FRIEND CIRCLE AND SOCIAL FEATURE-BASED ROUTING IN OPPORTUNISTIC MOBILE SOCIAL NETWORKS

by

Rahul Narendrasing Girase

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
August 2017

Committee Members:

Xiao Chen, Chair

Guowei Yang

Qijun Gu

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 acknowledgment. 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, Rahul Narendrasing Girase, authorize duplication of this work, in whole or in part, for educational or scholarly purposes only.

DEDICATED

to

my MOTHER, FATHER

and

my FRIEND

ACKNOWLEDGEMENTS

I would like to express my special appreciation and thanks to my advisor Dr. Xiao Chen, department of computer science at Texas State University, you have been a tremendous mentor for me. Your advice on both research as well as on my career has been invaluable. I could not have imagined having a better advisor and mentor for my work you have. She has never been ceasing in her belief in me and always provide clear guidance.

I would like to thank, Dr. Guowei Yang, department of computer science at Texas State University, to be part of my thesis committee.

I would like to thank, Dr. Qijun Gu, department of computer science at Texas State University, to be part of my thesis committee.

Finally, I must express my special thanks to my parents for providing me with unfailing support and continuous encouragement for this work. This accomplishment would not have been possible without them.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

ABSTRACT

Opportunistic Mobile Social Networks (OMSNs), formed by people moving around carrying mobile devices, enhance spontaneous communication among users that opportunistically encounter each other without additional infrastructure. The OMSNs we discuss here are special kind of delay tolerant networks (DTNs) that help enhance spontaneous interaction and communication among the users that opportunistically encounter each other, without additional infrastructure. Most of the existing routing algorithms proposed for the general-purpose DTNs do not consider social characteristics of nodes. A few papers consider static social feature. In this paper, we introduce the concepts of dynamic social feature and its enhancement enhanced dynamic social feature to capture nodes' dynamic contact behavior. Also, we introduce an online social feature which catches nodes' online behavior with other nodes. We propose three novel routing algorithms based on these features. The first one called EDSF uses enhanced dynamic social features for routing. The second one named FC adopts online social features for routing. And the third one FCSF utilizes the combination of both enhanced social features and online social features. The analysis of the algorithms is given by running simulations on real traces on an OMSN to show that our new algorithms outperform in the terms of delivery rate, time latency and number of forwardings.

I. **INTRODUCTION**

Over the past few years, with the proliferation of smartphones, PDAs, etc. Opportunistic Mobile Social Networks (OMSNs), formed by people carrying these mobile devices and moving around, have become more popular. Unlike popular online social networks such as Facebook and LinkedIn. OMSNs are a special kind of delay tolerant networks (DTNs) where the communication takes place on-the-fly by the opportunistic contacts among mobile users in a lightweight mechanism via local wireless bandwidth such as Bluetooth or Wi-Fi without a network infrastructure [9], [12], [20]. Delay Tolerant Network (DTN) is a type of wireless mobile networks that does not guarantee continuous network connectivity. It can appear in the following applications: satellite communication networks [23], village area networks [21], connected vehicle networks [22], and social communication networks [24]. OMSNs follow the same property of DTNs. Due to the time-varying network topology of OMSNs, end-to-end communication path is not guaranteed, which poses special challenges to routing, either unicast or multicast. Nodes in OMSNs can only communicate through a store-carry-forward fashion. When two nodes move within each other's transmission range, they communicate directly and lose contact when they move out of their ranges. 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 routing algorithms proposed for the general-purpose DTNs [1-4] do not consider social characteristics. For example, flooding which spreads message epidemically in the network until it reaches the destination, but this approach results in a large number of message copies leaving the network which consumes a high amount of bandwidth and energy and memory space. Furthermore, under high traffic loads they suffer from severe contention and message drops that can significantly degrade their performance and scalability. Therefore, different approaches are introduced to reduce the overhead such as Spray and Wait Routing [2] which distributes number of copies in the network and waits until it reaches to the destination, it reduces number of forwardings but results in high latency time. There are a few routing algorithms involving social factors [16-18] that take advantage of the fact that people having more similar social features in common tend to meet more often in OMSNs. Social features F_1 , F_2 , . . .,F_i can refer to

Nationality, City, Language, Affiliation, and so on. Each social feature F_i can take multiple values $f_1, f_2, \ldots, f_i$. For example, a social feature F_i can be *Language* and its values can be *English, Spanish*, and so on. Any social feature which is collected offline is called as offline feature. Offline features are generally collected in social conferences or meetings. In previous work [16], we argued that static social features may not always reflect node's dynamic contact behavior. For example, we just consider social features like ⟨City, Affiliation⟩, suppose destination D's social feature values in these two dimensions are ⟨New York, Student⟩, then the vectors of two candidate forwarders A and B who have the same social feature values as D will be set to ⟨1, 1⟩, which makes them indistinguishable.

In this research, we believe that routing can be further improved in two ways: (1) by upgrading the definition of dynamic social features to enhanced dynamic social feature and (2) by considering *online social features* like friendship status from Facebook, LinkedIn. The definition of the dynamic social features in Routing is based on node contact frequency, which can be easily obtained and inexpensive to maintain in OMSNs. It also reflects the intuition that people having more similar social features in common tend to have higher contact frequencies in OMSNs. But it cannot distinguish the cases when two nodes have the same meeting frequency with nodes having a certain social feature. Thus, we upgrade dynamic social features to *enhanced dynamic social features* in order to break the tie. Online social features are collected when the user gets online to any social site and so it is called as online features. Moreover, online social features which consider the friendship relation between each destination and each relay candidate. For example, consider Facebook, if people are friends then they tend to meet often.

Based on the enhanced dynamic social features and the online social features, we propose three novel algorithms: (1) **E**nhanced **D**ynamic **S**ocial **F**eature (EDSF) Algorithm which is based on enhanced dynamic feature. (2) **F**riend **C**ircle (FC)Algorithm which is based on an online social feature. (3) **F**riend **C**ircle &**S**ocial **F**eature which is the combination of both enhanced dynamic social features and online social features. We think that if we combine both features it will further improve routing performance. And then, we conduct simulations using two datasets downloaded from CRAWDAD [25-26] to evaluate performance. We compare our proposed algorithms with existing algorithms viz. Flooding and Spray and Wait. Simulation results show that our new algorithms outperform in terms of delivery rate, latency and number of forwardings. That proves that considering the online social feature and enhanced dynamic social feature both can improve performance of routing

Overview of the Upcoming Chapters

The organization of our current work is as follows:

- In Chapter II we survey relevant studies on Routing solutions in Delay Tolerant Networks and OMSN.
- In Chapter III, we introduce Preliminary
- In Chapter IV, Research Methodology includes algorithms and implementation.
- Chapter V, we present simulation results.
- Chapter VI, consists of the concluding remarks and future work.

II. RELATED RESEARCH

Although a significant amount of work and consensus exists on the general DTN architecture [1], there hasn't been a similar focus and agreement on DTN routing algorithms, especially when it comes to networks with "opportunistic" connectivity. This might be due to the large variety of applications and network characteristics falling under the DTN [2].

Delay Tolerant Networks

The frequently observed DTN is of Routing. Such networks are assumed to experience frequent, long-duration partitioning and may never have an end-to-end contemporaneous path. This problem contrasts with routing in conventional data networks which typically selects a shortest policy-compliant path in a connected graph without considering the availability of intermediate buffering and bandwidth capacity. Sushant Jain, Kevin Fall, Rabin Patra formulated the problem in DTN routing when the connectivity patterns are known, then provided a framework for evaluating various routing algorithms, and finally showed a simulation-based comparison.

Intermittently connected mobile networks don't have a complete path from a source to a destination most of the time. Such an environment can be found in very sparse mobile networks where nodes meet only occasional all over in wireless sensor networks where nodes sleep most of the time to conserve energy. A hybrid probabilistic routing algorithm is proposed (Ze Li and Haiying Shen) using multi-copies called HUM, in which a packet is initially replicated to a certain number of nodes, which sequentially forwards those packets to the destination node based on a probabilistic routing scheme [16].

Normally, one of the most basic requirements for enabling two nodes to communicate through a network is that there exists a fully connected path between them. However, there are scenarios where this is not the case, but where it still would be desirable to allow communication between nodes. Anders Lindgren, AvriDoria, OlovSchelen present a probabilistic protocol for routing in intermittently connected networks [12]. In such networks, there is no guarantee that a fully connected path between source and destination exist at any time, rendering traditional routing protocols unable to deliver messages between hosts. A probabilistic metric called delivery predictability is established at each node for each known destination indicating the predicted chance of that node delivering a message to that destination. When a node encounters another node, they exchange information about the delivery predictabilities they have and update their own information accordingly. Based on the delivery predictabilities, a decision is then made on whether or not to forward a certain message to this node.

Opportunistic Social Mobile Network

Social features are considered for message forwarding. Existing routing algorithms for Delay Tolerant Networks (DTNs) assume that nodes are willing to forward packets for others. In the real world, however, most people are *socially selfish*; i.e., they are willing to forward packets for nodes with whom they have social ties but not others, and such willingness varies with the strength of the social tie. One of the challenge is considering the accurate node for message forwarding, which can be achieved by one of the algorithms (IEEE INFOCOM 2010, Qinghua Li, Sencun C, Guohong Cao) viz. Social Selfishness Aware Routing(SSAR) algorithm to allow user selfishness and provide better routing performance in an efficient way. To select a forwarding node, SSAR considers both users' willingness to forward and their contact opportunity, resulting in a better forwarding strategy than purely contact-based approaches.

In some scenarios, conventional routing schemes fail as there exists no complete path from source to destination. These conventional methods try to establish complete end-to-end paths before any data is sent. To deal with such networks researchers have suggested to use flooding-based routing schemes. While flooding-based schemes have a high probability of delivery, they waste a lot of energy and suffer from severe contention which can significantly degrade their performance. Working on this (Thrasyvoulos, Konstantinos and Cauligi) researchers came across "single-copy" routing schemes that use only one copy per message, and hence significantly reduce the resource requirements of flooding-based algorithms.

Mobile Social Network It is a special kind of delay tolerant network (DTN) composed of mobile nodes that move around and share information with each other through their carried short-distance wireless communication devices. A main characteristic of MSNs is that mobile nodes in the networks generally visit some locations (namely, community homes) frequently while visiting other locations less frequently. To Overcome this, A novel zero-knowledge multi-copy routing algorithm was introduced (Mingjun, Jie, Liusheng, IEEE2014), homing spread (HS), for homogeneous MSNs, in which all mobile nodes share all community homes. HS is a distributed and localized algorithm [10]. It mainly lets community homes spread messages with a higher priority. By using homes to spread messages faster, HS achieves a better performance than existing zero-knowledge MSN routing algorithms, including Epidemic (with a given number of copies), and Spray & Wait [2].

