IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⸎ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11154

# Exploiting Friendship Relations for Efficient Routing in Mobile Social Networks

**Girish Mantha[1], Sathyanarayana K B[2], Pradeep H K[3]**

Assistant Professor, ISE, JNNCE, Shivamogga, India[1]

Assistant Professor, ISE, JNNCE, Shivamogga, India[2]

Associate Professor, ISE, JNNCE, Shivamogga, India[3]

**Abstract**: Routing in delay tolerant networks is a challenging problem due to the intermittent connectivity between nodes resulting in the frequent absence of end-to-end path for any source-destination pair at any given time. Recently, this problem has attracted a great deal of interest and several approaches have been proposed. Since Mobile Social Networks (MSNs) are increasingly popular type of Delay Tolerant Networks (DTNs), making accurate analysis of social network properties of these networks is essential for designing efficient routing protocols. In this work, a new metric that detects the quality of friendship relations between nodes accurately is introduced. Utilizing this metric, each node defines the community of nodes having close friendship relations with it either directly or indirectly. Later Friendship-Based Routing algorithm is introduced which uses this community information in forwarding of messages. Extensive simulations show that the introduced algorithm works efficiently.

**Keywords**: Social Network, MSN, DTN, Routing protocols, Metrix

## I. INTRODUCTION

The incredible growth in the capabilities and functionality of mobile devices has enabled new applications to emerge. Due to the potential for node mobility, along with significant node heterogeneity, characteristics such as very large delays, intermittent links and high link error rates pose a new set of challenges. Along with these challenges, end-to-end paths are assumed not to exist and message relay approaches are often adopted. While message flooding happens to be a simple and robust solution for such cases, its cost in terms of network resource consumption is unaffordable.
Routing in delay tolerant networks is a challenging problem due to the intermittent connectivity between nodes resulting in the frequent absence of end-to-end path for any source-destination pair at any given time. This problem has attracted a great deal of interest and several approaches have been proposed. Since Mobile Social Networks (MSNs) are increasingly popular type of Delay Tolerant Networks (DTNs), making accurate analysis of social network properties of these networks is essential for designing efficient routing protocols.
Delay Tolerant Networks [1] [2] are a class of wireless networks in which a stable path from source to destination is unlikely to exist at any time instance, thus, long and variable delays occur in routing of messages. These networks are usually sparse and the connection between their nodes changes frequently. Among many real life examples of DTNs, Mobile Social Networks (MSNs) are of growing significance as a result of the rapid and wide spread use of various personal wireless devices (e.g., cell phones, GPS devices) among people and their surroundings.
In Mobile Social Networks, there is a potential of collaborative data gathering via already deployed and human maintained devices. Therefore, opportunistic routing of messages in these networks has been studied by many researchers. However, due to the challenging network environment (intermittent connectivity causing lack of stable end-to-end path between nodes) in these networks, efficient routing of messages is not an easy task. To ease these difficulties and enable nodes to make right forwarding decisions while routing messages, inherent social network properties of these networks need to be utilized. The direct connectivity (opportunity for message transfers) between human-carried devices is enabled when they get into each other’s range. Thus, the relationship defining the frequency and duration of the connectivity between nodes has to be analyzed to route messages efficiently. For example, consider a high school network. A student has a higher chance to see students in the same class (and therefore higher chance to transfer data to them) than the students from other classes that this student can meet only during breaks.
Due to intermittent connectivity between nodes in MSNs determining routing path, considers values of various matrices Like Encounter Frequency, Average contact period, average separation period etc. Encounter Frequency is the number times two nodes come in contact, higher the contact times, larger the frequency and hence larger probability of data transfer between them. If a node has data to send it will into look into the node which has high encounter frequency, in doing so it will reduced the latency since the transmission will be fast. Total or average contact period is total time two nodes in data transfer mode over the time period. Larger contact time means, two nodes are socially close and hence larger the probability of data transfer when either node have the data to forward. Total time in between two nodes over

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11154

the time period is average separation period. Larger the average separation period, encounter frequency will be less and less probability of exchanging the data between such nodes.

Efficiency of routing algorithms are calculated based on Delivery ratio and Cost of routing. Delivery ratio is percentage of data received at the correct destination. Average cost is number of forwards needed to send data to correct destination. Number of forwards is count of number of hops from source to destination. Cost of routing is less if data takes less number of hops to reach destination. A Routing algorithm is more efficient if delivery ratio is high and average cost is less.

## II. Problem Specification

In this work, the problem of efficient routing in Delay Tolerant Networks is addressed. This is a challenging problem as there exists intermittent connectivity between nodes. Since Mobile Social Networks are increasingly popular type of DTNs, making accurate analysis of social network properties of these networks is essential for designing efficient routing protocols. An efficient routing algorithm needs to be designed for MSNs that considers the metrics such as high frequency and longevity.

Objectives

The following the Objectives of the study:
1. Design and Implementation of efficient routing algorithm in Mobile Social Networks. Although the connectivity of nodes is not constantly maintained, it is still desirable to allow communication between nodes. Therefore, it is necessary to pro- vide a routing protocol which tries to route packets throughout the times the link is available among the nodes. But this cannot be done by standard routing algorithms which assume that the network is connected most of the time.

2. To ensure that the use of Friendship behaviour between nodes in mobile social networks will help in building efficient routing algorithm. Best possible metrics for forwarding in Mobile social network is by analysing the social behaviour of each node and when a node has packet to send to destination, its social contacts are used for determining the route.

3. To ensure that Network will take less time to deliver the messages with least cost and with most efficient way. The designed Routing Algorithm should take less time to forward the packets to destination and hence with more efficiency decreasing the delay.

4. Comprehensive analysis is to be done on delivery ratio with respect to Encounter times and encounter history. Also with respect to Delivery ratio versus time of delivery of packets. Analysis should be done based on effect of varying Buffer space at each node and by varying Time-To-Live (TTL) for each message on Delivery ratio and performance.

## III. Design and Implementation

### A. Opportunistic Routing in MSN

The intermittent connectivity between nodes in an MSN makes the routing of messages possible only in opportunistic manner. That is, message exchanges occur only when two nodes come within the range of each other and one of them assesses that it has lower delivery probability than the other. Hence, the link quality between each pair of nodes needs to be estimated accurately (from contact history) to consider the possible forwarding opportunity arising from the encounter. As a result, the periodic encounters between nodes can be condensed to a single link weight and the corresponding graph including links with weights exceeding certain threshold can be constructed.

Impact Factor 7.39 ⁜ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11154

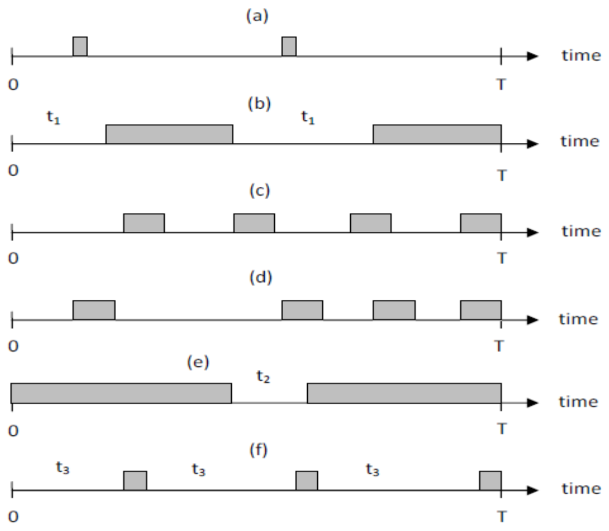


Fig 1: Six different encounter histories between nodes i and j in the time interval [0, T]. Shaded boxes show durations of the encounters between nodes.

To overcome deficiencies of exiting metric in defining quality of friendship between accurately, in this work a new metric is defined called Social Pressure metric (SPM). SPM helps each node to build friendship community, through which it will decide about forwarding opportunities to contacted node.

## B. Social Pressure Matrics:

To find a link metric that reflects the node relations (also the forwarding opportunities) more accurately, the following three behavioural features of close friendships: high frequency, longevity, and regularity are considered. In other words, for two nodes to be considered friends of each other, they need to contact frequently and regularly in long-lasting sessions. Here, frequency refers to average intermeeting time while regularity refers to the variance of the intermeeting time. Hence, two nodes may meet infrequently but regularly (e.g., once a week) and still be considered friends. This is of course a weaker friendship than the one with both frequent and regular contacts. The previous metrics take into account some of these features but not all of them at the same time. We account for these properties in a new metric that we called **Social Pressure Metric (SPM)**. It may be interpreted as a measure of a social pressure that motivates friends to meet to share their experiences. In this setting, this amounts to answering the question "what would be the limit with time unit tending to zero of the average message forwarding delay to one node j if the other i had a new message to deliver at each time unit?" (to make the measure, time independent, limits are used). Then, we define the link quality ($w_{i,j}$) between each pair as the inverse of this value. More formally:

$$SPM_{i,j} = \frac{\int_{t=0}^{T} f(t)dt}{T} \qquad \text{(Eq. 1)}$$

And

$$w_{i,j} = \frac{1}{SPM_{i,j}} \qquad \text{(Eq. 2)}$$

where f(t) represents the time remaining to the next encounter of the two nodes at time t. If at time t, the nodes are in contact, then f(t) = 0, otherwise, f(t) = $t_{next}$ - t, where $t_{next}$ is the time of the next meeting between nodes i and j. Hence, each intermeeting time $t_{inter}$ contributes the term $t^2_{inter}/(2T)$ to SPM. If there are n intermeeting times in the time period T, then

$$SPM_{i,j} = \left(\sum_{x=1}^{n} t^2_{inter,x}\right)/(2T) \qquad \text{(Eq. 3)}$$

and

$$w_{i,j} = (2T)/\left(\sum_{x=1}^{n} t^2_{inter,x}\right) \qquad \text{(Eq. 4)}$$

The larger the value of $w_{i,j}$, the closer the friendship (the higher the forwarding opportunities) between nodes i and j. Clearly, increasing the time the nodes are in contact decreases SPM as does equalizing the time between encounters. Finally, splitting the intermeeting times into the larger number of smaller pieces also decreases the SPM. Hence, indeed, this metric combines the three desired properties of the friendship behaviors discussed above into a single measure. To illustrate the benefits of this metric, we notice that when it is used to evaluate all cases in Fig. 1, the resulting weights will accurately indicate which case offers more forwarding opportunities. It should also be noted that SPM is computed from the history of the encounters of the node. As additional node encounters happen, the corresponding SPM value is updated easily. When node s wants to send data, it will determine the route based on SPM values, Smaller the value of SPM higher the probability to forward.

