Social Interest Based Routing Algorithm for Pocket Switched Network

Inspiring Excellence

By

MAHRIN TASFE
13101094
BINITA SAHA
13101170

Department of Computer Science & Engineering
BRAC UNIVERSITY

Supervised by: DR. AMITABHA CHAKRABARTY

Thesis report submitted to the BRAC University in accordance with the requirements of the degree of BACHELOR IN COMPUTER SCIENCE AND ENGINEERING in the Dept. of Engineering & Computer Science.

Submitted On:
21ST AUGUST, 2017

Abstract

Mobile Ad hoc Network (MANET) and Delay-Tolerant Network (DTN) exhibit a lack of fixed infrastructure and possess no a-priori knowledge of arrival and departure of hand-held devices (such as cell phones, tablets, portable gaming devices etc.). Hence, the nodes are sparsely distributed in the network and they are intermittently connected. For outdoing this, in the Pocket Switched Networking (PSN), we will exploit the device's mobility pattern to transmit messages more rapidly and contribute to the energy conservation and communication efficiency. Therefore, selecting the proper message forwarding algorithm is the foremost factor for efficient message dissemination and resource sharing; which is implemented using the wireless technology (Wi-Fi, Bluetooth). Such kind of method has wide applications in various scenarios such as disaster or war affected areas where regular connectivity methods are not approachable. In this paper, a novel social interest based routing approach called "Gossip" has been proposed which utilizes the social interests of the mobile device carriers to form social profiles for each user and afterwards opportunistically choose the bridging node depending on that to successfully transfer messages.

Dedication and acknowledgements

This thesis would not have been possible without the constant support of our honorable supervisor Amitabha Chakrabarty, PhD. He has directed us to right paths in the toughest times and encouraged us to take risks. Our heartfelt gratitude goes towards his efforts that made us complete this work.

We dedicate this work to our team member and friend Argha Kamal Roy. To our utter misfortune, he took an unexpected and untimely leave from this world. He has been a kind friend and we all who know him used to admire his kind and helpful attitude. This work started because of him. And his efforts and memories have been the force behind the completetion of our work after his early departure.

I, Mahrin Tasfe, would like to show my gratitude to my parents and my friend AKM Nivrito. Also I would like to thank Rupa Barua and Shauvik Shadman for letting us use their works as reference materials.

I, Binita Saha, would like to express my sincere gratitude to my Parents, my sister Puja for all the mental supports. I am also grateful to my other faculties and friends who were by my side in need.

AUTHOR'S DECLARATION

We, hereby declare that this thesis is based on the results found by ourselves. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole or in part, has been previously submitted for any degree.

SIGNATURE OF THE AUTHORS:

....
MAHRIN TASFE
13101094

....

BINITA SAHA
13101170

SIGNATURE OF THE SUPERVISOR:

....
DR. AMITABHA CHAKRABARTY
ASSISTANT PROFESSOR
DEPT. OF COMPUTER SCIENCE, BRAC UNIVERSITY

Table of Contents

Page

List of Tables viii

List of Figures ix

1 Introduction 1

1.1 Motivation . . . 3
1.2 Goals . . . 3
1.3 Problem Statement . . . 3
1.4 Solutions and Methodology . . . 4
1.5 Thesis Contribution . . . 4

2 Literature Review 7

2.1 Related Works . . . 7
2.1.1 Direct Delivery Routing . . . 7
2.1.2 First Contact Routing Protocol . . . 8
2.1.3 Epidemic Routing Protocol . . . 8
2.1.4 Spray and Wait . . . 9
2.1.5 HMSaW . . . 9
2.1.6 Prophet . . . 10
2.1.7 I-Prophet . . . 10
2.1.8 Bubble Rap . . . 11
2.1.9 HCBF . . . 12
2.1.10 HERO . . . 12
2.1.11 Friendship Based Routing . . . 13
2.1.12 Sane . . . 14
2.1.13 Social Circle . . . 14
2.1.14 EPCWH . . . 15
2.1.15 Chit-Chat . . . 16
2.2 Application domain . . . 17
2.3 Open Issues in PSN . . . 18