Intermittently Connected Mobile Networks It is sparse wireless networks where most of the time there does not exist a complete path from the source to the destination. These networks fall into the general category of Delay Tolerant Networks. There are many real networks that follow this paradigm, for example, wildlife tracking sensor networks, military networks, inter-planetary networks, etc. In this context, conventional routing schemes would fail. To deal with such networks researchers have suggested to use flooding-based routing schemes. While flooding-based schemes have a high probability of delivery, they waste a lot of energy and suffer from severe contention, which can significantly degrade their performance. Furthermore, proposed efforts to significantly reduce the overhead of flooding-based schemes have often be plagued by large delays. With keeping this in mind, we introduce a new routing scheme, called Spray and Wait [2], that "sprays" several copies into the network, and then "waits" till one of these nodes meets the destination. Spray and Wait combines the speed of epidemic routing with the simplicity and thriftiness of direct transmission. It initially "jump-starts" spreading message copies in a manner like epidemic routing. When enough copies have been spread to guarantee that at least one of them will find the destination quickly (with high probability), it stops and lets each node carrying a copy perform direct transmission. In other words, Spray and Wait could be viewed as a tradeoff between single and multi-copy schemes. Surprisingly, as we shall shortly see, its performance is better with respect to both number of transmissions and delay than all other practical single and multi-copy schemes, in most scenarios considered.

III. PRELIMINARY

In this section, we present the concepts of friend circle feature, dynamic social features, dynamic social features and enhanced dynamic features to prepare for later proposed routing algorithms.

1) **Delay Tolerant Network**

   In recent years, delay tolerant networks (DTNs), such as space communication and networking in sparsely populated areas, vehicular ad hoc networks, have been subject to extensive research efforts. Unlike traditional tethered networks like the Internet, a DTN is a sparse mobile network where the connection between nodes in the network changes over time, and as a result the communication constantly suffers from higher delays and disconnections. Since a contemporaneous end-to-end path may never exist in DTNs, effective communication in DTNs requires cooperation of all the nodes for routing and forwarding, where, the intermediate nodes on a communication path are expected to store, carry and forward the packets in an opportunistic way, which is also named as opportunistic data forwarding. However, in most cases, DTNs could consist of many resource constrained nodes, i.e. limited storage. If carried for a certain extent of time without an available downstream node, the packets must be dropped by the carrying node, which thus incurs very unreliable forwarding in DTNs. Therefore, efficient packet forwarding in DTNs becomes an especially challenging issue and several DTN packet forwarding scheme recently has been proposed to improve reliability [11]. DTN can be described abstractly using a graph. Each edge in this graph represents a contact. If there is no contact with another host, the message to be delivered needs to be stored in the local buffer of the current host until a connection comes again.

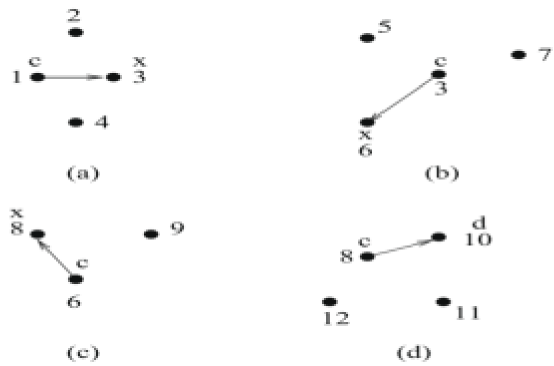

Figure 1 An example of routing process in DTNs.

2) **Opportunistic Social Mobile Network**

In recent years, opportunistic mobile social networks emerged as a new mechanism of communications in wireless networks. Unlike mobile ad hoc networks (MANETs) that require end-to-end communication paths for message exchange, the communication in opportunistic mobile social networks take place on the establishment of opportunistic contacts among mobile nodes, without the availability of end-to-end message routing paths. As the mobile devices can make contact only when humans come into contact, such networks are tightly coupled with human social networks. Therefore, the opportunistic mobile social networks exploit the human behaviors and social relationships to build more efficient and trustworthy message dissemination schemes [19].

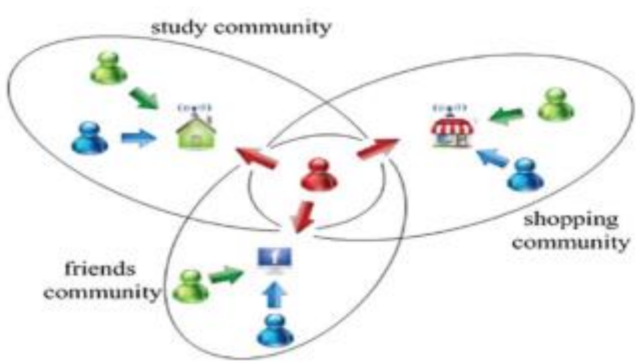

Figure 2 Mobile Social Network

3) **Static Social Features**

Suppose we consider n social features < f_1,f_2 …. f_n> in the network. We associate a node with a vector of static social features values <f_1, f_2 …. f_n> obtained from the user profile [7]. For example, if x's static social feature vector is <Facebook, New York, English>and y's static social feature vector is <Facebook, California, English>, then they will have 2 social features values Facebook and English in common out of 4 total unique social feature values Facebook, New York, Texas and English.

Therefore, their social similarity S (x, y) is $\frac{2}{4}$= 0.5.

4) **Dynamic social features**

Suppose we consider node X and node X has met N total number of nodes. By considering the previous history we are forming a group of meeting node with X.

$$X_i = \frac{N_i}{N_{total}} \quad (1)$$

Here, N_i is the number of meeting of node X with nodes having same social feature value fi and N_{total} is the total number of nodes X has met in the previous history. Thus, we can get a group of nodes which met with node x which will give us accurate information [16].

5) **Enhanced dynamic social features**

The above frequency-based dynamic social features cannot distinguish the case, for example, if A has met 1 Student out of 2 people it has met and B has met 5 Students out of 10 people it has X met in the history we observe. Both have the same frequency of 1/2 to meet a Student, but B is more active in meeting people. To favor the more active node, there are many ways to do it. Here, we come up with Definition 2 for the enhanced dynamic social features which will be proved to satisfy our needs in the later analysis section. The x_i ($0 \leq x_i \leq 1$) in node x's enhanced dynamic social features

X=(x_1,x_2,···,xm) is defined as follows:

$$X_i = \left(\frac{N_i+1}{\mathrm{N_{total}} + 1}\right)^{P_i} * \left(\frac{N_i}{\mathrm{N_{total}}+1}\right)^{1-P_i} \quad (2)$$

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

6) **Friend Circle**

In general Friend circle is a group of friends. It is formed from several friends getting together. We can form friend circle by using different ways like considering their static feature similarity, a dynamic feature from their previous meeting history or their friendship status using online features like Facebook, Twitter, LinkedIn.

IV. RESEARCH METHODOLOGY

In this section, we propose three routing algorithms using friend circle and enhanced dynamic social feature based on friend circle detection.

1) **Enhance Dynamic Social Feature (EDSF)Algorithm**

   Our first algorithm uses enhance dynamic social feature to make routing decisions. Details of the algorithm are in fig. 4. Its basic idea is as follows: First, a source node s has to send a message to destination and s is initial message holder on relay node x. When x meets a node y, if y is destination then y gets the message. Next, as shown in fig. 3, at the beginning of process every node will try to reach the destination. Then each node will try to reach a destination with another node till it reaches to the destination. At each step, two nodes with highest probability will form a path. We will compare the paths and check whether it is better to pass the message to y or not. Both x and y are called relay candidates in the decision. In this algorithm, we use distance matrix to derive all possible paths between source and the destination. Then it will choose the best path with high probability. For example, X is message holder and it has two paths to reach destination X->X1->D and X->X3->D, if X1 probability to reach destination is greater than X3 then first path will be chosen and X will deliver message to X1.

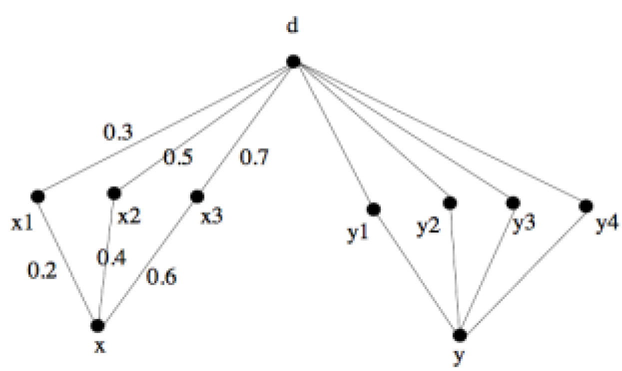

Figure 3 example for path finder

a) Distance Matrix

From the enhance dynamic social features, we can create a distance matrix as shown in Fig.5 to indicate the social difference or distance between each node. The social distance between two nodes X_i and X_j or Y_i and Y_j is defined as 1-S(X_i, Xj) or 1-(Y_i,Y_j). . The distance matrix will be used in the friend circle detection algorithm to separate nodes into two friend circle.

Enhanced Dynamic Social Feature (EDSF) Algorithm:

Require: The source node S and Destination Node D, S is the initial message holder and copy of message C=n;

1. **While** destination not received, the message **do**
2. On contact between message holder x and next node y
3. **If** y ∈ D **then**
   /* found destination y */
   deliver message to y.
   End IF
4. construct distance matrix.
5. Derive all possible path considering source and destination from distance matrix
6. Perform comparison between all the paths.
7. Select the path with higher probability and c=n-1;
8. **END While**

Figure 4 Our EDSF Algorithm for Routing

	X_1	X_2	X_3	Y_1	Y_2	....
X_1	0	1-S(X_1, X_2)	1-S(X_1, X_3)	1-S(X_1, Y_1)	1-S(X_1, Y_2)	
X_2	1-S(X_2, X_1)	0	1-S(X_2, X_3)	1-S(X_2, Y_1)	1-S(X_2, Y_2)	
X_3	1-S(X_3, X_1)	1-S(X_3, X_2)	0	1-S(X_3, Y_1)	1-S(X_3, Y_2)	
Y_1	1-S(Y_1, X_1)	1-S(Y_1, X_2)	1-S(Y_1, X_3)	0	1-S(Y_1, Y_2)	
Y_2	1-S(Y_2, X_1)	1-S(Y_2, X_2)	1-S(Y_2, X_2)	1-S(Y_2, Y_1)	0	
....						