## C. System Architecture

The high level design of the Social based routing algorithm comprises of five modules is shown in Figure 2. The proposed social based routing consists of different modules such as, node initialize, Data receiver, Neighbor Detected, neighbor left, Friendship based community formation.

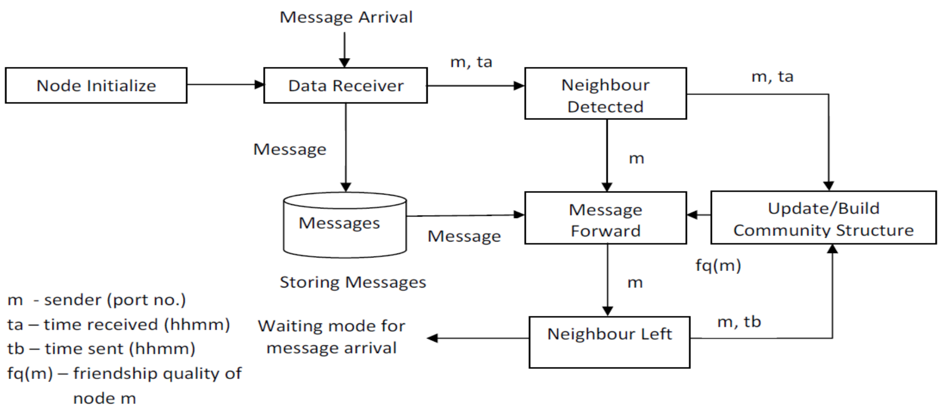


Fig 2: System Architecture.

At the beginning 'Node Initialize' method in each node will initializes nodes parameters. Data receiver will be waiting for messages from contacted node, when a message is received, it checks if destination is self, if so, calls neighbour detected else it will store the messages for later sending. Neighbour detected module updates the timers with ta of receiving node. Update/Build community module updates SPM and RSPM values to form friendship community. Message forward module will send messages from list to node m. once the contacted node finishes sending message, it calls neighbour left module to update to timers with tb for the node m.

## D. Node Initialization

At the beginning each node initializes its parameters as described in below pseudo code. When a new node i comes in contact with current node j, new nodes parameters are initialized once, for all future contacts with node j with i.

**Pseudo code 1** initialize (node j)

```
1: for each i ∈ N and i! = j do
2:      cur_total1 [i] = 0
3:      $t_{end}^{pre}$ (i) = 0
4:      for each k ∈ N and k! =i and k! =j do
5:           $t_a^{start}$ (i, k) = 0
6:           $t_a^{end}$ (i, k) = 0
7:           cur_total2 [i][k] = 0
8:      end for
9: end for
```

Pseudo code 1 initialises parameters cur_total1 and time of last contact, needed to calculate Social Pressure Matrix (SPM) and start and end times of previous contact times of node k with node i, current contact time to calculate Relative Social Pressure Matrix (RSPM). SPM values are needed to find direct friends and RSPM values are needed to find indirect friends on current node with respect to contacted node.

## E. Data Receiver

Data receiver waits for nodes contacting the current node to forward packets. When a packet is received, node checks if the destination is current node or not. If the destination is self, it calls neighborDetected method to process. If the destination is not current node, message is stored for future forwarding. Pseudo code 2 describes the Data Receiver process.

**Pseudo code 2:** receive (node m, time t)

```
1.if destination = current_node
2.        Call neighborDetected
3.        Call messageForward
4.        Call neighborLeft
5.else
6.        Store the packet in the list.
7.End if else
```

F. Neighbour Detected by Handling Direct Relationships

When a node j sends a packet to node i, and if the destination of that packet is node i itself, then node i will calculates its direct and indirect friends for future use. Below sections gives insight about how these are updated. When node i has packets to send, and if node j comes in contact the node i, it will decide based on SPM values to forward the packets to it or not. SPM values are calculated as below.

$$SPM_{i,j} = \frac{\int_{t=0}^{T} f(t)dt}{T} \qquad \text{(Eq. 5)}$$

and

$$w_{i,j} = \frac{1}{SPM_{i,j}} \qquad \text{(Eq. 6)}$$

where f(t) represents the time remaining to the next encounter of the two nodes at time t. If at time t, the nodes are in contact, then f(t) = 0, otherwise, f(t) = $t_{next}$ - t, where $t_{next}$ is the time of the next meeting between nodes i and j. Hence, each intermeeting time $t_{inter}$ contributes the term $t^2_{inter}/(2T)$ to SPM. If there are n intermeeting times in the time period T, then

$$SPM_{i,j} = \left(\sum_{x=1}^{n} t^2_{inter,x}\right)/(2T) \qquad \text{(Eq. 7)}$$

and

$$w_{i,j} = (2T)/\left(\sum_{x=1}^{n} t^2_{inter,x}\right) \qquad \text{(Eq. 8)}$$

G. Neighbour Detected by Handling Indirect Relationships

To find indirect friendships between nodes in a way relevant for routing, we propose to use relative SPM (or simply RSPM) metric. Consider the sample encounter history shown in Figure 3 in which the upper diagram shows the contacts between nodes i and j, while the lower one shows the contacts between nodes j and k. We define $RSPM_{i,k|j}$ as the answer to the question 'what would be the average delivery delay of node i's continuously generated messages if they followed the path <i,j,k>?'. Each indirect information passing consists of two stages. The first one starts at the last meeting of node i with node j and ends at the time node i's next contact with node j ends (assuming that any message generated at node i can be transferred to node j when they are in contact). Here, if there are several subsequent meetings with j before any meeting of j with k, then the last one is considered. We denote duration of this stage as $t_{a,x}$ where x denotes the number of indirect information passing occurring. During this stage, node i transfers messages to node j. The second stage starts when the first one ends and it finishes when node j meets node k. The duration of this session is denoted $t_{b,x}$. During this stage, the messages accumulated at j merely wait for the meeting with the destination (without accumulating further at node j). Example is given in Figure 2, in which in time T, there are three full information passing sessions between nodes i and k via node j, and the beginning of the fourth one. Denoting the number of such sessions as n, RSPM $_{i,k|j}$ is computed as:

$$RSPM_{i,j|k} = \left(\sum_{x=1}^{n} \int_{0}^{t_{a,x}} (t_{b,x} + t_{a,x} - t)dt\right)/T \qquad \text{(Eq. 9)}$$

$$= \frac{\sum_{x=1}^{n}\left(2\ t_{b,x}\ t_{a,x} + t^2_{a,x}\right)}{2T} \qquad \text{(Eq. 10)}$$

Since the intermediate node, j, records all of its past contact times with i and k, it can compute the value of RSPM $_{i,k|j}$.

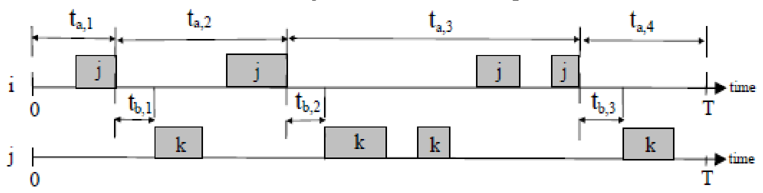


Fig 3: Encounter history between nodes i and j (upper diagram) and between nodes j and k (lower diagram) in the same time interval [0, T].

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⪪ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11154

When a new node m is encountered, node j updates the value of SPM[m] and for each of its other contacts i, it updates the value of RSPM[i][m] if node m is encountered first time after its meeting with node i (Pseudo code 3).

**Pseudo code 3** neighborDetected (node m, time t)

```
1: $t_{cur}^{start}$ (m) = t
2: nt = $t_{cur}^{start}$ (m) – $t_{pre}^{start}$ (m)
3: cur total1[m] += nt(nt+1)/2
4: SPM[m] = cur_total[m]/2t
5: for each i ∈ N and i != m do
6:     $t_b^{end}$ (i,m) = t
7:     if $t_a^{end}$ (i,m) > $t_a^{start}$ (i,m) then
8:         $t_b$(i,m) = $t_b^{end}$ (i,m) – $t_b^{start}$ (i,m)
9:         $t_a$(i,m) = $t_a^{end}$ (i,m) - $t_a^{start}$ (i,m)
10:        cur_total2[i][m]+=2$t_b$(i,m)$t_a$(i,m)+($t_a$(i,m))$^2$
11:        RSPM[i][m] = cur_total2[i][m]/2t
12:        $t_a^{start}$ (i,m)= $t_a^{end}$ (i,m)
13:    end if
14: end for
```

H. Neighbour left

When the meeting of a node with another node m ends, it also updates the value of SPM[m] and sets the end time of current $t_a$(m, k) and start time of next $t_b$(m, k) to the current time t for each of its other contacts k (Pseudo code 4).