3 Proposed Routing Algorithm **21**
3.1 Design of our Proposed Algorithm 21
3.1.1 Real Life Example of our proposed algorithm 21
3.1.2 Design of Gossip algorithm 22
3.1.2.1 Social Profile 22
3.1.2.2 Decay Function 23
3.1.2.3 Growth Function 23
3.1.2.4 Time calculation: (From Algorithm 1) 24
3.1.2.5 Real-time Transient Social Relationship (RTSR) Calculation: (From Algorithm 2) 24
3.1.2.6 Routing Decision Calculation: (From Algorithm 3) 24
3.2 Pseudo Code of Gossip 25
3.3 Dataset 26
3.4 Simulation Setup 26
3.4.1 Simulator 27
3.4.2 Node Class 27
3.4.3 Message Class 27
3.4.4 Simulator Class 27
3.4.5 Gossip Class 28
3.4.6 Algorithm Classes 28
3.4.6.1 First Contact 28
3.4.6.2 Bubble Rap 28
3.4.6.3 Chitchat Class 28
3.4.7 Data 29

4 Simulation Results and Analysis **31**
4.1 Setting of Default Experimental Parameters 31
4.2 Results 32
4.2.1 Delivery ratio 32
4.2.2 Drop Rate 33
4.2.3 Message Replication Rate 33
4.2.4 Transmission Cost 34
4.2.5 Average Latency or Delay 35

5 Conclusion & Future Work **39**
5.1 Conclusion 39
5.2 Future Work 39

Bibliography **41**

List of Tables

Table **Page**

3.1 Variables used in Gossip algorithm 22
3.2 Dataset Analysis 26

4.1 List of Parameters and their default values for First Contact and Bubble Rap 31
4.2 List of Parameters and their default values for Chitchat and Gossip 32

List of Figures

Figure **Page**

2.1 Mechanism of Direct Delivery Routing 8
2.2 mechanism of Bubble Rap Routing 11
2.3 Motion Path of the nodes 15
2.4 PSN is one kind of Opportunistic Networking 18
2.5 Store and forward mechanism used by Pocket Switched Network 19

4.1 Delivery Ratio vs. Simulation Length 33
4.2 Drop Rate vs. Simulation Length 34
4.3 Message Replication Rate vs. Simulation Length 34
4.4 Message Transmission Rate vs. Simulation Length 35
4.5 Average Latency vs. Simulation Length 36
4.6 Delivery ratio vs. No of social interests used in Gossip 36
4.7 Drop rate vs. No of social interests used in Gossip 37
4.8 Message replication rate vs. No of social interests used in Gossip 37
4.9 Message transmission Rate vs. No of social interests used in Gossip 38
4.10 Average latency vs. No of social interests used in Gossip 38

Chapter 1

Introduction

In the situation where devices or nodes are sparsely connected in a network, there may not be any direct end to end possible connectivity between two devices or nodes. Since the nodes are generally mobile because of their owners' mobility which are influenced by the geographic and social constrain eventually leading to a social pattern [5]. Hence, the nodes are intermittently connected [36]. To deal with this problem, Delay Tolerant Network (DTN) was developed where messages are passed using the store-and-forward approach in which the forwarding nodes or the devices store the messages until they find opportunistic intermediate hop node. The intermediate node will help the message to reach the destination node or device [11]. But it comes with some added limitations such as limited bandwidth, buffer size, security and energy. The DTN routing protocols can be categorized into three types-Flooding based, Forwarding based and Social based. The Social based DTN is based on the analysis of social patterns, relationships and implications portrayed by the social entities [28]. Pocket switched networking is born from the Social based DTN.

In this technological era, hand-held devices (such as cell phones, tablets, portable gaming devices etc.) can be seen within the hands of almost all the people. People are more connected to each other through wireless technology than ever before [1]. Previously these technological devices were mostly end to end contact based and their performance was scalable [10], [36]. But due to the mobility of the technological devices, they are now facing the challenges such as intermittently connected network, lack of fixed infrastructure and no previous observational data (such as- no former knowledge about the location of destination), frequent information update and causing massive burden on the intermediate or the replay node [18]. For this purpose, Pocket Switched Network (PSN) has been introduced [28]. PSN is self-learning as well as completely decentralized and for this purpose it runs in ad-hoc manner [31]. Neither central infrastructure nor any third party, like Telephone service provider, is needed in PSN. It uses the neighboring mobile devices as intermediate nodes which are selected with the intention of lessening the burden of frequent information updating [5], [15]. Using the intermediate nodes as hop or relay the data is passed to the desired destination mobile device or node [15].