Figure 5 Distance Matrix. The distance between each node from other nodes

1) Friend Circle (FC)Algorithm

Our Second friend circle feature algorithm is named as FC Algorithm shown Fig.5. It's basic idea as follows: First, we will have Source Node S (Initial message holder), destination node D, x is relay node and C is number of copies of messages. When x meets a node y, if y is destination D, y will get the message. Next, if y is not a destination then, we will generate distance matrix to check whether it is better to pass message to another node or not. Both x and y will be considered as relay nodes. To separate x friend circle and y friend circle we use friend circle detection algorithm based on their social friendship status. Friend circle detection will take distance matrix as an input.

a) Friend Circle Detection Algorithm

Here we used K-means clustering algorithm to split nodes into two friend circle. We choose this because it matches our needs and there is an existing Java package available for this algorithm so that we don't have to reinvent the algorithm. The idea of k-means friend circle detection algorithm is as follows: At the beginning of the process, each node will represent its own friend circle. Then friend circles will combine into larger friend circle until all nodes end up being in one friend circle. At each step friend circle with shortest distance are combined. The distance between friend circle is defined as distance between nodes that are farthest away from each other. We feed distance matrix and value of K is 2 to package and obtain two friend circles as a result.

Friend Circle (FC) Algorithm:

Require: The source node S and Destination Node D, S is the initial message holder and copy of message C=n;

1. **While** destination not received, the message **do**
2. On contact between message holder x and next node y
3. IF y ∈ D **then**

   /* found destination y */

   deliver message to y

   End IF
4. Construct distance matrix.
5. Create F_1 and F_2 from friend circle detection algorithm.
6. Compare the social similarity of F_1 and F_2 with x and y resp.
7. Select the Friend circle with high probability and c=n-1.
8. **END While**

Figure 6 Our FC Algorithm for Routing

2) Friend Circle Social Feature (FCSF)Algorithm

This algorithm is the combination of both EDSF Algorithm and FC Algorithm. First, it will use same distance matrix which we create in FC Algorithm using friend circle feature and will form friend circle using friend circle detection algorithm. After splitting nodes into two friend circles, we choose best friend circle with higher probability. From selected friend circle, we create another distance matrix by considering enhanced dynamic social features. From distance matrix, it derives all possible paths between source and destination. From all derived paths, it will choose a path with the highest probability. In this algorithm by combining both algorithms features, we hope to improve the accuracy of the algorithm to get better results.

FCSF Algorithm:

Require: The source node S and Destination Node D, S is the initial message holder and copy of message C=n;

1. **While** destination not received, the message **do**
2. On contact between message holder x and next node y
3. **IF** y ∈ D **then**

   /* found destination y */

   deliver message to y.

   End IF
4. Generate distance matrix.
5. Create F_1 and F_2 from friend circle detection algorithm.
6. Compare the social similarity of F_1 and F_2 with x and y respectively.
7. Select the Friend circle with high probability.
8. construct distance matrix from the selected friend circle.
9. Derive all possible path considering source and destination from distance matrix
10. Perform comparison between all the paths.
11. Select the path with higher probability and c=n-1.
12. **END While**

Figure 7 Our FCSF Algorithm for Routing

V. SIMULATIONS

In this section, we evaluate the performance of our algorithms by comparing them with existing algorithms like flooding algorithm and spray and wait algorithm using custom simulator written in Java. The simulations were conducted using a real conference dataset first reflecting an OMSN created at IEEE Infocom 2006 in Miami and second data set Socialblueconn 2015.

Details of two datasets as Follows:

Dataset 1: The Infocom 2006 dataset

The Infocom 2006 trace has been widely used to test routing algorithms in mobile social networks. The trace recorded conference attenders' encounter history using Bluetooth small devices (iMotes) for four days at the conference. The trace dataset consists of two parts: contacts between iMote devices that were carried by participants and self-reported social features of the participants collected using a questionnaire form. The six social features extracted from the dataset were *Affiliation, City, Nationality, Language, Country and Position*.

Dataset 2: The Socialblueconn 2015 dataset

This trace collects Bluetooth encounter records of 15 students on the University of Calabria campus in Italy using an ad-hoc Android application called Socialblueconn. The trace dataset consists of the contacts between Bluetooth devices carried by the participating students and their social profiles including Facebook friends and self-declared interests. There are 9 interest categories labeled from A to I representing their preferred transportation methods sport, music, cinema, literature, multimedia entertainment, politics, social network. We used the student's self-declared interests as their social features.

1) **Comparison with existing algorithm**

   We compared following routing protocols.

   a) Flooding Algorithm: The message is spread epidemically across the network until it reaches the destination.

   b) Spray and wait Algorithm: Algorithm sprays several copies into the network and then waits till one of those messages gets delivered to the destination.

c) Enhanced Dynamic Social Feature (EDSF) Algorithm: Our first algorithm proposed in this paper uses enhance dynamic social feature to deliver the message to destination.

d) Friend Circle (FC) Algorithm: Our second algorithm proposed in this paper uses online features i.e. friendship status from social sites to deliver the message to destination.

e) FCSF Algorithm: Our third algorithm proposed in this paper uses enhance dynamic social feature as well as an online social feature to deliver the message to the destination.

2) **Evaluation metrics**

We used three important metrics to evaluate the performance of the routing algorithms. We define a successful routing as the one that successfully delivers the message to the destination. The three metrics are:

a) **Delivery Ratio**: The ratio of successfully delivered messages to the number of generated messages.

b) **Delivery Latency**: The average time between the message generation and arrival time at the destination

c) **Number of forwardings**: The number of messages needed to get delivered to the destination.

3) **Simulation Setup**

The simulation is carried out using 2 datasets for each algorithm. In experiment 1-3 dataset 1is used whereas in experiment 4-6 dataset 2 is used. We tried with 2, 5 and 10 copies of messages. In each experiment, we randomly generated the source and destination.

4) **Simulations Results**

a) Experiment 1

In this experiment, we used dataset 1 as input to our algorithm. And set number of copies equal to 2 to see how our algorithms perform as compared to existing algorithms in the terms of Time Latency, Number of Forwardings and Delivery ratio.

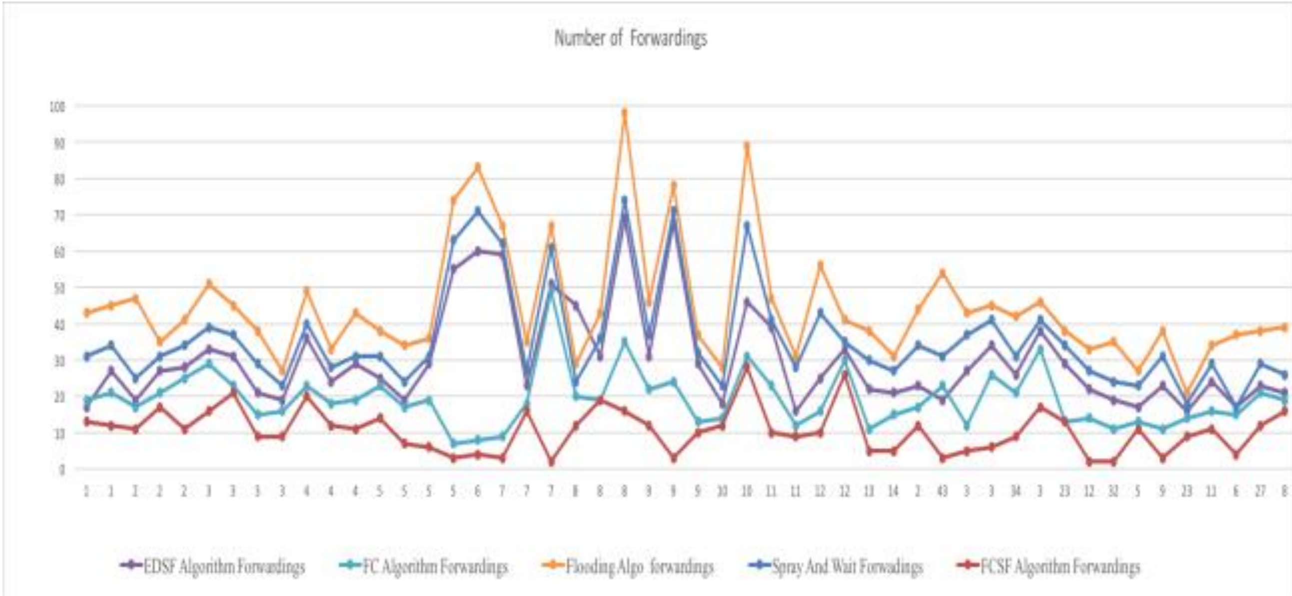

Figure 8 Dataset1 Experiment 1 Comparison on basis of Number of Forwardings

Number of Forwardings:

- As shown in figure 8, Flooding algorithm has the highest number of forwardings as it spreads the messages across the network which increases the number of forwardings for the message.
- Spray and wait algorithm slightly performs better than flooding algorithm, it sends messages to certain nodes in the network and waits for message to get delivered.
- EDSF Algorithm uses enhanced dynamic features for decision making in best node finding and it performs better than flooding and spray and wait algorithm.
- FC Algorithm uses friend circle feature for decision making in best node finding and as shown in the figure it performs better than all three algorithms.
- As we can clearly see in the figure 8 that FCSF algorithm clearly outperforms in number of forwardings. As it uses combination of enhanced dynamic social features and friend circle feature. It clearly gives a better idea about the best node in the network which can carry the message and it helps to improve number of forwardings.

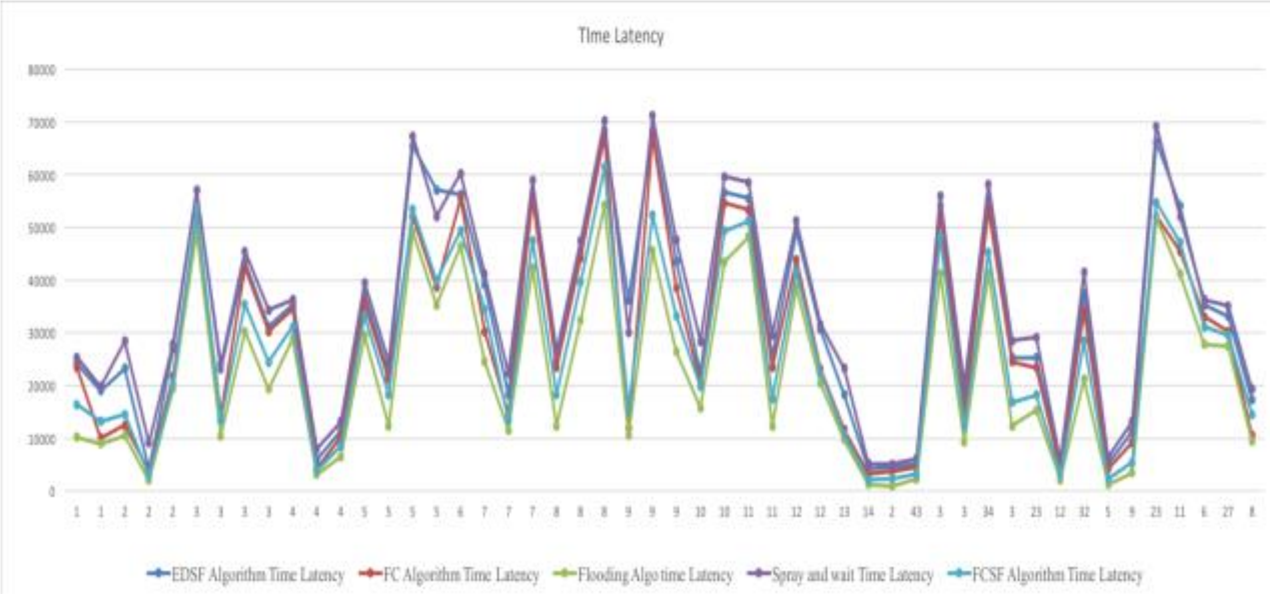

Figure 9 Dataset1Experiment 1 Comparison on basis of Time Latency

Time Latency:

- As shown in figure 9, as expected we can see that the flooding algorithm has less time latency as compared to all existing algorithms, because it spreads the message across the network which improves possibility to meet destination very sooner.
- Spray and wait algorithm has greater time latency as compared to all our algorithms because of its spray and wait nature.
- EDSF Algorithm and FC Algorithm performs nearly same in time latency.
- FCSF algorithm performs better than spray and wait, EDSF Algorithm and FC Algorithm because it has a limited number of nodes to travel so it always tries to get best node network.