**Pseudo code 4** neighborLeft (node m, time t)

```
1: $t_{pre}^{end}$ (m) = t
2: SPM[m] = cur_total1[m]/2t
3: for each k ∈ N and k != m do
4:     $t_a^{end}$ (m, k) = t
5:     $t_b^{start}$ ( (m, k) = t
6: end for
```

I. Friendship Community Formation

Using encounter history, each node can compute qualities ($w_{i,j}$ values) of its links with other nodes using SPM values. Then, it can define its friendship community as a set of nodes having a link quality with itself larger than a threshold (τ). This set will include only direct friends. However, two nodes that are not close friends directly (they even may not have contacts at all) still can be close indirect friends. This happens if they have a very close friend in common so that they can contact frequently through this common friend. To detect these indirect friendships, a node needs RSPM values from its friends. Once such RSPM Values are received and updated at the encounter times with friends, each node can form its friendship community using the following equation:

$$F_i = \{j \mid w_{i,j} > \tau \text{ and } i \,!= j\} \cup \{k \mid w_{i,j,k} > \tau \text{ and } w_{i,j} > \tau \text{ and } i \,!= j!=k\} \qquad \text{(Eq. 11)}$$

where $w_{i,j,k}$ = 1/ RSPM $_{i,k|j}$ . The above equation enables nodes to detect their one-hop direct and two-hop indirect friends. The introduced method for detecting the indirect strong links between nodes is different than previous approaches which basically consider the links between node pairs separately and assume a virtual link between node i and k if $w_{i,j}$ $w_{j,k.}$ > τ. However, in this model, we can detect indirect relations more accurately. For example, if node j has a weak direct link with node k, $w_{i,j}$ $w_{j,k.}$ may be less than τ. However, if node j usually meets node k in a short time right after its meeting with node i, our metric can still identify node k as a friend of node i. This definition of indirect node relations is particularly meaningful within the context of routing because a node receives a message from one of its contacts and sends it to another contact. That is, it holds the message between its meetings with two different nodes. Hence, this metric accurately estimates the quality of indirect message exchange opportunity between two nodes.

Relativity is significant in handling indirect node relations because recent studies [34]-[35] have shown that the intermeeting times between most of the node pairs fit to log-normal distribution. Thus, their future contact times might depend on their past contacts and the time passed since their last encounters (due to non-memoryless property of the log-normal distribution). Moreover, some other studies [36] point out that the mobility of many real objects are at least weakly periodic in which case also the future contact times depend on the time passed since the last encounter.

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⫘ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11154

J. Forwarding Algorithm

Once a node constructs its friendship community for each period based on its current encounter history, it decides whether to forward a message to the encountered node using the procedure in Pseudo code 5. If a node i having a message for node d meets with node j, it forwards the message to j if and only if node j's friendship community ($F_j$(pid)) and node j is a stronger friend of node d than node i is. Accordingly, even if node j has a stronger link with node d than node i has, if node j does not include d in its current friendship community (i.e., weight of link between j and d is less than τ), node i will not forward the message to node j.

**Pseudo code 5**. messageForward (met node j, time t, periodlength l)

```
1:  // i = id of the node running the algorithm
2:  Request/Receive friendship quality (fq(j,d) = max{ $w_{j,d}$, wj,any,d}) of j for each destination d of i's current messages
3: // fq(j,d) is not returned if it is less than τ
4:  **for** each message m with destination d **do**
5:          **if** fq(j,d) > fq(i,d) **then**
6:                  Forward the message m to j
7:          **end if**
8: **end for**
```

fq(i.j) is friendship quality of node of i. fq(i.j) returns the friendship quality(weight) of the link between node i and j if it is greater than τ. The value of τ will be set on the basis experimental analysis. In this project τ's value is average weight of all friends of node i. The flow chart in Figure 4 shows the message Forwarding scheme.

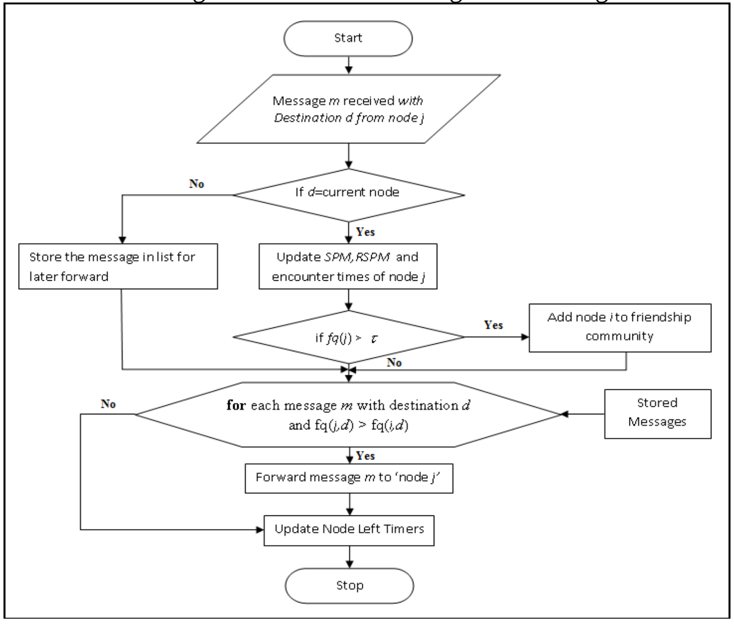

Start
Message m received with Destination d from node j
If d=current node
No
Yes
Store the message in list for later forward
Update SPM, RSPM and encounter times of node j
if fq(j) > τ
Yes
Add node i to friendship community
No
for each message m with destination d and fq(j,d) > fq(i,d)
No
Stored Messages
Yes
Forward message m to 'node j'
Update Node Left Timers
Stop

Fig 4 : System Flowchart

## III. RESULTS AND ANALYSIS

A. Experimental setup

The Routing using proposed algorithm was carried out using with fixed number of nodes and specific format of the data. In this simulation five nodes are configured to send and receive data. Each node will run send module and receive module continuously, sender ready to send message to four other nodes, receiver ready to receive any message to self node or to forward to any other node. Each node acts either as source, destination or intermediate node to messages.

Each node will be waiting for messages at specific port number. Figure 5 shows five nodes each with different port number. Since DTN's are characterized by 'Store-and-forward' based routing, each node will have its own storage. If there is no connection available at a particular time, a DTN node can store and carry data until it encounters other nodes.

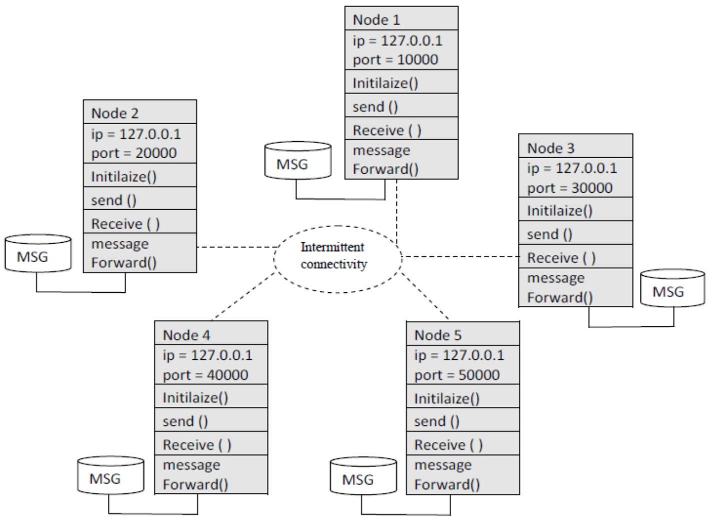


Fig 5 : Experimental Setup

B. Test data and parameters

In the simulations, messages between the nodes are maintained with specific format. Figure 6 shows different fields in the message. Since nodes are static and executed from the single machine, IP address of all them are same. It is assumed that Performance is similar to real time transfer of messages between nodes with different IP address.

| **Message** | | | | | | |
|---|---|---|---|---|---|---|
| **Source** | **Destination** | **RSPM** | **Time sent** | **TTL(mSec)** | **Length(mSec)** | **Message** |
| 10000 | 30000 | 1.45 | 10.10.47 | 30000 | 3000 | Message from node 1 |

Fig 6: Message Format

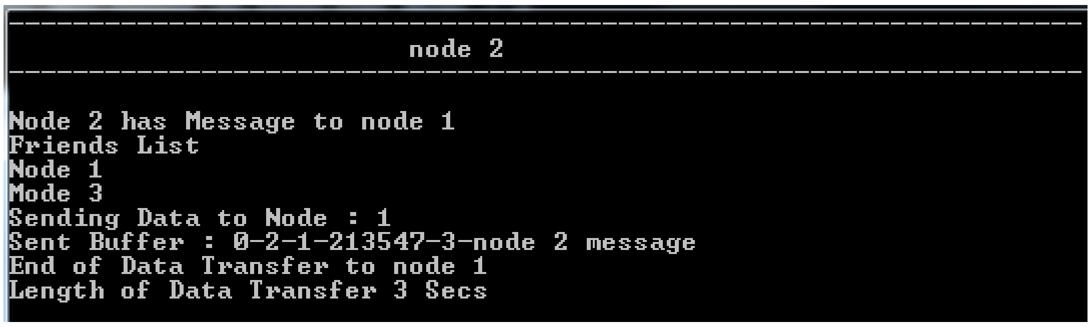
**Source** – port number of sender.
**Destination** – destination port number.
**RSPM** – Relative Social Pressure Matrix
**Time sent** – Messages Delivery time from the source.
**TTL** – Time-To-Live.
**Length** – Duration of message transfer in milliseconds.
**Message** – Data to send

RSPM values gives information about indirect friendship between nodes. Lower the value higher the quality of friends. Time-To-Live filed gives life time of the messages. Based on situation TTL filed value is assumed. At any intermediate node if transfer time exceeds TTL message will be dropped. Simulation is assumed the each message will take certain amount of time to transfer. Length field specifies time message takes to transfer from node to node.