The application domain of PSN is always in on the increase with the enhancement of communication protocols and popularity of the hand held devices [36]. It can be used in the economic (e.g. - broadcasting sales advertisements) and social (e.g. - sharing social interests, communicating with people) sectors even in the military missions (e.g. - battle fields) [22], [36].

1.1 Motivation

Pocket Switched Network (PSN) is one kind of Delay Tolerant Network (DTN). Most of the traditional routing algorithms and protocols in the field of DTN are not able to deliver messages because these routing schemes are merely dependent on the nodes mobility. Not to mention, transmission range of the hand-held devices has reached a limit and after that it cannot be increased because of the technological issues and most of these hand-held mobile devices suffer from discontinuity. Hence, routing algorithms of modern days needs to endure large delays. On the contrary, PSN needs no fixed infrastructure to work. It works though opportunistically using people's behavior and random mobility characteristics to form mobility pattern based on geographical locations, social interests, economy and other properties to transmit messages efficiently and effectively.

In this thesis work, we have proposed a novel routing algorithm called Gossip. It is a social interest based routing algorithm which works in the pocket switched network. It provides superior results in terms of delivery rate, drop rate, message replication overhead and transmission cost than other routing algorithms.

1.2 Goals

The main objective of our thesis was to propose an algorithm inspired by the traditional and modern algorithms previously done by the other researchers and attempt to make a algorithm enhanced in efficiency and performance measuring criteria such as low drop date, increased delivery ratio, low transmission cost and message replication rate and faster routing etc. Our proposed algorithm Gossip is inspired by Chitchat. For performance analysis Gossip was compared with the modern algorithm Chitchat and the other traditional algorithms (Bubble Rap, First contact). It has been able to outrun the previously mentioned algorithms in performance.

1.3 Problem Statement

Pocket switched network mainly consists of a particular number of hand-devices working as relay nodes and mobility of these nodes depend on the mobility of their owners. Hence the mobility of the nodes or the hand-held devices is unpredictable and impulsive. Moreover, in PSN mobile devices used as forwarding nodes or hops use store and forward process for faster delivery of the messages. However employing every encountered node as relay node and flooding them with the data which is inconsistent to their social interest is not an effective decision. Thus under these circumstances, finding an efficient routing algorithm which will be able to utilize other nodes as hops according to their social interests and use their buffer size effectively is a crucial issue without flooding data to all the nodes in the community or the network.

1.4 Solutions and Methodology

For solving the above mentioned challenges connected with pocket switched network, we have tried to provide an efficient routing algorithm called Gossip. It will not flood the data to all the encountered nodes in a community or network rather it will try to choose its forwarding nodes or hops on the basis of similarity among its close friend 's interests. Hence the congestion in the network will decrease and the chances of delivering the messages successfully to the destination node will eventually increase. Moreover, in the literature review section we have given a brief description about the traditional and modern routing protocols of PSN and they have used the concept of contact transmission, flooding. Our proposed algorithm Gossip is a social interest based routing algorithm and its mechanism works by using the probability of communication and social profiles of its users. It is inspired by Chitchat [22] which is also a social profile based routing algorithm. The social profiles of the users are generally generated considering the social interests of frequently encountered nodes. We have assigned a set of social interests to nodes which will eventually form a social profile for each node. The transmission of message is done among the nodes in a particular PSN network though comparing the similarities of their interest and their frequently encountered node's interests. Hence a message with a particular interest has greater chances of reaching an individual node or a group of nodes with similar likings and increasing the delivery ratio and decreasing the latency.

1.5 Thesis Contribution

Numerous researchers have tried to propose many efficient routing protocols and algorithms associated with pocket switched network resolving all the problems related to PSN. We have done a brief literature review illustrating all these routing algorithms both the traditional and the modern ones and provided a novel routing algorithm called Gossip which is supposed to outperform all the mentioned algorithms of literature review in many performance measuring parameters.