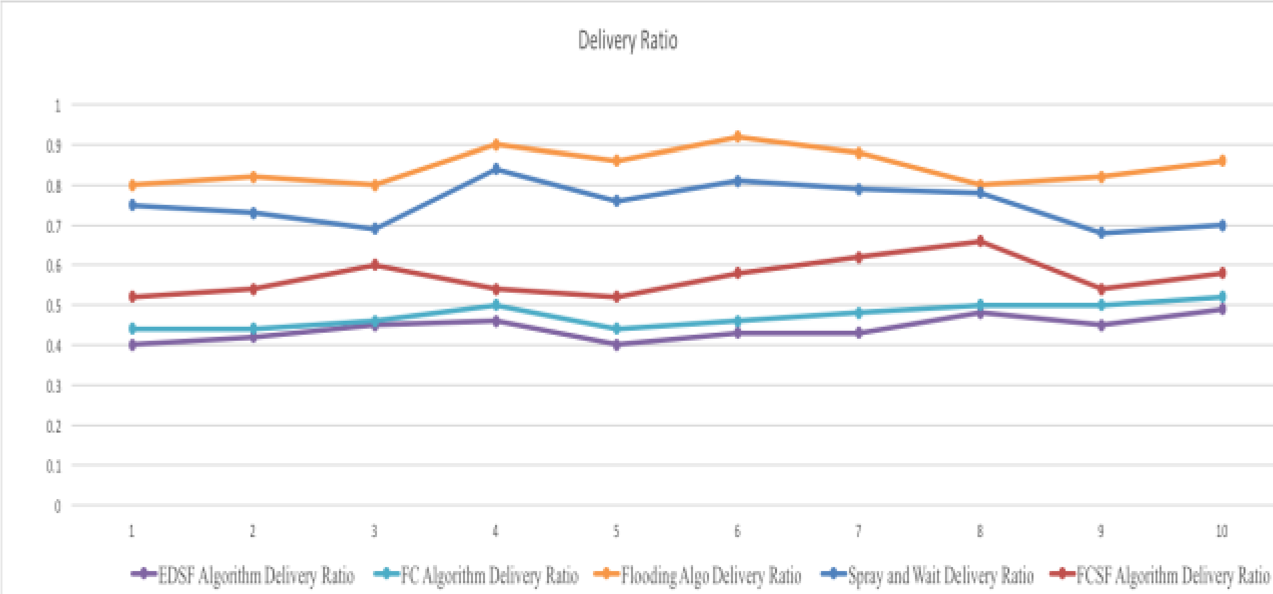

Figure 10 Dataset1Experiment 1 Comparison on basis of Delivery Ratio

Delivery Ratio:

- In figure 10, As expected Flooding algorithm will always have a better delivery ratio as compared to all existing algorithms.
- Spray and wait perform better in this case and has more delivery ratio as compared to the algorithm.
- EDSF Algorithm and FC Algorithm have very poor ratio as compared to flooding and spray and wait algorithm for less number of copies.
- FCSF algorithm performs better than the EDSF Algorithm and FC Algorithm by combining their enhanced dynamic social features and friend circle feature.

b) Experiment 2

In this experiment, we increased the number of copies from 2 to 5 to see how it affects the results of our algorithms compared to existing algorithms in terms of Time Latency, Number of Forwardings and Delivery ratio.

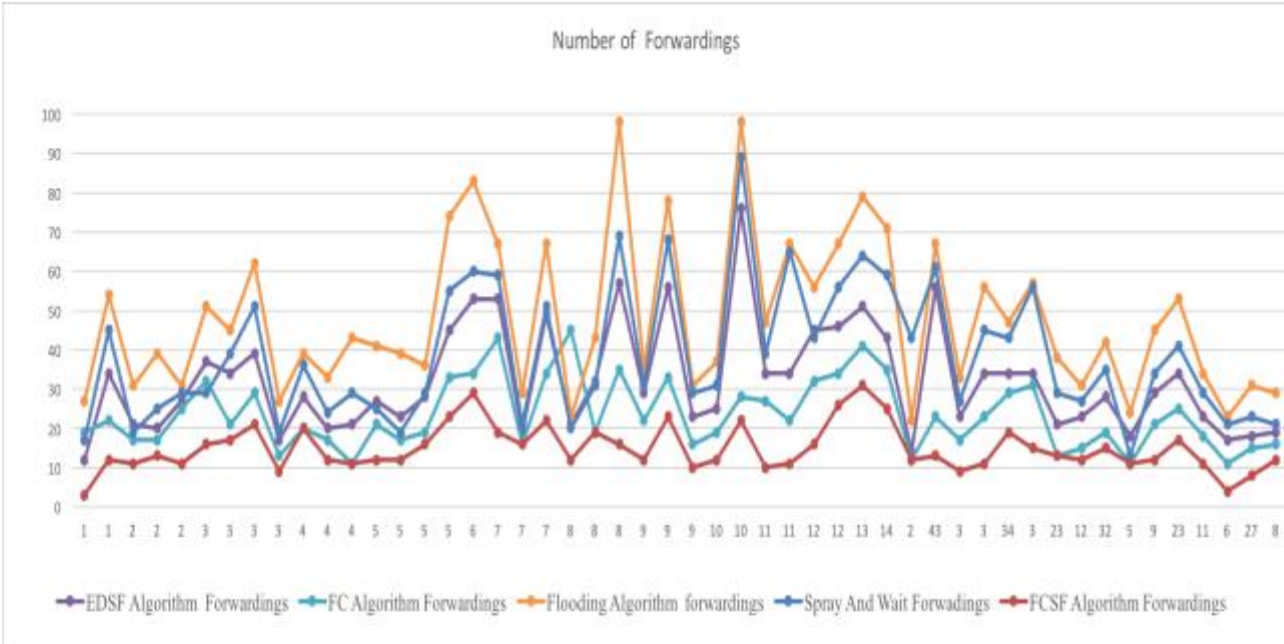

Figure 11 Dataset1Experiment 2 Comparison on basis of Number of Forwardings

Number of Forwardings:

As shown in figure 11, we can see the same results as we increase the copies to 5. FCSF algorithm outperforms clearly in Number of forwardings.

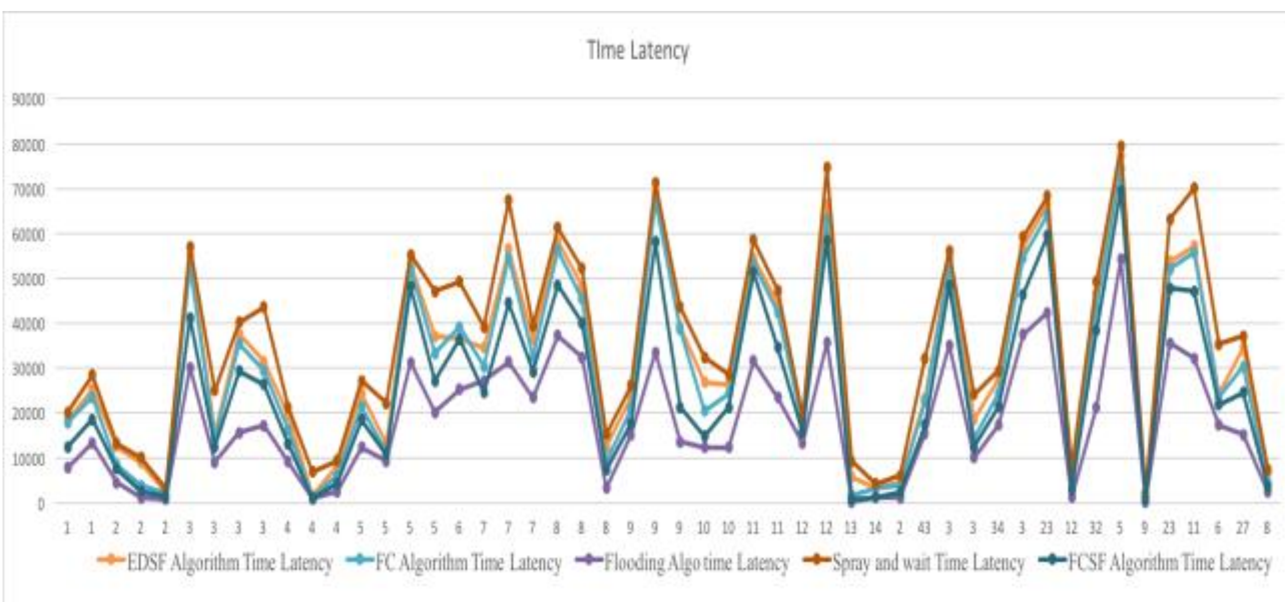

Figure 12 Dataset1 Experiment 2 Comparison on basis of Time Latency

Time Latency
As shown in figure 12, we can clearly see same results which were generated previously in Experiment 1.