C. Results

A. Message Delivery (1 hop distance) Neighbor Detected and Left

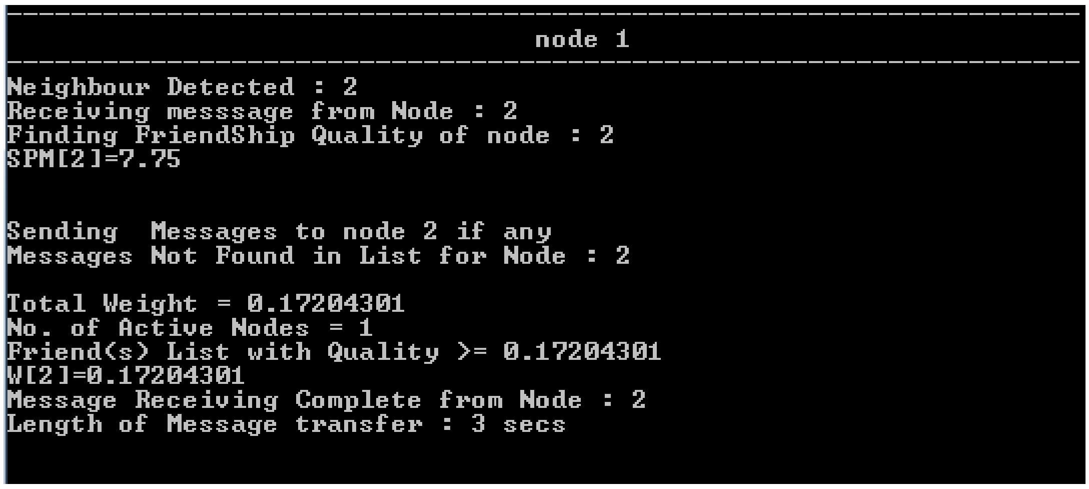
Delay Torrent Networks adopt opportunistic based routing, in this scenario Node 2 have a message to Node1, when it comes in contact with Node1 it will forward the message. Figure 7 shows Node 2 output scenario.



```
-------------------------------------------------------------------------
                              node 2
-------------------------------------------------------------------------

Node 2 has Message to node 1
Friends List
Node 1
Mode 3
Sending Data to Node : 1
Sent Buffer : 0-2-1-213547-3-node 2 message
End of Data Transfer to node 1
Length of Data Transfer 3 Secs
```

Fig 7: Node 2, Message forwarding

Figure 8 shows message received at Node1. Upon receiving a message, Node 2 extracts message to get sent time and calculates SPM values and updates RSPM values of node 2.



```
-----------------------------------------------------------------------
                              node 1
-----------------------------------------------------------------------
Neighbour Detected : 2
Receiving messsage from Node : 2
Finding FriendShip Quality of node : 2
SPM[2]=7.75


Sending  Messages to node 2 if any
Messages Not Found in List for Node : 2

Total Weight = 0.17204301
No. of Active Nodes = 1
Friend(s) List with Quality >= 0.17204301
W[2]=0.17204301
Message Receiving Complete from Node : 2
Length of Message transfer : 3 secs
```

Fig 8: Node 1, Message received from Node 2

B. SPM and Weights updating

Each node computes SPM and weights to build friendship community. Figure 9 shows encounter history of Node 1 with Node 2 in the time interval [0,100 seconds]. SPM and weights are calculated with respect to Node 1.

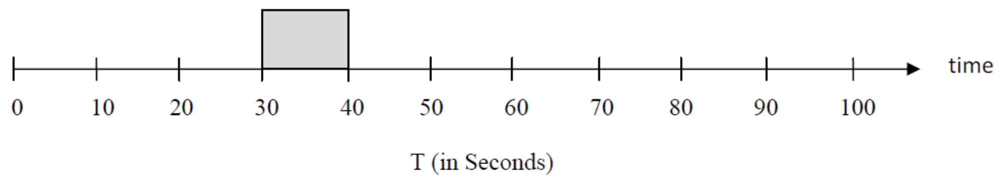


Figure 9: Encounter history between Node 2 and Node 1

We know, After Node 2 detected

$SPM_{i,j} = \left(\sum_{x=1}^{n} t_{inter,x}^{2}\right)/(2T)$ ( Ref Equation 11)

$= (30(30+1)/2) / (2*30)$

$= 7.75$

After Node 2 completes data transfer

$SPM_{i,j} = \left(\sum_{x=1}^{n} t_{inter,x}^{2}\right)/(2T)$ ( Ref Equation 12)

$= 465/(2*40)$

$= 5.81$

Calculating weights for friendship community formation, we know

$w_{i,j} = \frac{1}{SPM_{i,j}}$ (Ref Equation 13)

$= 1/5.81$

$= 0.1721$

Figure 5.4 shows the output screen of Node 1, after Node 2 has left.

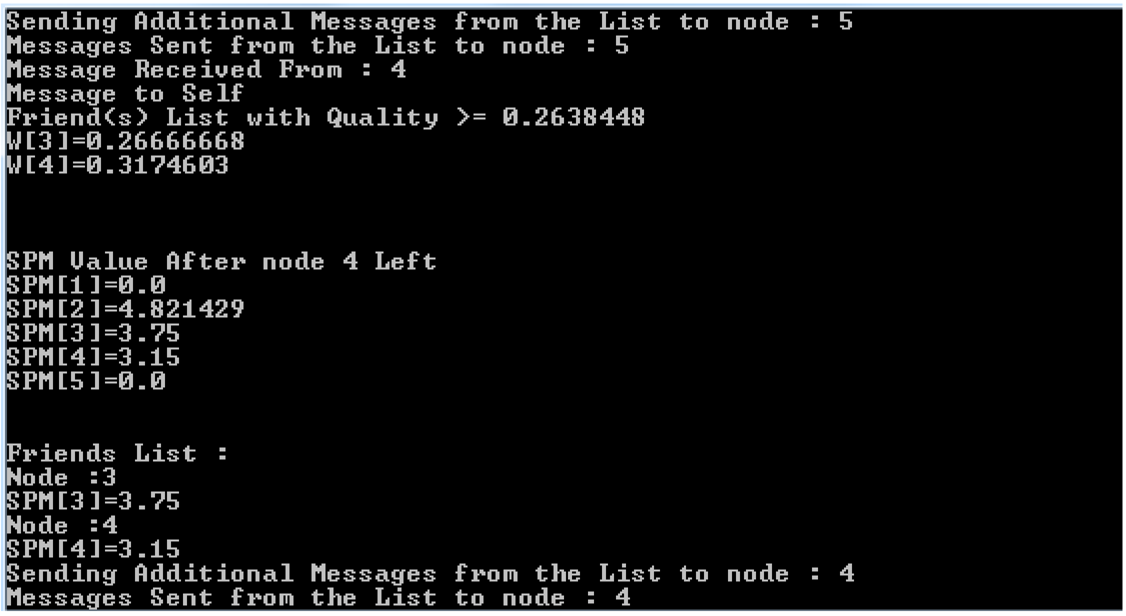
C. Community formation by each node (threshold τ )

When a node encounters other node, based on this encounter history, each node calculates friendship quality. Then it can define its friendship community as a set of nodes having a link quality with itself larger than threshold (τ). Value of τ is implementation dependent. Permissible values are between zero and highest SPM value at that point of time among friends. Smaller the values of τ, larger the community size and vice versa. For the simulation purpose value of τ is assumed to be average value of all weights of nodes currently in friend list.

Figure 10 shows output screen of Node 1. It shows that, Node 1 is currently in contact with Node 2, Node 3, and Node 4. And corresponding values of Weights (link quality) and SPM are shown. Among 3 friends Node 1 chooses Node 3 and Node 4 as its community because Node 3 & 4 has weight less than the average values combining Node 2, 3 and 4.

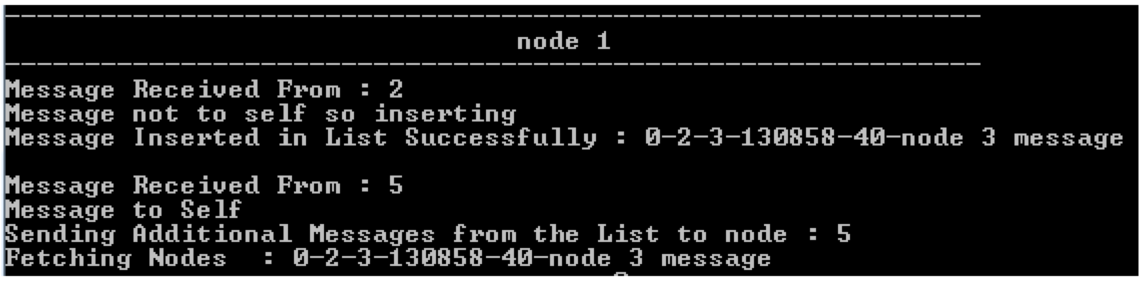
Impact Factor 7.39 ※ Vol. 11, Issue 1, January 2022
DOI: 10.17148/IJARCCE.2022.11154

Whenever any node comes in contact with Node 1, Node 1 will receive all the messages destined to Node 2 and 3. This friendship community of Node 1 may change in future based on encounter times.


```
Sending Additional Messages from the List to node : 5
Messages Sent from the List to node : 5
Message Received From : 4
Message to Self
Friend(s) List with Quality >= 0.2638448
W[3]=0.26666668
W[4]=0.3174603



SPM Value After node 4 Left
SPM[1]=0.0
SPM[2]=4.821429
SPM[3]=3.75
SPM[4]=3.15
SPM[5]=0.0


Friends List :
Node :3
SPM[3]=3.75
Node :4
SPM[4]=3.15
Sending Additional Messages from the List to node : 4
Messages Sent from the List to node : 4
```
Fig 10: Friendship community of Node 1

D. Storing and forwarding of messages

When a node receives message, it extracts the message to check if the message is to the self or not. If the message is to the self, it receives and updates the SPM and RSPM values. If the destination is not self, message is stored in the list. Figure 11 shows output of Node 1 when it receives a message from Node 2. When Node 1 receives a message from Node 2 with destination as Node 3, message is pushed into the list for later forwarding.


```
---------------------------------------------------------------
                             node 1
---------------------------------------------------------------
Message Received From : 2
Message not to self so inserting
Message Inserted in List Successfully : 0-2-3-130858-40-node 3 message

Message Received From : 5
Message to Self
Sending Additional Messages from the List to node : 5
Fetching Nodes  : 0-2-3-130858-40-node 3 message
```
Fig 11: Message Receiving & Storing in Node 1

E. Message forwarding

Once a node constructs its friendship community for each period based on its current encounter history, routing algorithm decides whether to forward a message to the encountered node based on this friendship community. In Figure 12, Node 1 has messages in list with destination (D) 3 and 2. Fq[i]=1 in a node represents, node i belongs to current friendship community of that node.

