1-11). doi:10.1109/INFOCOM.2006.228
- Cerf, V., Burleigh, S., Hooke A., Torgerson L., Durst, R., Scott, K., Fall, K., & Weiss, H. (2007). Dealy-Tolerant networking architecture. *IETF, Request for Comment (RFC): 4838*. Retrieved from <http://tools.ietf.org/pdf/rfc4838.pdf>
- Chuah, M., & Yang, P. (2009). Context-aware multicast routing scheme for disruption tolerant networks. *Journal of Ad Hoc and Ubiquitous Computing*, 4(5), 269-281. doi:10.1145/1163610.1163622
- Deng, X., Chang, L., Tao, J., Pan, J., & Wang, J. (2013). Social profile-based multicast routing scheme for delay-tolerant networks. In *Proceedings of IEEE ICC* (pp. 1857-1861). doi:10.1109/ICC.2013.6654791
- Erramilli, E., Crovella, M., Chaintreau, A., & Diot, C. (2008). Delegation forwarding. In *Proceedings of ACM MobiHoc* (pp. 251-260). doi:10.1145/1374618.1374653
- Gao, W., Li, Q., Zhao, B., & Cao, G. (2009). Multicasting in delay tolerant networks: a social network perspective. In *Proceedings of ACM MobiHoc* (pp. 299-308). doi:10.1145/1530748.1530790
- Han, J. W., Kamber, M., & Pei, J. (2012). *Data mining: concepts and techniques*. San Francisco, CA: Morgan Kaufmann.
- Hossmann, T., Spyropoulos, T., & Legendre, F. (2009). From contacts to graphs: Pitfalls in using complex network analysis for DTN routing. In *Proceedings of IEEE INFOCOM* (pp. 260-265). doi:10.1109/INFCOMW.2009.5072147

- Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J., & Diot, C. (2005). Pocket switched networks and human mobility in conference environments. In *Proceedings of ACM WDTN* (pp. 244-251). doi:10.1145/1080139.1080142
- Hui P., Crowcroft, J., & Yoneki, E. (2008) Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of ACM MobiHoc* (pp. 241-250). doi:10.1145/1374618.1374652
- Jain, S., Fall, K., & Patra, R. (2004). Routing in a delay tolerant network. In *Proceedings of ACM SIGCOMM* (pp. 145-158). doi:10.1145/1015467.1015484
- Johnson, D. B., & Maltz, D., A. (1996). Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, (pp. 153-181) . Retrieved from [http://www.cs.colorado.edu/~rhan/CSCI_7143_Fall_2007/Papers/Johnson96_DS R.pdf](http://www.cs.colorado.edu/~rhan/CSCI_7143_Fall_2007/Papers/Johnson96_DS_R.pdf)
- Jones, E. P., & Ward, P. A. (2004). Routing strategies for delay-tolerant networks. In *Proceedings of ACM SIGCOMM*. Retrieved from <http://ccng.uwaterloo.ca/~pasward/Publications/dtn-routing-survey.pdf>
- Ko, Y.-B., & Vaidya, N. H. (2000). Location-aided routing (lar) in mobile ad hoc networks. *Wireless Networks*, 6(4), 307-321. doi:10.1023/A:1019106118419
- Lee, U., Oh, S. Y., Lee K.-W., & Gerla, M. (2008). RelayCast: scalable multicast routing in delay tolerant networks. In *Proceedings of IEEE ICNP* (pp. 218-227). doi:10.1109/ICNP.2008.4697040
- Mei, A., Morabito, G., Santi, P., & Stefa, J. (2011). Social-aware stateless forwarding in pocket switched networks. In *Proceedings of IEEE INFOCOM* (pp. 251-255). doi:10.1109/INFCOM.2011.5935076