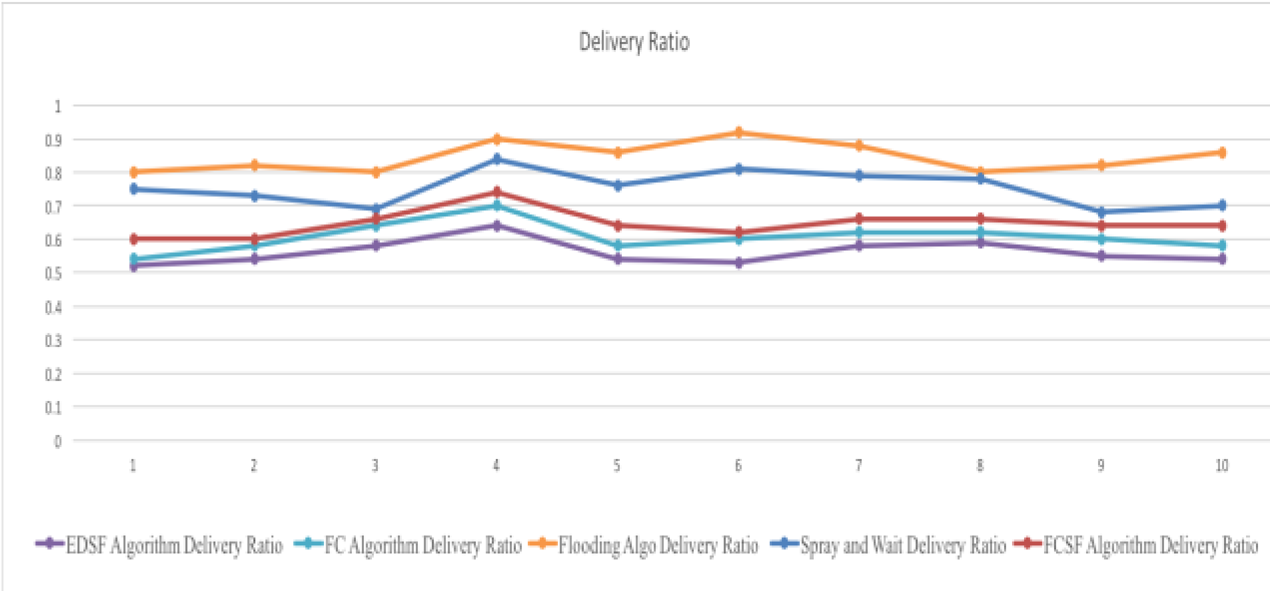

Figure 13 Dataset1Experiment 2 Comparison on basis of Delivery Ratio

Delivery Ratio:

In figure 13, As we increase the number of copies in the network it slightly improves the performance of our EDSF algorithm, FC algorithm and FCSF algorithm. This can be seen by comparing it with Experiment 1.

c) Experiment 3

In this experiment, we now increased the number of copies from 5 to 10 to see how it affects the results of our algorithms as compared to existing algorithms in the terms of Time Latency, Number of Forwardings and Delivery ratio.

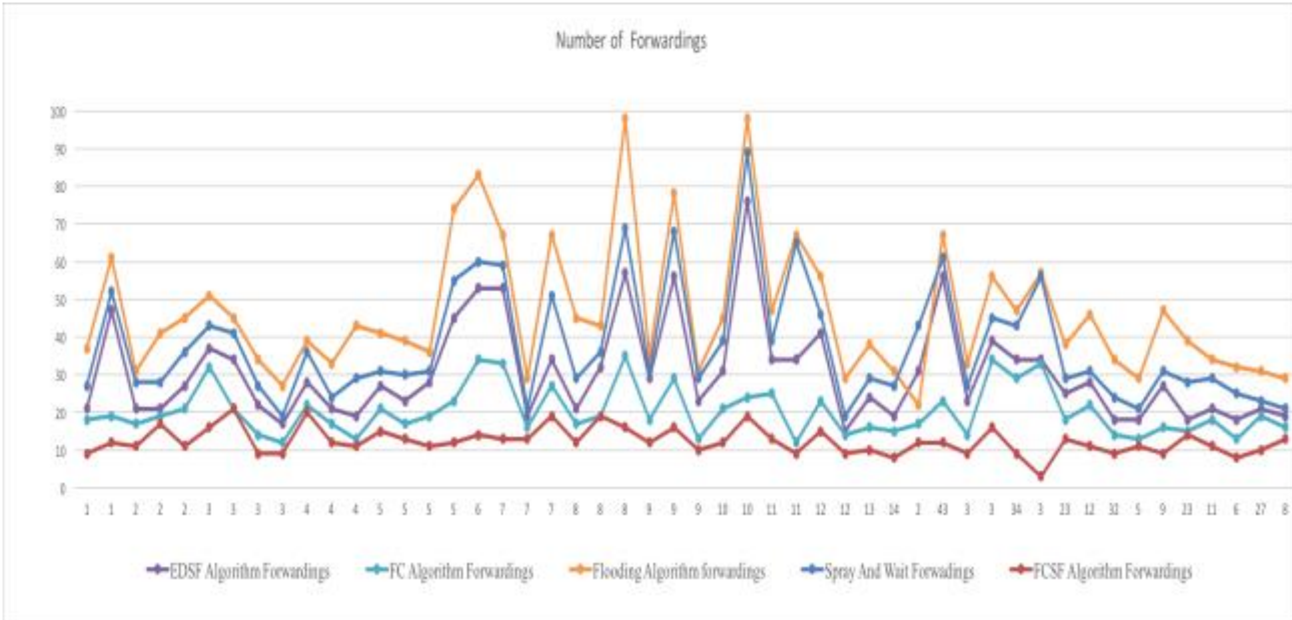

Figure 14 Dataset1 Experiment 3 Comparison on basis of Number of Forwardings

Number of Forwardings:
As shown in figure 14, we can clearly see the same results as that of generated previously in Experiment 1 and Experiment 2.

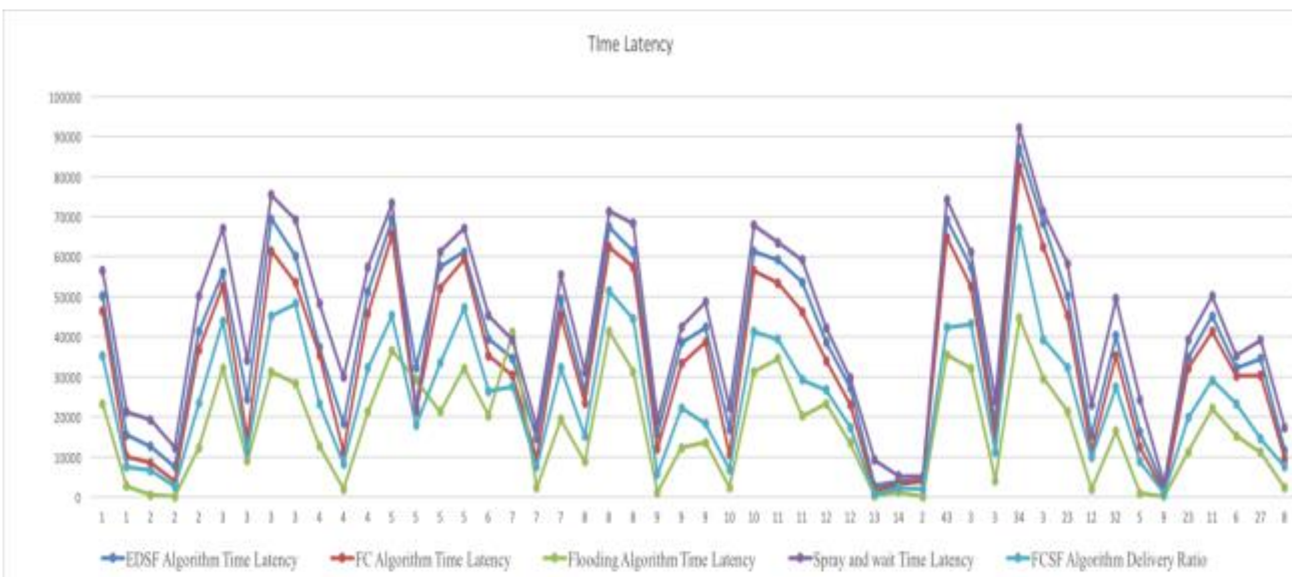

Figure 15Dataset1 Experiment 3 Comparison on basis of Time Latency

Time Latency:
As shown in figure 15, we can clearly see that same results are generated as were generated previously in Experiment 1 and Experiment 2.

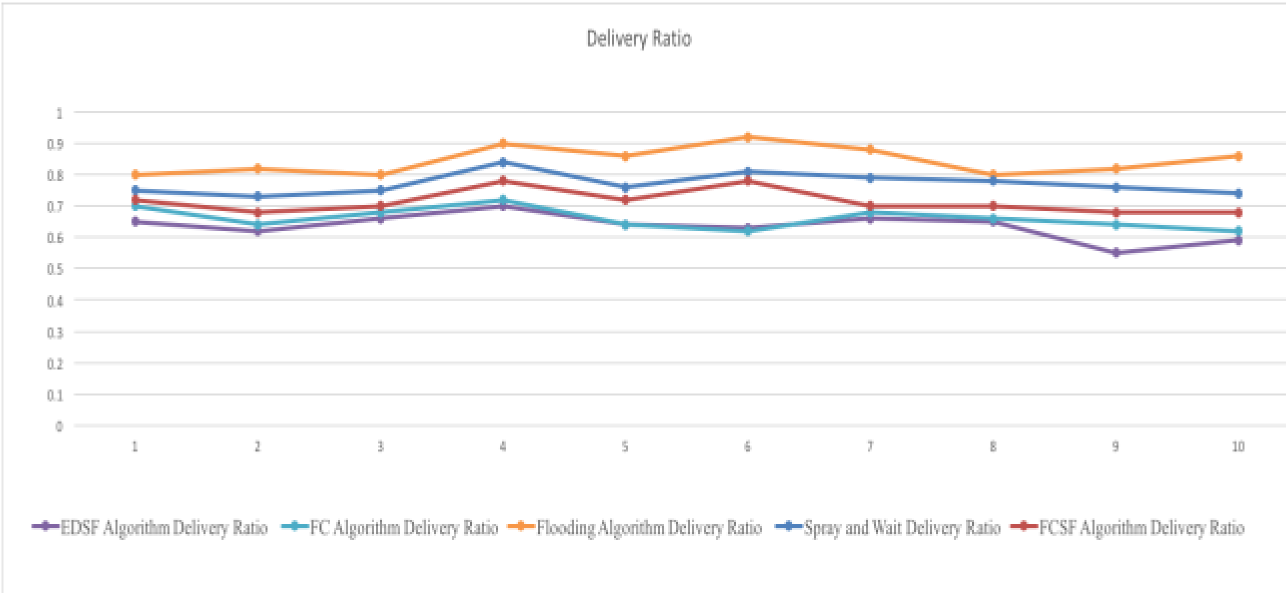

Figure 16 Dataset1 Experiment 3 Comparison on basis of Delivery Ratio

Delivery Ratio:

As shown in figure 16, As we increase the number of copies in the network it slightly improves the performance of our EDSF Algorithm, FC Algorithm and FCSF algorithm this can be examined by comparing it with Experiment 1 and Experiment 2.