The sequences of events in message forwarding are:
1. Node 2 comes in contact with Node 1.
2. Node 1 sends message with D = 2 to Node 2 from the List. Since Node 2 is friend with Node 3 it also sends message with D = 3.
3. Node 3 comes in contact with Node 2.
4. Node 2 sends message which it received from Node 1 to Node 3.

Impact Factor 7.39 ⁑ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11154

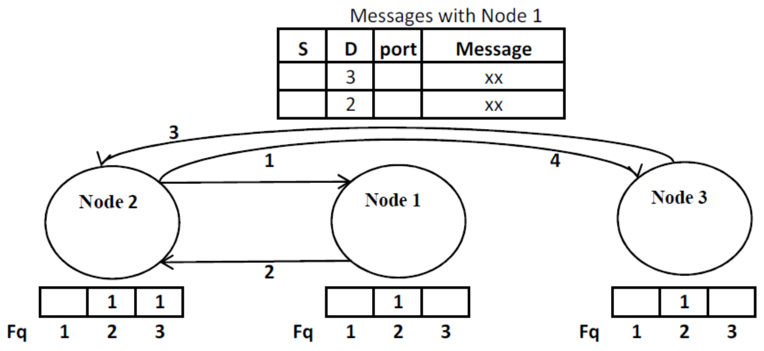
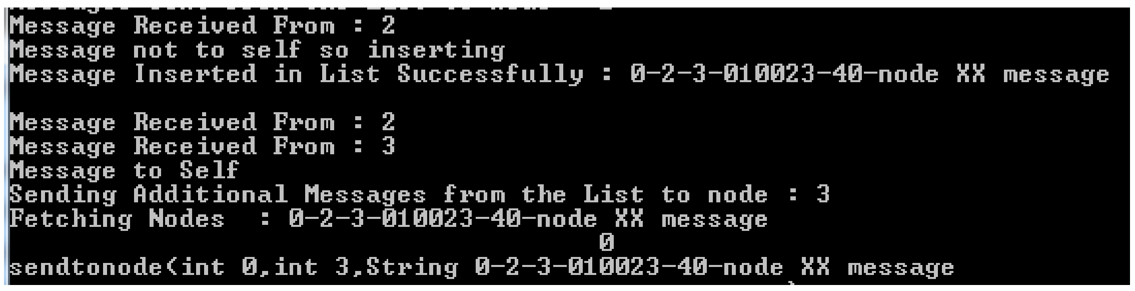


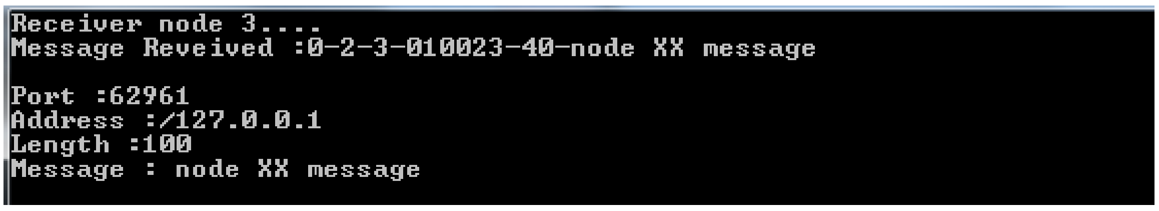
Figure 12: Message forwarding scenario

Output Figure 13 shows that, Node 1 is receiving a message from Node 2 with destination as Node 3. Since message is to be forwarded later it is inserted into the list.



```
Message Received From : 2
Message not to self so inserting
Message Inserted in List Successfully : 0-2-3-010023-40-node XX message

Message Received From : 2
Message Received From : 3
Message to Self
Sending Additional Messages from the List to node : 3
Fetching Nodes  : 0-2-3-010023-40-node XX message
                                         0
sendtonode(int 0,int 3,String 0-2-3-010023-40-node XX message
```

Fig 13: Node 1, Message Received from Node 2



```
Receiver node 3....
Message Reveived :0-2-3-010023-40-node XX message

Port :62961
Address :/127.0.0.1
Length :100
Message : node XX message
```

Fig 14: Node 3, Message Received from Node 1

Figure 14 shows Node 3 receiving message “node xx message” from Node 1.

## F. Buffer and TTL Overflow

In this section two special cases are discussed. First, Buffer overflow condition. Second, TTL overflow. Size of buffer space plays an important role in determining efficiency routing algorithm. If the buffer size is low, rate of overflow will be high, increasing drop rate of messages. Due to increase in the drop rate, more and more messages are lost and hence decreasing the delivery ratio. If the delivery ratio is low routing algorithm is less efficient. If the buffer size if high amount of overhead in parsing the messages at each node is high. Increase in parse time will add to the delivery time hence increasing the delay. If the delay is high routing algorithm is less efficient. Hence proposed algorithm is evaluated against size of the buffer.

TTL determines the time limit within which message have to reach the destination. In real world if a message reaches destination after long delay, it may not have significance. And hence timely delivery of messages is important. Each message is set with a TTL. If TTL value is low, more messages are dropped decreasing the delivery ratio hence decreasing the routing efficiency. High TTL value has no significance. Routing algorithm efficiency is checked against these two parameters.

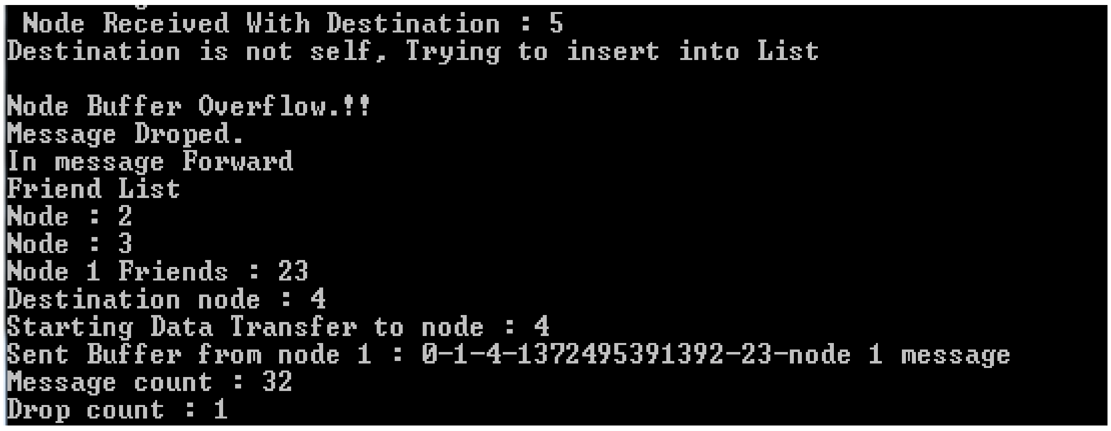
Since Delay Torrent Networks works on opportunistic based routing, they need to store the messages till the nodes come in contact. Since every node is assumed to be having limited buffer size, buffer overflow conditions will arise. In such case incoming message is dropped without informing the sender. Figure 15 shows the overflow scenario for Node 1. Output was taken when Buffer size set to 10.

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⁜ Vol. 11, Issue 1, January 2022

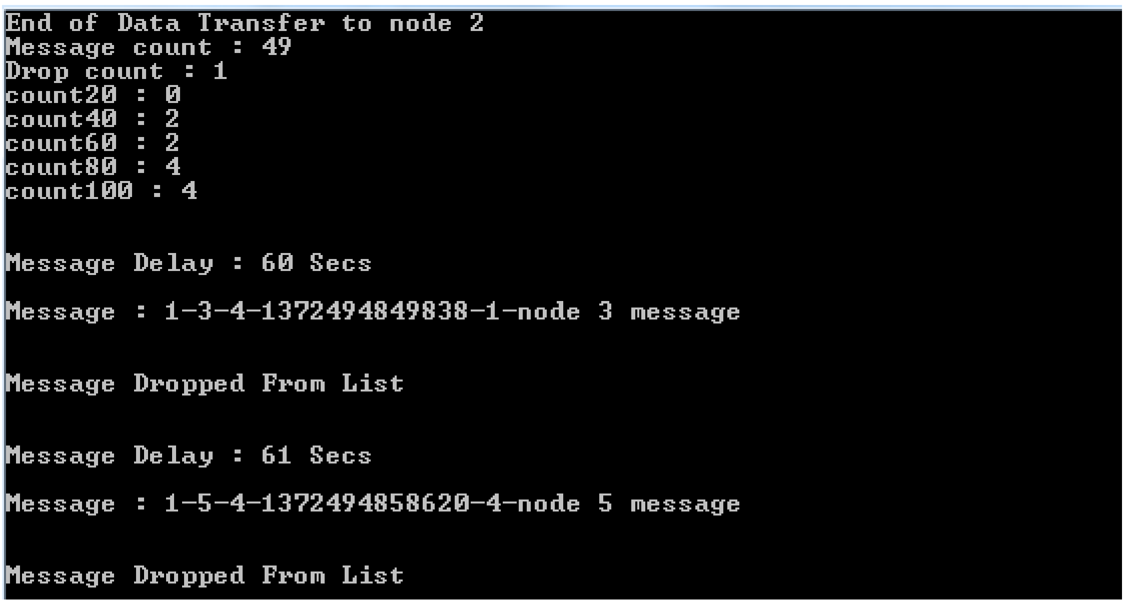
DOI: 10.17148/IJARCCE.2022.11154



```
 Node Received With Destination : 5
Destination is not self, Trying to insert into List

Node Buffer Overflow.!!
Message Droped.
In message Forward
Friend List
Node : 2
Node : 3
Node 1 Friends : 23
Destination node : 4
Starting Data Transfer to node : 4
Sent Buffer from node 1 : 0-1-4-1372495391392-23-node 1 message
Message count : 32
Drop count : 1
```

Fig 15: Buffer overflow in Node 1

Second, Time-To-Live for messages. Social Based Networks practically will not confirm the exact time of delivery because of intermittent connectivity between nodes. Each message from source-destination set with TTL values. In this simulation TTL is varied for different scenarios. Figure 16 shows the output scenario of Node 1 with TTL = 50 seconds. Each node checks TTL field of message, if it time exceeds, that message will be dropped.



```
End of Data Transfer to node 2
Message count : 49
Drop count : 1
count20 : 0
count40 : 2
count60 : 2
count80 : 4
count100 : 4


Message Delay : 60 Secs

Message : 1-3-4-1372494849838-1-node 3 message


Message Dropped From List


Message Delay : 61 Secs

Message : 1-5-4-1372494858620-4-node 5 message


Message Dropped From List
```

Fig 16: TTL overflow in Node 1

G. Performance Analysis

In this section, a detailed result analysis on Friendship quality, Delivery ratio and effects of parameters like Time-To-Live and Buffer size on Delivery ratio are analyzed in detail.