Report Outline

- **Chapter 1** is the brief description about our motivation behind this thesis, problem statement and our proposed procedures to overcome those problems and a summary of our contribution and goals.
- **Chapter 2** is the literature review of few traditional and modern routing algorithms related to our topic. We have also briefly described the challenges associated with PSN and its application domain.
- **Chapter 3** shows our proposed algorithm inspired by our literature review and simulation setup.
- **Chapter 4** discusses our simulation results and comparison with other algorithms.
- **Chapter 5** is the closure of our report with a summarization of the whole report and our future works.

CHAPTER 2

Literature Review

In this chapter we have discussed about both the traditional and the modern routing algorithms and protocols. We have also included the challenges in the pocket switched network and the continuous expanding application domain of the PSN network.

2.1 Related Works

Considering the pocket switched network (PSN) does not require any fixed infrastructure and routes massages though choosing people as intermediate nodes or hops for data dissemination and resources sharing; many researchers have proposed many protocols and routing algorithms which will be beneficial to us regarding energy conservation and communication efficiency. Among them we have chosen a few traditional and modern routing protocols and prepared a small literature review.

2.1.1 Direct Delivery Routing

In the direct delivery routing only one copy of the message is available in the network which is forwarded to the destination node though direct communication by the source node [12],[2]. Thus neither the relay node nor the prior knowledge of the network is needed. So minimum bandwidth and network resources are used along with minimum message overhead. Moreover successful message transmission of this protocol relies on the direct contact between the source and destination node, hence this protocol suffers badly from high delivery delays and low delivery rates because of high node encounter uncertainties [2]. The figure- 2.1 , illustrates the working procedure of Direct Contact. Here the source node A has a message for destination node and it will store the message in its buffer until it encounters the destination node.

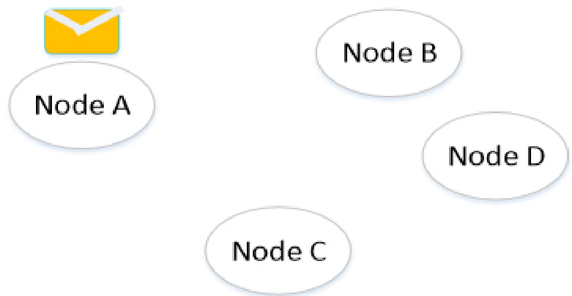

Figure 2.1: Mechanism of Direct Delivery Routing

2.1.2 First Contact Routing Protocol

According to Gamit et al. First Contact Routing Protocol is one of the single copy message forwarding protocols of DTN. Here, the source node or the intermediate node transfers the message to the first node that comes within its contact range for successfully delivering it to the destination node and then deletes the local copy of that message after successful transmission. Thus only a single copy of the message is available in network which ensures minimum network resource usage. However there are chances of losing that single message due to node failures and buffer overflows. Moreover the relay nodes which will forward the messages are selected randomly based on their selection history, a single node is only selected once. That means if a node has forwarded message before, it will not be selected again. Hence sometimes best suitable forwarding nodes are not selected which results in high delivery delays. So this protocol is not able to ensure the successful delivery of message to the destination node which results in low delivery ratio[12].

2.1.3 Epidemic Routing Protocol

Epidemic Routing Protocol is one of the flood based routing protocols used in DTN [32]. This protocol mainly works by replicating the message and flooding out to every possible node; hence eventually reaches the destination node. The source node forwards or "infects" every possible node with the replicated message copy that does not have the copy of the message and the "infected node" follows the forwarding or "node infecting" pattern as well. The core benefit of the epidemic is achieving low latency and highest delivery ratio but by exposing the network with the risk of traffic congestion which may decrease delivery efficiency, transmission power loss and over usage of bandwidth, buffer size [39] and other complications. Epidemic routing protocol is inappropriate for large networks [10].

Several recovery schemes have been proposed in order to minimize the limitations associated with epidemic. For handling the messages which has already reached destination IMMUNE scheme is followed. In this scheme, an "anti-packet" is stored at each node for stopping "reinfection". In a more severe approach, "IMMUNE-TX" scheme is followed where the "anti-packets" are flooded among the nodes to stop further infection [39].