Conclusion for Dataset 1:
From Experiment 1, 2 and 3, FCSF algorithm consistently outperforms in terms of number of forwardings and showed continuous increase in delivery ratio. This means, by adding enhanced dynamic social feature and friend circle we can improve routing performance and give very substantiate outcome for message forwarding. In this way, it can be verified that by using enhanced dynamic social feature, routing performance can be improved by considering source and destination of friend circle feature and social feature.

d) Experiment 4

In this experiment, we used dataset 2 as input to our algorithms. And set number of copies equal to 2 to see how our algorithms perform as compared to existing algorithms in the terms of Time Latency, Number of Forwardings and Delivery ratio.

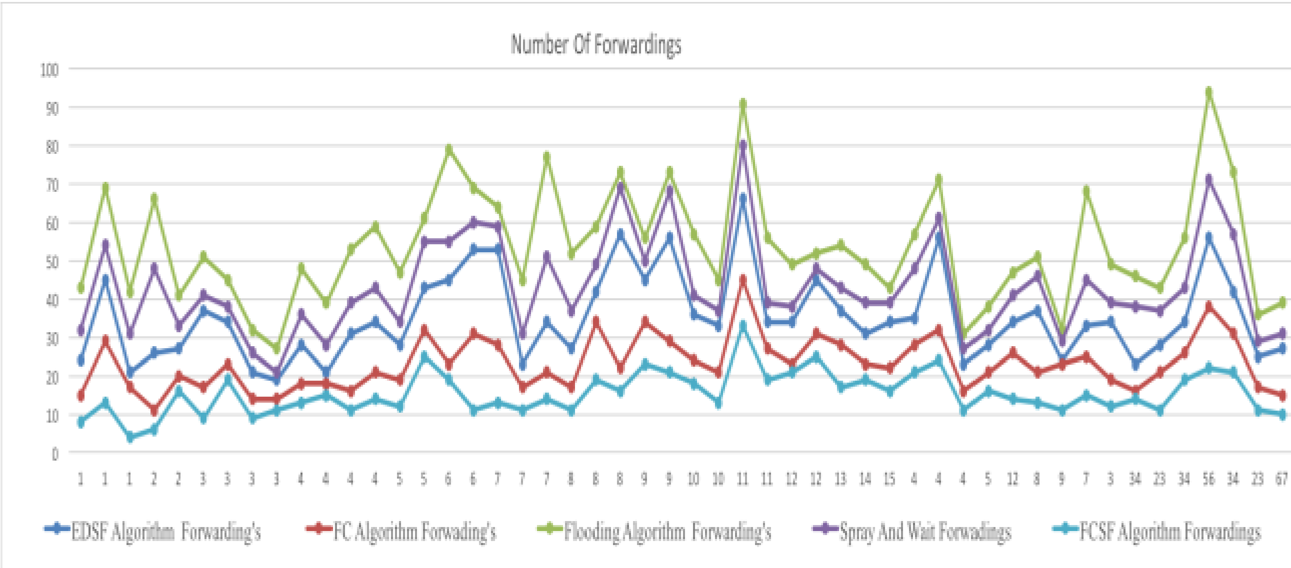

Figure 17 Dataset2 Experiment 4 Comparison on basis of Forwardings

Number of Forwardings:

- As shown in figure 17, Flooding algorithm has the highest number of forwardings as it spreads the messages across the network which increases the number of forwardings for the message.
- Spray and wait algorithm is slightly better than flooding algorithm as it sends messages to certain nodes in the network and waits for message to get delivered.
- EDSF Algorithm uses enhanced dynamic features for decision making in best node finding and it performs better than flooding and spray and wait algorithm.
- FC Algorithm uses friend circle feature for decision making in best node finding and as shown in figure it performs better than all three algorithms.
- As we can clearly see in the figure 17 that our FCSF algorithm clearly outperforms in number of forwardings. It uses a combination of enhanced dynamic social features and friend circle feature. It clearly gives a better idea about the best node in the network which can carry message and it helps to improve number of forwardings.

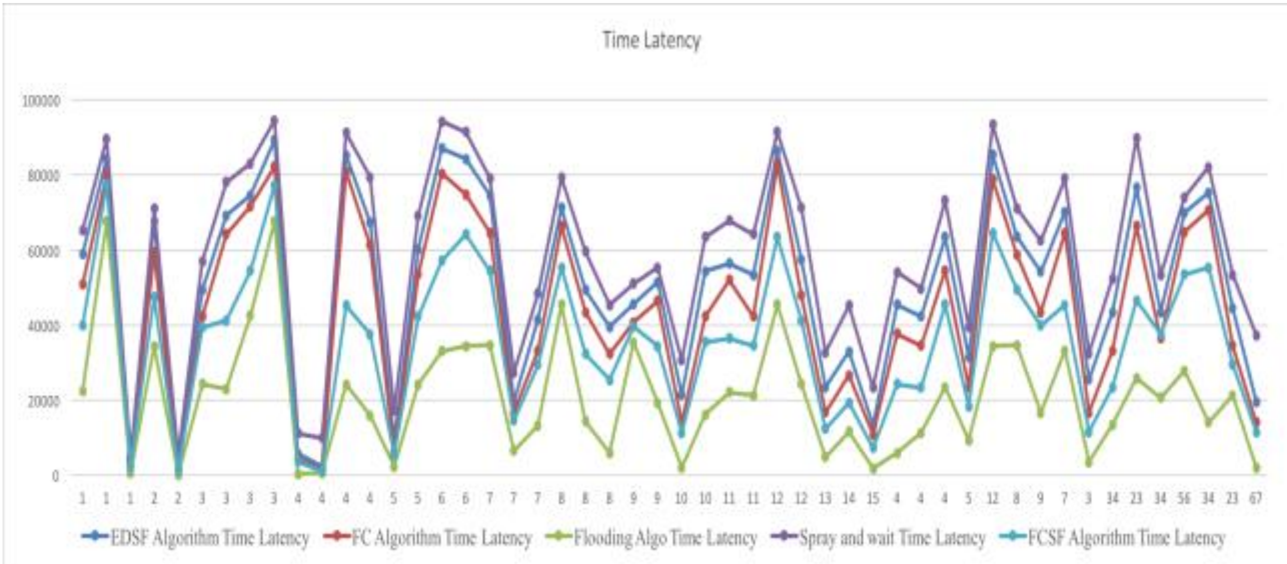

Figure 18 Dataset2 Experiment 4 Comparison on basis of Time Latency

Time Latency:

- As shown in figure 18, as expected we can see that the flooding algorithm has less time latency as compared to all existing algorithms, because it spreads the message across the network which improves the possibility to meet destination very sooner.
- Spray and wait algorithm has greater time latency compare to all our algorithms because of its spray and wait nature.
- EDSF Algorithm and FC Algorithm performs nearly same in time latency.
- FCSF algorithm performs better than spray and wait, EDSF Algorithm and FC Algorithm because it has limited number of nodes to travel so it always tries to get best node network and because of it performs better.

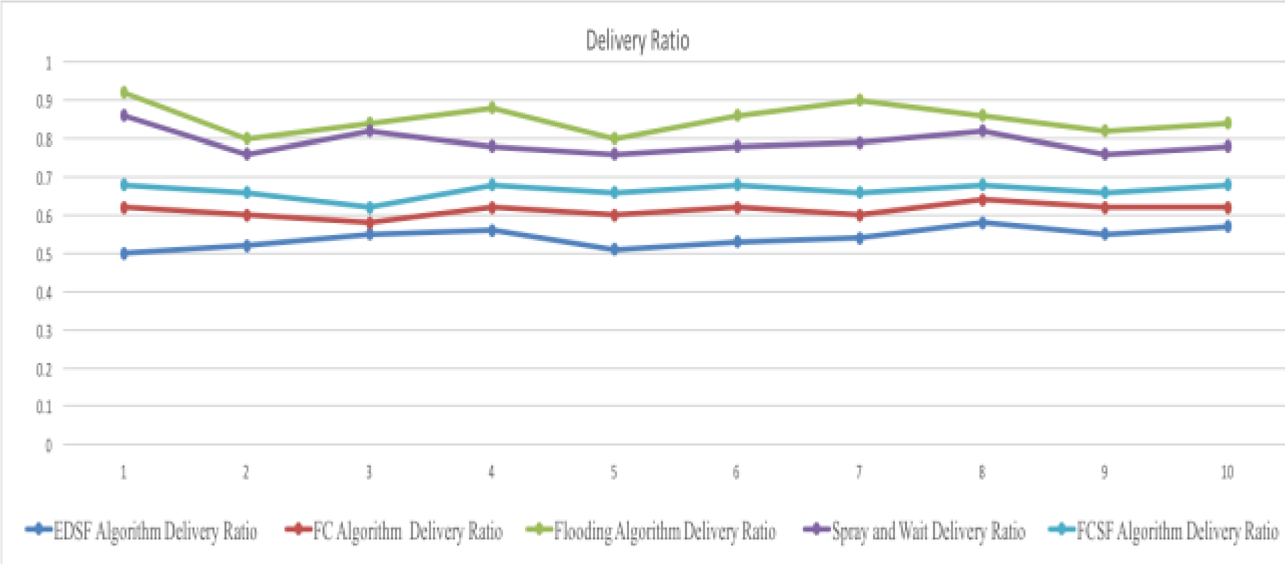

Figure 19 Dataset2 Experiment 4Comparison on basis of Delivery Ratio

Delivery Ratio:

- As shown in figure 19, As expected Flooding algorithm will always have better delivery ratio as compared to all existing algorithms.
- Spray and wait perform better in this case and has more delivery ratio as compared to the algorithm.
- EDSF Algorithm and FC Algorithm has very poor ratio as compared to flooding and spray and wait algorithm for less number of copies.
- FCSF algorithm performs better than the EDSF Algorithm and FC Algorithm by combining their static and dynamic features.

e) Experiment 5

In this experiment, we increased the number of copies from 2 to 5 to see how it affects the results of our algorithms as compared to existing algorithms in the terms of Time Latency, Number of Forwardings and Delivery ratio.