- Mongiovi, M., Singh, A. K., Yan, X., Zong, B., & Psounis, K. (2012). Efficient multicasting for delay tolerant networks using graph indexing. In *Proceedings of IEEE INFOCOM* (pp. 1386-1394). doi:10.1109/INFOCOM.2012.6195503
- Motani, M., Srinivasan, V., & Nuggehalli, P. (2005). Peoplenet: engineering a wireless virtual social network. In *Proceedings of MOBICOM* (pp. 243-257). doi:10.1145/1080829.1080855
- Pentland, A., Fletcher, R., & Hasson, A. (2004). DakNet: rethinking connectivity in developing nations. *Computer*, 37(1), 78-83. doi: 10.1109/MC.2004.1260729
- Perkins, C. E., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of ACM SIGCOMM* (pp. 234-244). doi:10.1145/190809.190336
- Perkins, C. E., Royer, E. M., & Das, S. R. (2002). Ad hoc on-demand distance vector (AODV) routing, *IETF Internet draft*. Retrieved from <http://www.ietf.org/rfc/rfc3561.txt>
- Pietilainen, A., & Diot, C. (2012). Dissemination in opportunistic social networks: the role of temporal communities. In *Proceedings of ACM MobiHoc* (pp. 165-174). doi: 10.1145/2248371.2248396
- Rothfus, D., Dunning, C., & Chen, X. (2013). Social-similarity-based routing algorithm in delay tolerant networks. In *Proceedings of IEEE ICC* (pp. 1862-1866). doi:10.1109/ICC.2013.6654792
- Scott, J., Crowcroft, P., Hui, P., & Diot, C. (2006). Hagggle: a networking architecture designed around mobile users. In *Proceedings of IFIP WONS*. Retrieved from <http://research.microsoft.com/pubs/132418/wons2006-hagggle.pdf>

- Scott, J., Gass, R., Crowcroft, J., Hui, P., Diot, C., & Chaintreau, A. (2009). *CRAWDAD trace cambridge/haggle/imote/infocom2006* (v.2009-05-29). Retrieved from <http://crawdad.cs.dartmouth.edu/cambridge/haggle/imote/infocom2006>
- Shannon, C. E., Petigara, N., & Seshasai, S. (1948). A mathematical theory of communication. *Bell System Technicl Journal*, 27(1), 379-423. Retrieved from <http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf>
- Srinivasan, V., Motani, M., & Ooi W. (2006). Analysis and implications of student contact patterns derived from campus schedules. In *Proceedings of MOBICOM* (pp. 86-97). doi:10.1145/1161089.1161100
- Vahdat, A., & Becker, D. (2000). *Epidemic routing for partially connected ad hoc networks* (Report No. CS-200006). Durham, NC: Duke University. Retrieved from <http://issg.cs.duke.edu/epidemic/epidemic.pdf>
- Wang, Y., & Wu, J. (2012). A dynamic multicast tree based routing scheme without replication in delay tolerant networks. *Journal of Parallel and Distributed Computing*, 72(3), 424-436. doi:10.1016/j.jpdc.2011.11.007
- Wang, Y., Chen, W. D. (2001). Supporting IP multicast for mobile hosts. *Mobile Network and Applications*, 6(1), 57-66. doi:10.1023/A:1009865804468
- Wu, J., & Wang, Y. (2012). Social feature-based multi-path routing in delay tolerant networks. In *Proceedings of IEEE INFOCOM* (pp. 1368-1376). doi:10.1109/INFCOM.2012.6195500
- Wyatt, J., Burleigh, S., Jones, R., Torgerson, L., & Wissler, S. (2009). Disruption tolerant networking flight validation experiment on NASA's EPOXI mission. In *Proceedings of SPACOMM* (pp. 187-196). doi:10.1109/SPACOMM.2009.39

- Xi, Y., & Chuah, M. (2009). An encounter-based multicast scheme for disruption tolerant networks. *Computer Communications*, 32(16), 1742-1756.
doi:10.1016/j.comcom.2008.09.031
- Yang, S., & Wu, J. (2013). Adaptive backbone-based routing in delay tolerant networks. In *Proceedings of IEEE MASS* (pp. 356-364). doi:10.1109/MASS.2013.10
- Zhao, W., Ammar, M., & Zegura, E. (2005). Muticasting in delay tolerant networks: semantic models and routing algorithms. In *Proceedings of ACM WDTN* (pp. 268-275). doi:10.1145/1080139.1080145