2.1.4 Spray and Wait

In [33] Spyropoulos et al. have proposed Spray and Wait protocol which exploits the speed of epidemic routing [32] as well as the simplicity, economic resource and energy usage of direct transmission . This protocol falls under the category of flood based routing protocols and undifferentiated from other flood based protocols it does not have any priori-knowledge about the network as well as the node mobility [12]. Thus, it works in an epidemic manner but with limited copies of messages in the network.

This protocol works in two phases such as-

- Spray Phase: every message generated at the source node will have L number of copies which will be forwarded to L relay nodes. If a node has more than one copy of message copies and the next forwarding node has no copies of the message, then it will hand over half of its messages to the next forwarding node. Handing over half of the messages process is called Binary Spray and Wait. On the contrary, nodes with single copy of the message will work as Direct Transmission.

- Wait Phase: If the destination node is not found in the spray phase, then each of the relay nodes with message copies will forward the message to only the destination node (Direct Transmission).

2.1.5 HMSaW

Mehta et al. have tried to improvise the Spray and Wait algorithm [33] exploiting human mobility. Their enhanced algorithm estimates the "relay weight" for every destination node using the human walk metrics which is eventually calculates the delivery probability. This modified algorithm is called Human Mobility based Spray and Wait Routing protocol (HMSaW) [23].

- Spray Phase: In HMSaW Mehta et al. have solved the problem associated with choosing an inactive node as a "relay node" and handing-over half of the messages to the relay node lessening the chances of message transmission [33]. In the beginning fixed numbers of copies (L) are created. Each node floods neighbor discovery message (NBR_{dis}) within its communication range periodically to find all the neighboring nodes and then using the NBR_{dis} change degree of connectivity is calculated. Thenceforth Relay Weight is calculated with using the degree of connectivity, flight length parameter and pause time parameter. The message will be forwarded to the relay node with highest Relay Weight within the transmission range otherwise the message will be buffered. Moreover, the message will be deleted from a node either if it has been acknowledged by some relay node or if the buffer size is full.

- Wait Phase: If only one copy of the message is left then the wait phase will start and the message transferred via direct transmission method only to the destination node.

2.1.6 Prophet

Among the probabilistic model-based routing protocols Prophet [21] is popular. It is based on history of node encounters and transitive delivery probabilities. Lindgren et al. have noticed that real users follow a particular mobility pattern; hence the nodes or users who meet regularly have a higher probability to deliver the message to destination. They have used the probability of message delivery not only to choose the forwarding node to reduce congestion in the network but also to lessen the communication overhead. Thus it increases the delivery probability and decreases power consumption [32].

In this protocol, when two nodes are connected they not only share the message but also the delivery predictability information. For choosing the efficient forwarding node, they used a probabilistic metric which is updated after each encounter of nodes. When two nodes have dis-connectivity for a certain time, the probabilistic metric value starts degrading which is shown in equation- 2.1. Here, $P_{(a,b)}$ defines the probability metric of every node a, for each unknown destination b [21], γ^k is the aging constant and k is the number of time unit that have lapsed since the last time the metric was aged.

$$P_{(a,b)} = P_{(a,b)_{old}} \times \gamma^k \tag{2.1}$$

Frequency of the node encounters is also observed to calculate the transitive property of the delivery predictability. In the equation- 2.2 is showing, when node A frequently meets with node B and node B frequently visits node C, then node C is a good hop or message forwarding node for node A.

$$P_{(a,c)} = P_{(a,c)_{old}} + (1 - P_{(a,c)_{old}}) \times P_{(a,b)} \times P_{(b,c)} \times \beta \tag{2.2}$$

Choosing the forwarding node is very challenging since there is no certainty of finding a forwarding node with higher probability within a reasonable time. Moreover, if we are following a threshold probability value for choosing the routing node, there is chance that we might never find a node passing the threshold value [21].