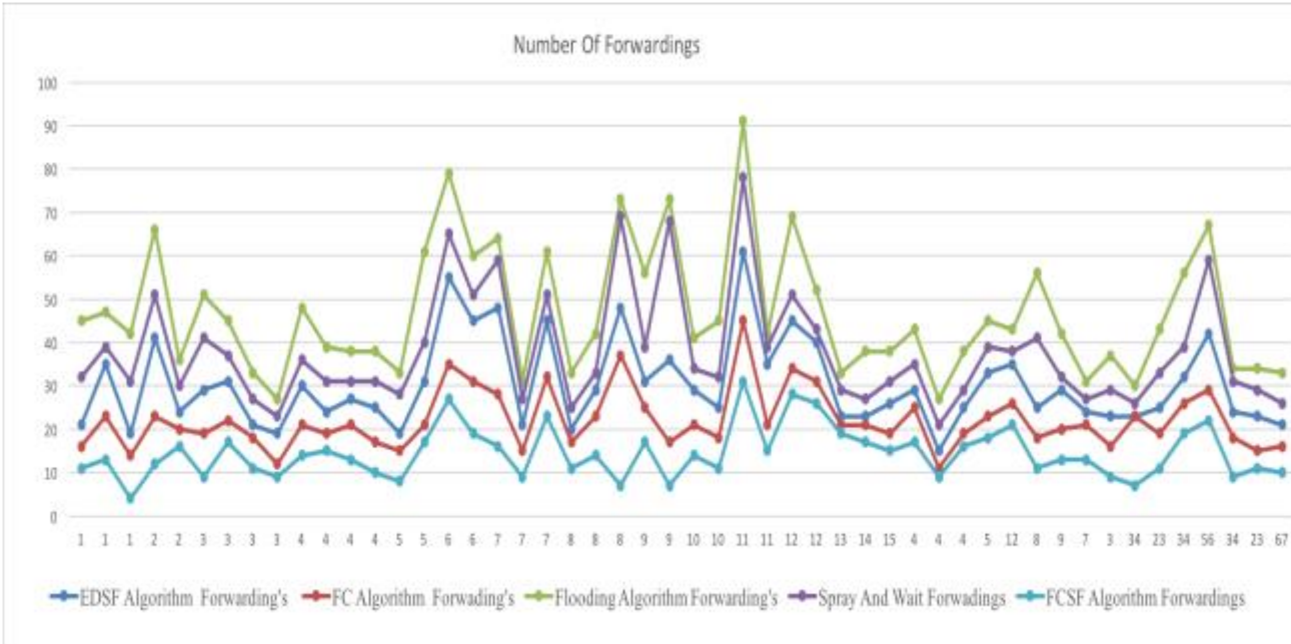

Figure 20 Dataset2 Experiment 5Comparison on basis of Forwardings

Number of Forwardings:

As shown in figure 20, we can see the same results as we increase the copies to 5. FCSF algorithm always outperforms in number of forwardings.

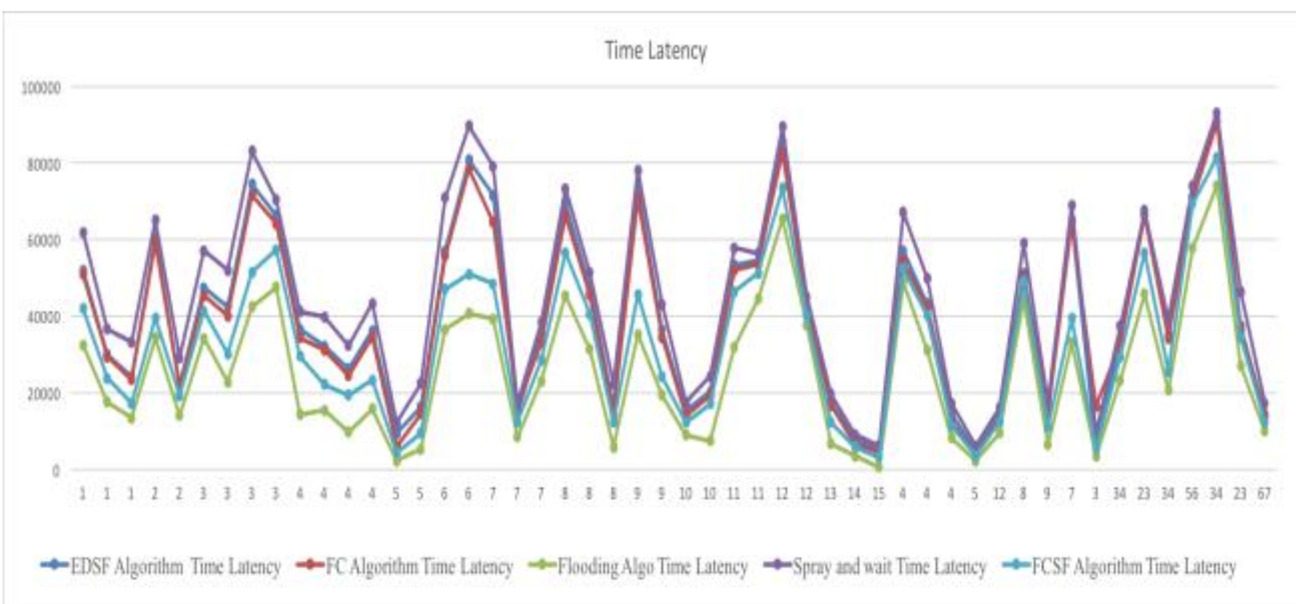

Figure 21 Dataset2 Experiment 5 Comparison on basis of Time Latency

Time Latency:
As shown in figure 21, we can clearly see same results as what we got previously in Experiment 4.

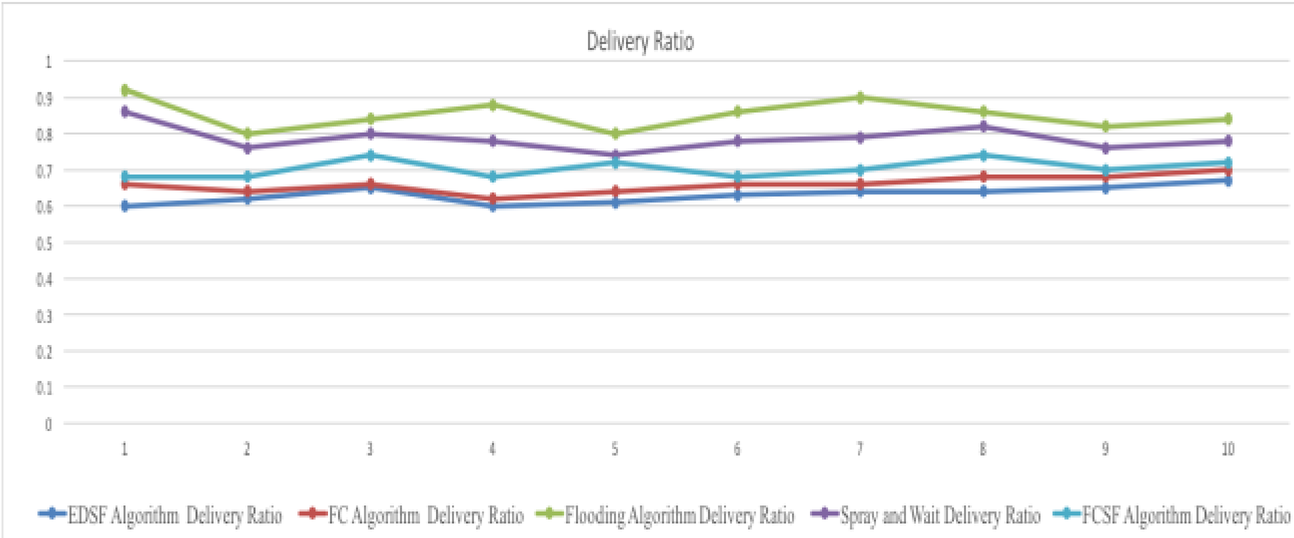

Figure 22 Dataset2 Experiment 5Comparison on basis of Delivery Ratio

Delivery Ratio:

As shown in figure 22, as we increase the number of copies in the network it slightly improves the performance of our EDSF algorithm, FC algorithm and FCSF algorithm. It can be easily examined by comparing it with Experiment 4.

f) Experiment 6
In this experiment, we increased the number of copies from 5 to 10 to see how it affects the results of our algorithms compared to existing algorithms in the terms of Time Latency, Number of Forwardings and Delivery ratio.

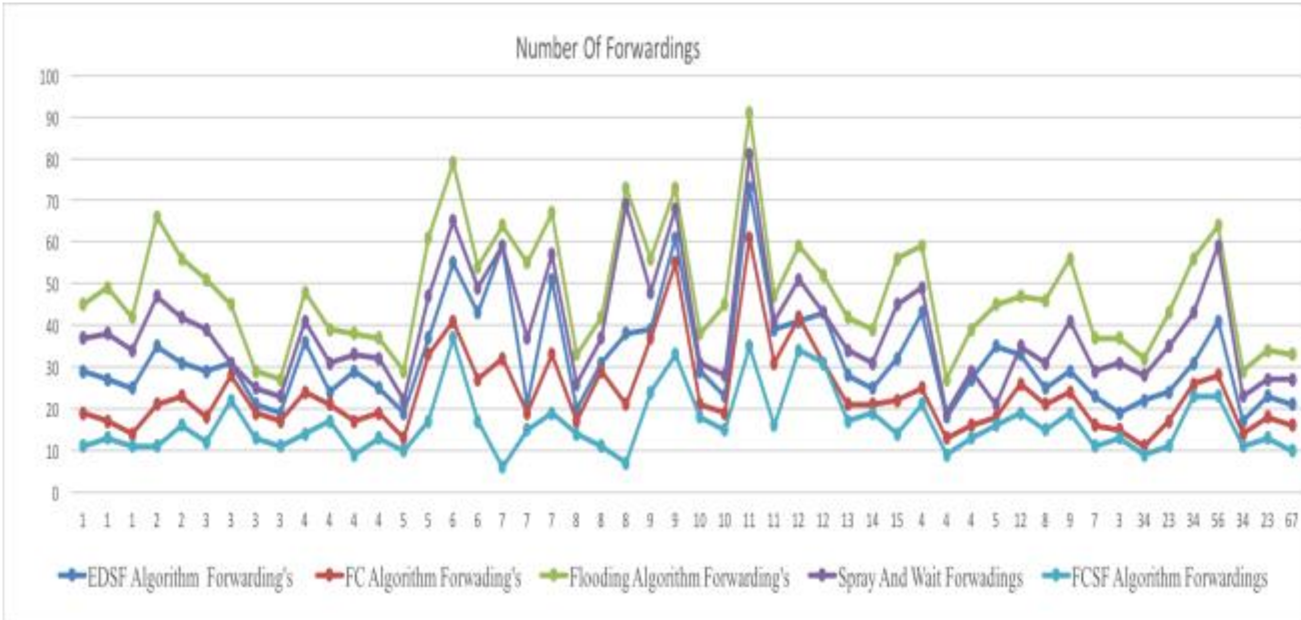

Figure 23 Dataset2 Experiment 6 Comparison on basis of Forwardings

Number of Forwardings:

As shown in figure 23, we can clearly see the same results as generated previously in Experiment 4 and Experiment 5.