The basis of routing in MSN is accurate analysis of node movements. The more accurate the values are, higher the efficiency of routing algorithm. Relations between nodes are called friendship qualities. In the proposed algorithm a new friendship quality metric SPM is introduced. The values of SPM are based on three basic properties of MSN nodes like high frequency, longevity and regularity. The values of SPM determine the quality of friendship between nodes. The efficiency of routing algorithm depends on the accurate calculation of SPM values. Hence, for performance analysis of proposed routing algorithm, friendship quality is chosen.

Primary objective of routing algorithm is to increase the delivery ratio. If the delivery ratio is high, routing algorithm is more efficient and vice versa. Hence for performance analysis of proposed routing algorithm delivery ratio is taken as main parameter.

Friendship Quality Analysis

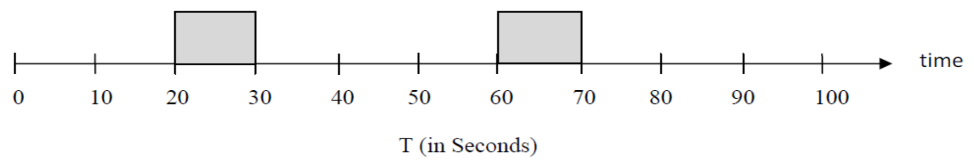
Main objective of this dissertation is accurate analysis of friendship quality, based on which routing algorithm takes decisions about message forwarding. This section gives accurate analysis of friendship qualities based on encounter histories. Figure 17 shows encounter histories of node 1 with five other nodes in the topology.

IJARCCE

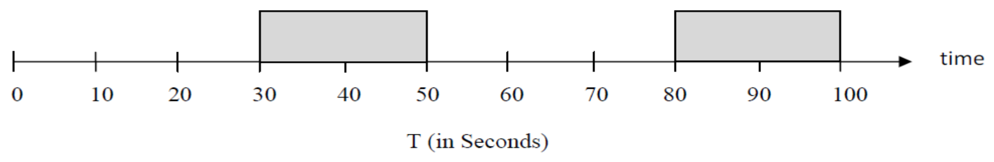
International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 7.39 ⋇ Vol. 11, Issue 1, January 2022

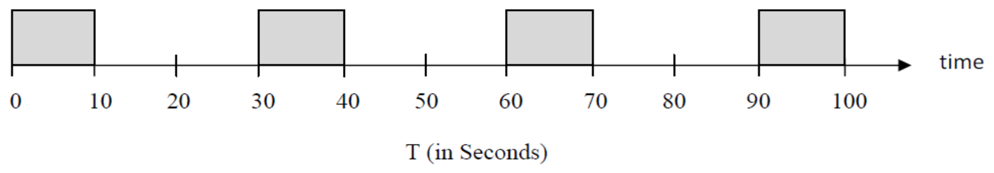
DOI: 10.17148/IJARCCE.2022.11154



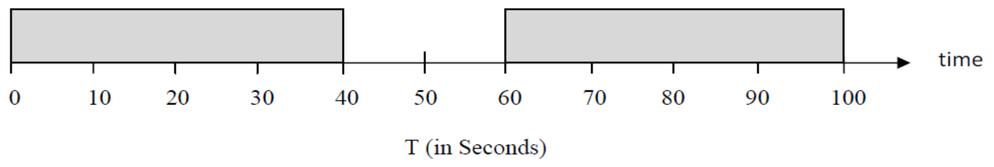
(a)



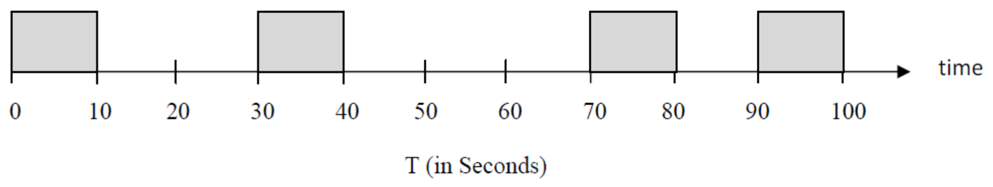
(b)



(c)



(d)



(e)

Fig 17: Encounter history of Node 1 with Node 2(a), Node 3(b), Node 4(c), Node 5(d) and Node 6(e) in the time interval [0,100] Seconds. Shaded boxes show duration of the encounters between nodes.

Considering encounter history of node 1 with 5 other node as shown in Figure 17, SPM and weights are calculated shown in Table 1. Encounter times is, length of time two nodes are in contact over time period T (here 100 seconds).
TABLE 1: NODE 1, SPM AND WEIGHTS

| **Destination** | **Source** | **Encounter Time (Seconds)** | **SPM** | **Weights** |
|---|---|---|---|---|
| 1 | 2 | 20 | 4.82 | 0.21 |
| | 3 | 40 | 3.59 | 0.28 |
| | 4 | 40 | 3.6 | 0.28 |
| | 5 | 80 | 0.33 | 3.03 |
| | 6 | 40 | 6.41 | 0.16 |

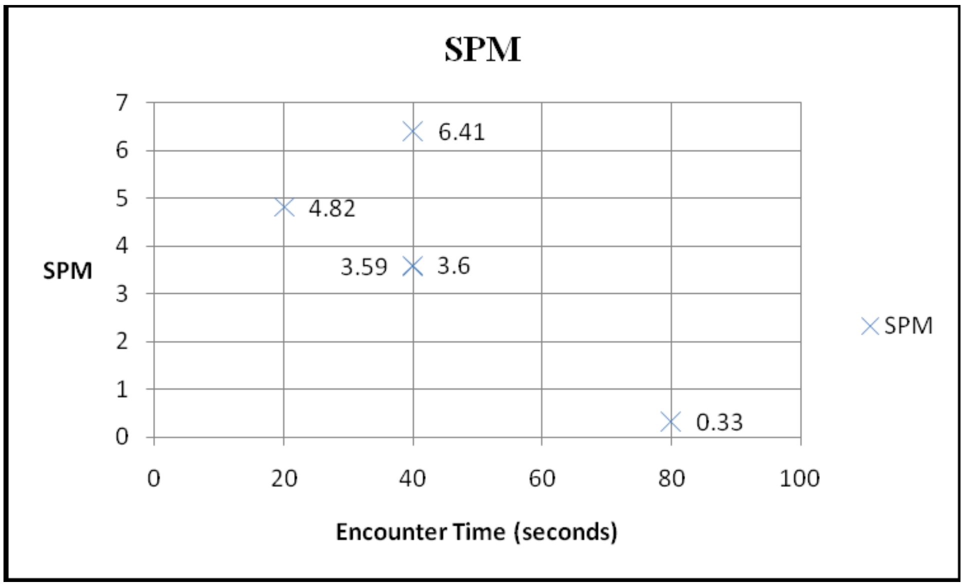


Fig 18: Encounter time versus SPM.

For two nodes to be considered friends of each other, they need to contact frequently and regularly in long-lasting sessions. High frequency refers to average intermeeting time while regularity refers to variance of intermeeting time.
Figure 18 shows SPM values of nodes with different encounter times. Nodes 2's contact time is 20 seconds and hence SPM value is high. Node 3, Node 4 and Node 6 has same contact time. SPM values of Node 3 & 4 are same because both contact regularly and frequently. Comparing with Node 3 & 4, Node 6 is not regular hence even though contact times are same SPM value is high. Node 5 is regular, contact time longer also frequency is high, hence high friendship quality compared with other nodes.
The friendship quality analysis from above section shows that SPM values clearly identify quality of friendship between nodes using the contact histories. Also SPM value will consider three behavioral features of MSN nodes like high frequency, longevity and regularity in finding the quality of friendship.

**Delivery ratio Vs Time analysis**
For delivery ratio analysis, initially Buffer space to store messages nodes receive is sufficiently available, and the bandwidth is high enough to allow the exchange of all messages between node encounter times. These assumptions are reasonable in view of capabilities of today's technology. Table 2 shows delivery ratio (%) after each 20 seconds time interval. For simulation purpose each node sends 30 messages to the random destination. each message longevity is 3 seconds. Frequency is constant for all the nodes.