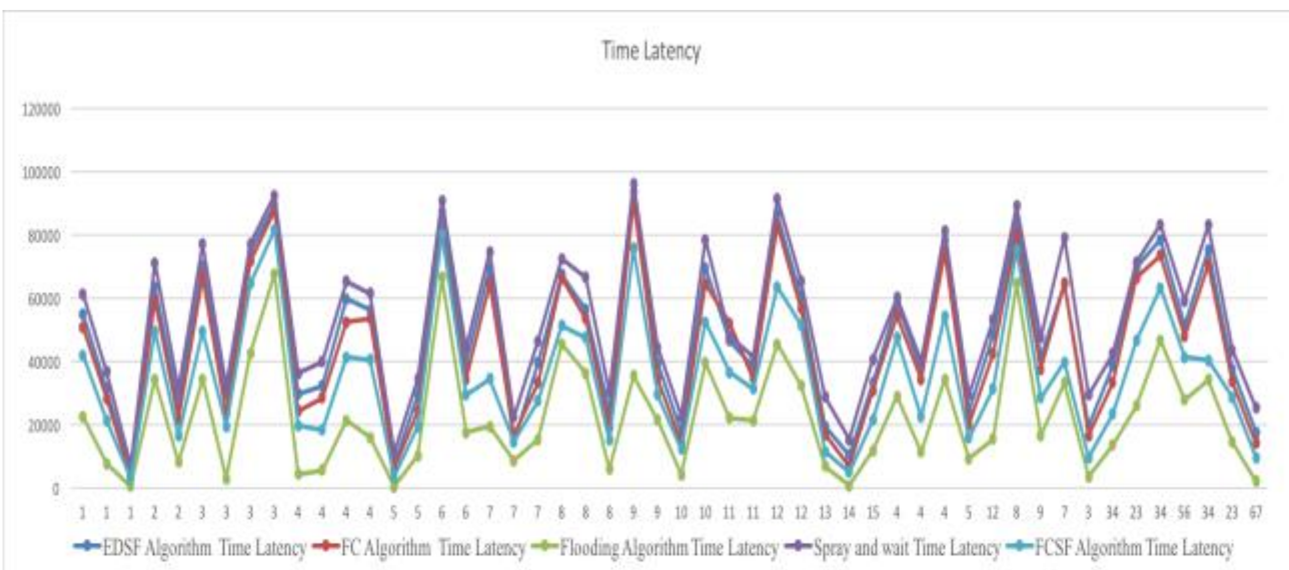

Figure 24 Dataset2 Experiment 6 Comparison on basis of Time Latency

Time Latency:

As shown in figure 24, we can clearly see the same results as that of previously generated in Experiment 4 and Experiment 5.

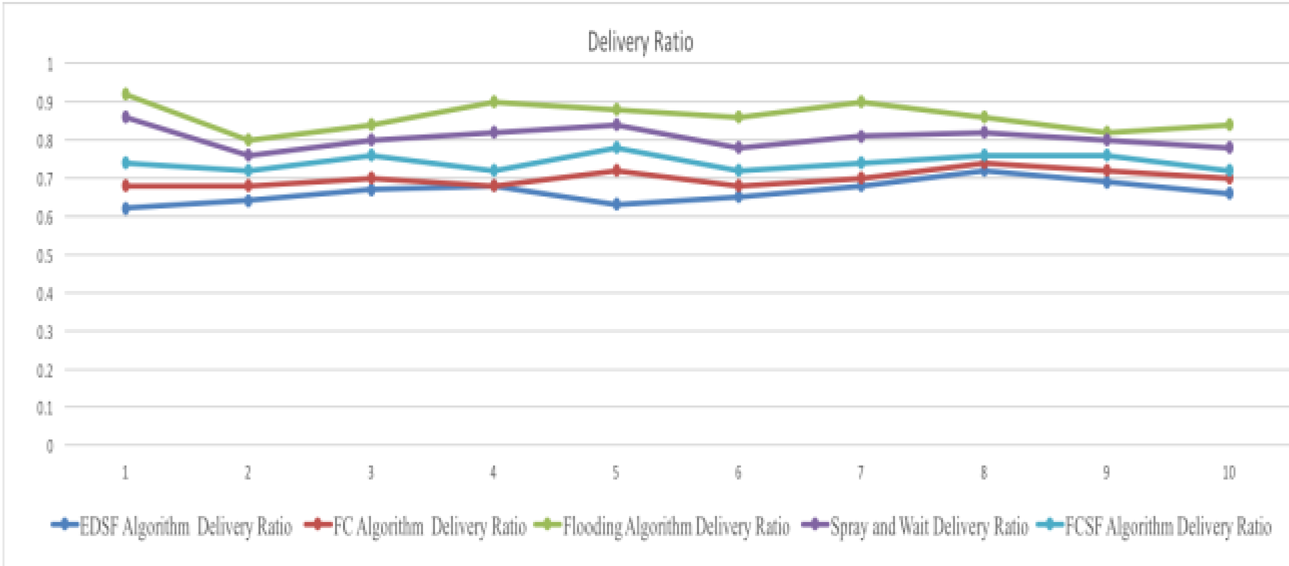

Figure 25 Dataset2 Experiment 3 Comparison on basis of Delivery Ratio

Delivery Ratio:

As shown in figure 25, As we increase the number of copies in the network it slightly improves the performance of our EDSF algorithm, FC algorithm and FCSF algorithm, we can see that by comparing it with Experiment 1 and Experiment 2.

Conclusion for Dataset 2:

From Experiment 4, 5 and 6, FCSF algorithm consistently outperforms in terms of number of forwardings and showed continuous increase in delivery ratio. This means by adding social feature and friend circle we can improve routing performance and give very substantiate outcome for message forwarding. In this way, it can be verified that by using enhanced dynamic social feature, routing performance can be improved by considering source and destination of friend circle feature and social feature.

Summary:

In summary, by considering both datasets i.e. Infocom 2006 and Socialbluconn 2015, as expected Flooding algorithm showed better performance in time latency and delivery ratio with highest number of copies, also spray and wait algorithm performed better in delivery ratio with high time latency and number of forwardings, but DSF Algorithm, FC Algorithm and FCSF algorithm with 2,5 and 10 message copies showed improved performance in delivery ratio with less number of forwardings and time latency was better than spray and wait algorithm but not as good as compared to flooding. From these results, we can conclude that by considering friend circle feature and social features routing performance in OMSN can be improved.

VI. CONCLUSION AND FUTURE WORK

Conclusion

In this thesis, we proposed three Algorithms, EDSF Algorithm, FC Algorithm and FCSF Algorithm. In the algorithm, we used enhanced dynamic social feature and online social feature to more accurately capture nodes on their contact behavior in the network to form a friend circle. Simulation results were given using two real traces of an OMSN which showed that our new algorithms consistently outperform the existing one in number of forwardings, time latency and continually growing delivery ratio.

Future work

In the future, we will continue to improve the efficiency of our algorithms and testing them using different traces with new social features in OMSNs as they become available.

REFERENCES

[1] Delay tolerant networking research group. http://www.dtnrg.org.

[2] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Spray and wait: Efficient routing in intermittently connected mobile networks," in *Proc. ACM SIGCOMM Workshop on Delay Tolerant Networking (WDTN)*, 2005.

[3] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *Proc. IEEE Conf. Sensor and Ad Hoc Communications and Networks (SECON)*, 2004, pp. 235–244.

[4] T.Spyropoulos,K.Psounis,andC.S.Raghavendra,"Efficient Routing in intermittently connected mobile networks: the single-copy case," Univ. Southern California, Los Angeles, Tech. Rep. CENG-2005-10, 2005.

[5] Mingjun Xiao, JieWu,and Liusheng Huang, Community-home-based Multi-copy Routing inMobile Social Networks, Transactions on Parallel And Distributed Systems Vol: Pp No:99 Year 2014.

[6] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," in Proc. of ACM MobiHoc, 2008

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

[8] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Multiple-copy routing in intermittently connected mobile networks. Technical Report CENG-2004-12,USC,2004.

[9] Mingjun Xiao, *Member, IEEE*, JieWu,,and Liusheng Huang, *Member*, *"*Community-home-based Multi-copy Routing in Mobile Social Networks *"*IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS VOL:PP NO:99 YEAR 2014

[10] Rongxing Lu, Xiaodong Lin, and Xuemin (Sherman) Shen, SPRING: A Social-based Privacy-preserving PacketForwarding Protocol for Vehicular Delay Tolerant Networks, IEEE INFOCOM 2010.

[11] Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *SIGMO- BILE Mobile Computing and Communication Review*, 2003.

[12] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. In *Proc. of ACM SIGCOMM*, 2004.

[13] Yuxing He, Cong Liu, Yan Pan, Jun Zhang, Jie Wu, Yaxiong Zhao, Shuhui Yang, and Mingming Lu. *A* Localized Efficient Forwarding Algorithm in Large-scale Delay Tolerant Networks.In*2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems.*

[14] Charles Shang, Britney Wong, Xiao Chen, Wenzhong Li, Suho Oh. Community and Social Feature-based Multicast in Opportunistic Mobile Social Networks-2014.

[15] Ze Li and Haiying Shen ” Utility-based Distributed Routing in Intermittently Connected Networks “37th International Conference on Parallel Processing,2008.

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

[17] J. Wu and Y. Wang, “Social feature-based multi-path routing in delay tolerant networks,” in INFOCOM, 2012 Proceedings IEEE, March 2012, pp. 1368–1376

[18] A. Mei, G. Morabito, P. Santi, and J. Stefa. Social-aware stateless forwarding in pocket switched networks. In *Proceedings of IEEE INFOCOM*, pages 251–255, 2011.

[19] J. Wu and Y. Wang, Opportunistic Mobile Social Networks, 1st ed. Hoboken: CRC Press, 2014;2015.

[20] https://en.wikipedia.org/wiki/Opportunistic_mobile_social_network

[21] A. Pentland, R. Fletcher, and A. Hasson. Daknet: rethinking connectivity in developing nations. Computer, 37(1):78–83, 2004.

[22] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In Proceedings of IEEE INFOCOM, volume 6, pages 1–11. Barcelona, Spain, 2006.

[23] J. Scott, J. Crowcroft, P. Hui, and C. Diot. Haggle: a networking architecture designed around mobile users, 2006.

[24] J. Wyatt, S. Burleigh, R. Jones, L. Torgerson, and S. Wissler. Disruption tolerant networking flight validation experiment on nasa'sepoxi mission. In Procceedings of SPACOMM, pages 187–196, 2009.

[25] 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.

[26] A. cupoto, A. Socievole, F.D Rango, Crawdad trace uncial/Socialblueconn dataset (v.2015-02-08) (downloaded from http://crawdad.org/unical/socialblueconn/20150208/)