TABLE 2: DELIVERY RATIO (%)

| **Time(Secs)** | **20** | **40** | **60** | **80** | **100** |
|---|---|---|---|---|---|
| **Run 1** | 16 | 28 | 42 | 60 | 70 |
| **Run 2** | 15 | 26 | 41 | 55 | 68 |
| **Run 3** | 15 | 25 | 43 | 64 | 73 |
| **Run 4** | 14 | 26 | 44 | 60 | 74 |
| **Run 5** | 16 | 25 | 43 | 61 | 69 |

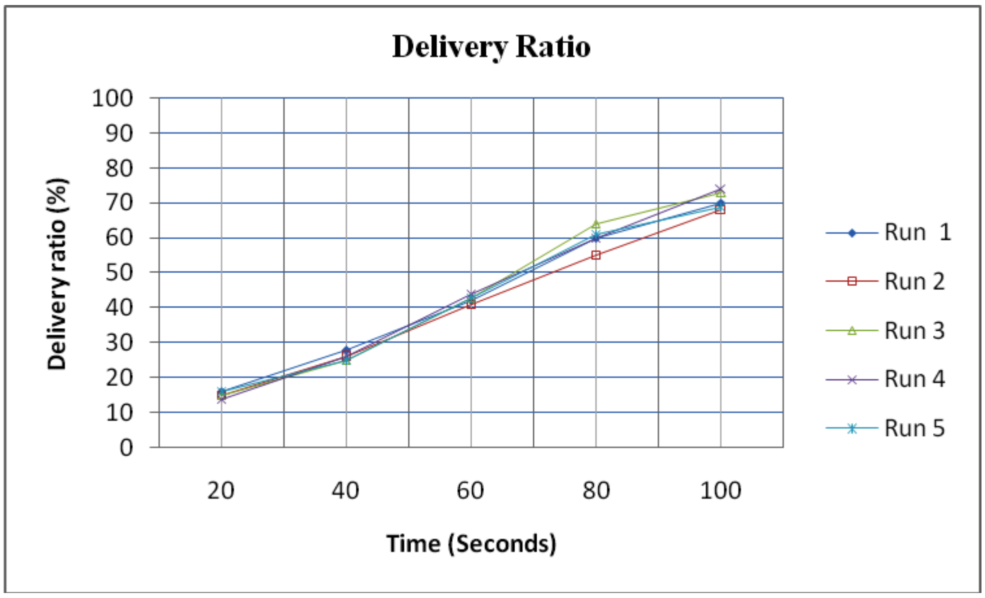


Fig 19: Overall Delivery

Figure 19 shows the percentage delivery of messages, in initial buildup time each node builds its friendship community. Overall, 150 messages are sent from all the 5 nodes in 90 seconds.
Figure 20 shows overall message delivery percentages. Simulation was ran 5 times and the percentage of delivery ratios are shown. Graph shows that average 71% of messages are delivered to the correct destination.

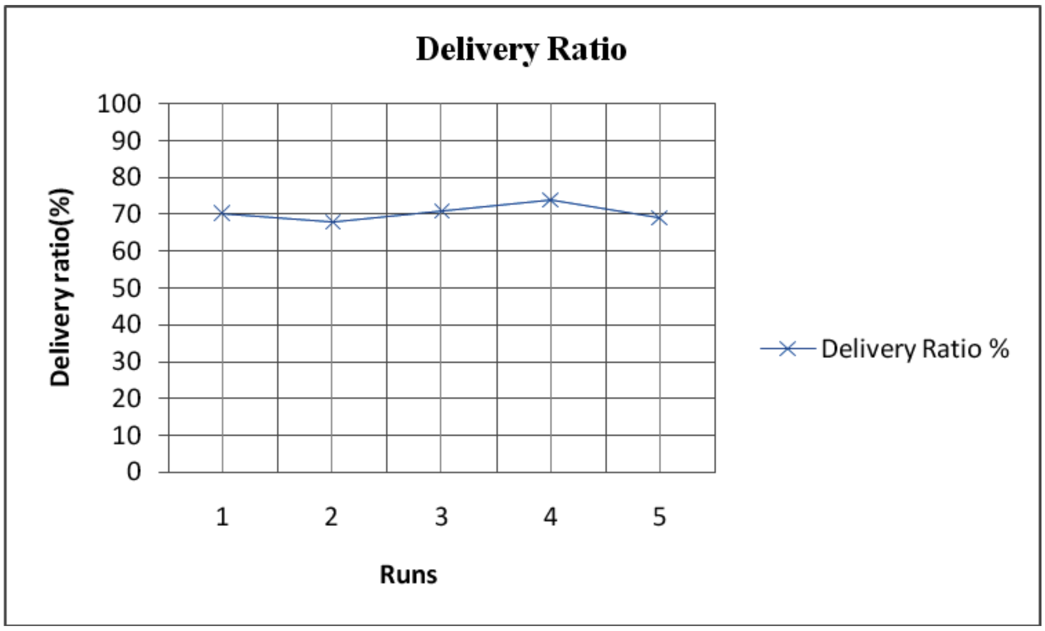


Fig 20: Overall Delivery ratio

**Effect of Time-To-Live on delivery ratio**
Table 3 shows delivery ratios of algorithm at different values of TTL. When a message takes more hops or when node having message will not come in contact with the destination or intermediate node which is friend of destination, message waiting period in the buffer will increase. If overall waiting time of the message exceeds TTL, then it is dropped. If drop rate increases delivery ratio decreases and hence algorithm efficiency will decrease.

Impact Factor 7.39 ⁜ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11154

TABLE 3: DELIVERY RATIO WITH DIFFERENT TTL

| **TTL\Time(Secs)** | **20** | **40** | **60** | **80** | **100** | **No of Messages Dropped (%)** |
|---|---|---|---|---|---|---|
| **TTL 10 Sec** | 10 | 27 | 50 | 54 | 58 | 14 |
| **TTL 20 Sec** | 14 | 28 | 47 | 55 | 65 | 7 |
| **TTL 30 Sec** | 15 | 29 | 43 | 64 | 73 | 2 |

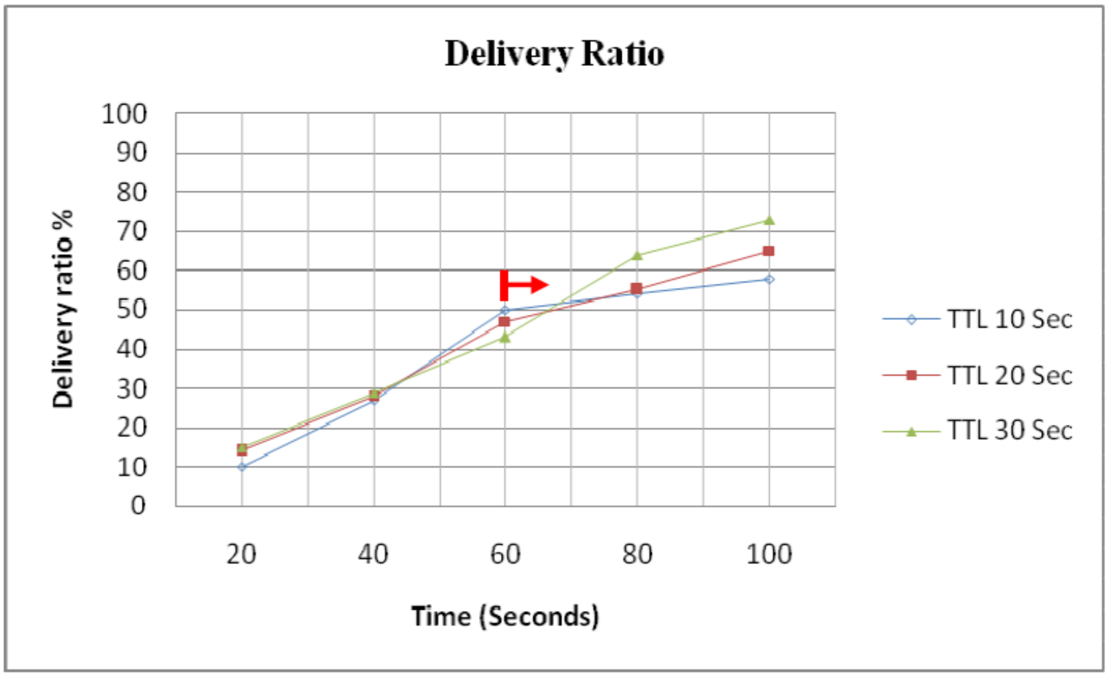


Fig 21: Delivery ratio with different TTL

Figure 21 shows that, as the TTL values increases, delivery ratio also increases. Till the arrow mark in graph, all the three scenarios have similar delivery ratio, its because, till that time no message will exceed TTL. Later drop rate in the scenarios with lesser TTL will increase, hence overall delivery rate of that scenario will be lower compared with higher TTL scenario's. Figure 22 shows the percentage drop in the messages. As the TTL value decreases delivery ratio will decreases.

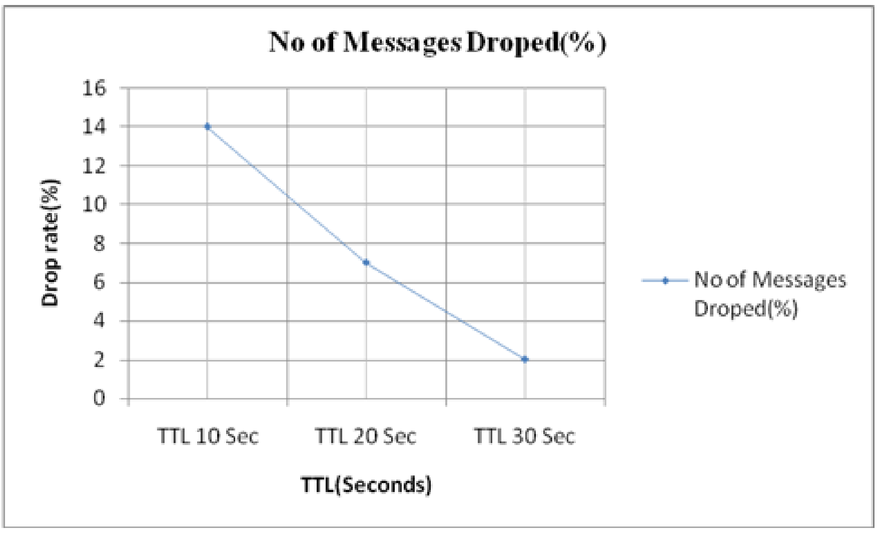


Fig 22: Drop percentage Vs TTL

**Effect of Buffer size on delivery ratio**

Delay tolerant networks are characterized by ‘Store and Forward’. When a node receives a message with destination not self, it is store in the buffer. When size of buffer is full messages are dropped from the node. Table 4 shows effect of buffer size on delivery ratio.

TABLE 4: DELIVERY RATIO WITH DIFFERENT BUFFER SIZE

| **Buff Size\Time(Secs)** | **20** | **40** | **60** | **80** | **100** | **Drop (%)** |
|---|---|---|---|---|---|---|
| **Buffer Size 10** | 10 | 25 | 30 | 51 | 65 | 10 |
| **Buffer Size 20** | 11 | 26 | 40 | 47 | 67 | 4 |
| **Buffer Size 30** | 25 | 37 | 43 | 52 | 71 | 3 |

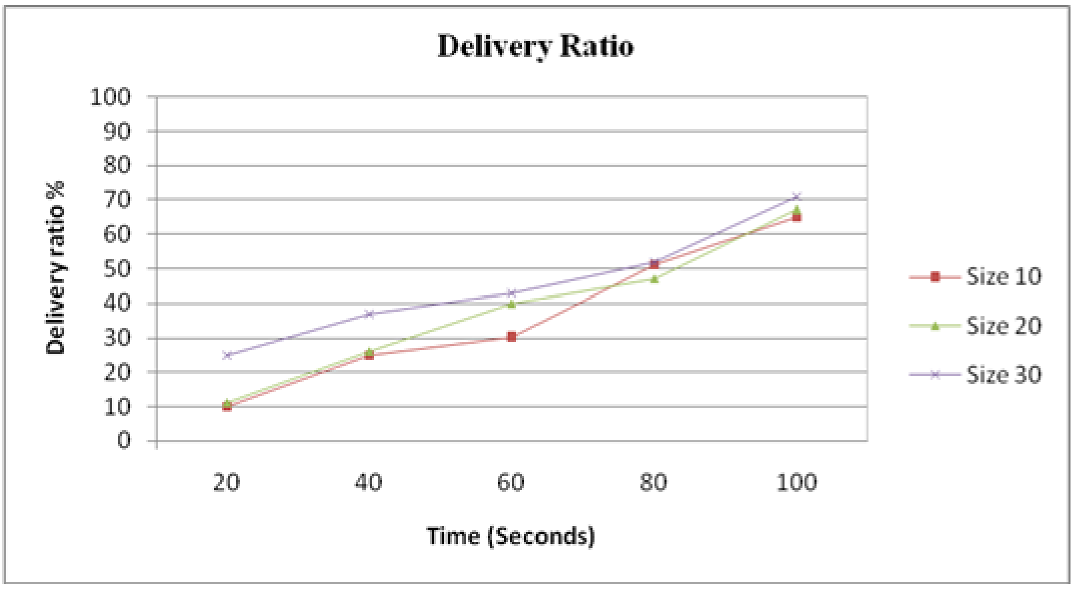


Fig 23: Drop percentage

Figure 23 shows drop percentages of messages when different buffer sizes are chosen. The results show that routing efficiency of the algorithm increases as buffer space increases because messages are not dropped. Initially fewer messages are generated, fewer messages are overflow, and thus more messages could be delivered, increasing routing efficiency.

## IV. CONCLUSION

In this work, routing problem in mobile social networks which are a type of DTNs is studied. The novelty of proposed algorithm is accurate analysis of social network properties of these networks. First, a new metric called SPM is introduced to detect friendship-based node relations accurately. Local community is formed based on this new friendship detection metrics. Then a new routing algorithm is designed in which a node forwards its messages to those nodes that contain the destination node in their friendship communities. While routing, indirect relations between nodes are considered in a novel way making them amenable. The proposed algorithm is evaluated through simulations considering various parameters, which concludes the analysis that, algorithm works efficiently. The only complexity of algorithm lies in nodes exchanging RSPM values with its friends, which is an overhead. In contrast, the other social based algorithms impose a significant control message overhead caused by exchange of the summary vectors during contact times.

## REFERENCES

[1] F. Li and J. Wu, “LocalCom: A Community-Based Epidemic Forwarding Scheme in Disruption-Tolerant Networks,” Proc. IEEE Comm. Soc. Sixth Ann. Conf. Sensor, Mesh, and Ad Hoc Comm. And Networks (SECON), pp. 574-582, 2009.

[2] E. Bulut, Z. Wang, and B.K. Szymanski, “Impact of Social Networks in Delay Tolerant Routing,” Proc. IEEE GLOBECOM, 2009.

[3] A. Vahdat and D. Becker, "Epidemic Routing for Partially Connected Ad Hoc Networks," Technical Report CS-200006, Duke Univ., 2000.
[4] K. Harras, K. Almeroth, and E. Belding-Royer, "Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding Schemes in Sparse Mobile Networks," Proc. Fourth IFIP-TC6 Int'l Conf. Networking Technologies, Services, and Protocols; Performance of Computer and Comm. Networks; Mobile and Wireless Comm. Systems, May 2005.
[5]T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case," IEEE/ACM Trans. Networking, vol. 16, no. 1, pp. 77-90, Feb. 2008.
[6] E. Bulut, Z. Wang, and B. Szymanski, "Cost Effective Multi-Period Spraying for Routing in Delay Tolerant Networks," IEEE/ACM Trans. Networking, vol. 18, no. 5, pp. 1530-1543, Oct. 2010.
[7] A. Lindgren, A. Doria, and O. Schelen, "Probabilistic Routing in Intermittently Connected Networks," ACM SIGMOBILE Mobile Computing and Comm. Rev., vol. 7, no. 3, pp. 19-20, 2003.
[8] E.P.C. Jones, L. Li, and P.A.S. Ward, "Practical Routing in Delay Tolerant Networks," Proc. ACM SIGCOMM Workshop Delay Tolerant Networking (WDTN), 2005.
[9] T. Spyropoulos, K. Psounis, and C.S. Raghavendra, "Efficient Routing in Intermittently Connected Mobile Networks: The Single-Copy Case," IEEE/ACM Trans. Networking, vol. 16, no. 1, pp. 63-76, Feb. 2008.
[10] J. Burgess, B. Gallagher, D. Jensen, and B.N. Levine, "MaxProp: Routing for Vehicle Based Disruption-Tolerant Networks," Proc. IEEE INFOCOM, Apr. 2006.
[11] Y. Wang, S. Jain, M. Martonosi, and K. Fall, "Erasure-Coding Based Routing for Opportunistic Networks," Proc. ACM SIGCOMM Workshop Delay-Tolerant Networking, 2005.
[12] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using Redundancy to Cope with Failures in a Delay Tolerant Network," Proc.ACM SIGCOMM, 2005.
[13] E. Bulut, Z. Wang, and B. Szymanski, "Cost Efficient Erasure Coding Based Routing in Delay Tolerant Networks," Proc. IEEE Int'l Conf. Comm. (ICC), 2010.
[14]Y. Liao, K. Tan, Z. Zhang, and L. Gao, "Estimation Based Erasure Coding Routing in Delay Tolerant Networks," Proc. Int'l Conf. Wireless Comm. and Mobile Computing, July 2006.
[15]L. Chen, C. Yu, T. Sun, Y. Chen, and H. Chu, "A Hybrid Routing Approach for Opportunistic Networks," Proc. ACM SIGCOMM, pp. 213-220, Sept. 2006.
[16]E. Daly and M. Haahr, "Social Network Analysis for Routing in Disconnected Delay Tolerant Manets," Proc. ACM MobiHoc, 2007.
[17] P. Hui, J. Crowcroft, and E. Yoneki, "BUBBLE Rap: Social Based Forwarding in Delay Tolerant Networks," Proc. ACM MobiHoc,2008.
[18] E. Bulut, Z. Wang, and B.K. Szymanski, "Impact of Social Networks in Delay Tolerant Routing," Proc. IEEE GLOBECOM, 2009.
[19] F. Li and J. Wu, "LocalCom: A Community-Based Epidemic Forwarding Scheme in Disruption-Tolerant Networks," Proc. IEEE Comm. Soc. Sixth Ann. Conf. Sensor, Mesh, and Ad Hoc Comm. And Networks (SECON), pp. 574-582, 2009.
[20]J. W. Byers, M. Luby and M. Mitzenmacher, Accessing Multiple Mirror Sites in Parallel: Using Tornado Codes to Speed Up Downloads, in Proceedings of INFOCOM, 1999.
[21]V. Erramilli, M. Crovella, A. Chaintreau, and C. Diot, Delegation forwarding, in Proceedings of ACM MobiHoc, pp. 251260, 2008.
[22]J.M. Pujol, A.L. Toledo, and P. Rodriguez, "Fair Routing in Delay Tolerant Networks," Proc. IEEE INFOCOM, 2009.
[23]H. Dubois-Ferriere, M. Grossglauser, and M. Vetterli, "Age Matters: Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages," Proc. ACM MobiHoc, 2003.
[24] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost communication for rural internet kiosks using mechanical backhaul. In MobiCom '06. ACM, September 2006.

## BIOGRAPHY

Girish Mantha, Assistant Professor from JNN College of Engineering Shivamogga, Karnataka. Completed M.Tech in Computer Science from VTU Belagavi and BE in Information Science from BVBCET Hubli, Karnataka. Research interests include IoT, Security of IoT applications using Block chain, Mobile Social Networks. Undertaken research project Funded by NewGen IEDC, Dept of DST, GOI, Applied and received funding from DST and GOI of about 20 Lakhs for various events. Awarded as Best Achiever award for Drafting 'JNNCE-Green Policy' on the occasion of JNNCE-Sammilana – 2019. Having interest in trekking, Trekked many Himalayan circuits and major south Indian peaks.

Impact Factor 7.39 ⁜ Vol. 11, Issue 1, January 2022

DOI: 10.17148/IJARCCE.2022.11154

Sathyanarayana K B, Assistant Professor from JNN College of Engineering Shivamogga, Karnataka. Completed M.Tech in Networking and Internet Engineering from VTU Belagavi and BE in Computer Science and Engineering from VIT, Bengaluru, Karnataka. Research interests include Machine Learning, Image Processing, IoT. Guided many web application.

Pradeep H K, Associate Professor from JNN College of Engineering Shivamogga, Karnataka. Completed M.Tech in Computer Science from VTU Belagavi and BE in Electronics and Communication Engineering from SJCE, Mysore. Research interests include IoT, Security of IoT, Process automation