2.1.7 I-Prophet

In [38], Zhang et al. have proposed an Improved Prophet (I-prophet) lessening the limitations associated with the initial Prophet algorithm [21]. During calculating the delivery probability,

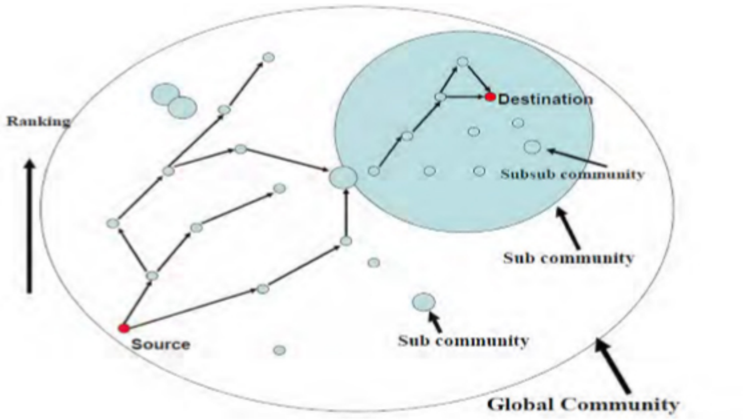

Figure 2.2: mechanism of Bubble Rap Routing

Prophet [21] considered only the probabilities of real-world encounter of the nodes but did not considered the condition of the residing network. I-prophet considers the link or the connection status between the pair of nodes to estimate the encounter nodes probability of reaching the destination node. This links or connections are affected by time and distance between the nodes; hence a link table is maintained. Delivery probability is calculated between node A and node B using equation- 2.3 considering the communication cycle (T_{sum}), communication connection time (T_{con}), connection disconnect time (T_{brk}) where $P_{int} \in [0,1]$ is an initialization constant.

$$P_{(a,b)} = P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}}) \times P_{int} \times (T_{con}/T_{sum}) \tag{2.3}$$

The next relay node is chosen with the highest delivery probability for transferring the message to the destination node.

2.1.8 Bubble Rap

Numerous history based and social network based protocols have been proposed and Bubble Rap falls under this category. It has been created combining the concepts of message forwarding decision based on social network and human community forming due to social movement patterns and human behavior [5], [32].

Bubble Rap uses the k-Clique method for centrally detecting and uniquely identifying each community [10], [17]. Since k-clique method is designed for binary graphs and one person can belong to multiple communities, thus defining the threshold for edge of the contact graphs is essential. Moreover, nodes in each community contain two metrics- local ranking and the global ranking for making message forwarding decisions. The local ranking signifies that particular node's popularity in its own community and the global ranking shows each node's popularity in the whole network [32]. If the source and the destination nodes are in the same community, then it checks the local ranking of the encounter nodes. If the encounter node has higher local ranking which means it has higher chances of reaching the destination node and forwarding the message. On the contrary, for the destination node residing in a separate community, the message is forwarded to encountered node with the higher global ranking [5], [17]. In the figure- 2.2 data forwarding strategies of Bubble Rap have been demonstrated. Here the red circles represent the source node and destination node and the large blue circle shows the local community. The small blue circles are the nodes working as relay nodes for successful transmission of data to the destination node.

2.1.9 HCBF

In [31] Rasul et al. tried to upgrade their Community-Based-Forwarding (CBF) algorithm by improving the intra-community routing of nodes using a new proposed metric Unique Interactions (UI). UI helps to select social but more diverse nodes which are more suitable for intra-community routing. Researchers of [31]'s reorganized algorithm is called Hybrid Community-Based Forwarding (HCBF).

Initial CBF algorithm which is a community based algorithm used community and popularity metrics to avoid the usage of routing tables, suffered from increased delivery latency. For solving this problem they defined two ways. One is to improve the community making approach and recognizing more composite community structures. Another is to enhance the intra-community routing. HCBF aims to improvise the intra-community routing.

Message forwarding approach of the algorithm: If the relay node and the destination are in the same community then Local Probability (LP) is used for forwarding the message to the destination node. If the encountered node is neither from the destination nor the relay node's community in that case succeeding relay node is selected on the basis of maximum Community Betweenness Count (CBC) among the encountered nodes. On the other hand, if both the relay and the encounter nodes are not part of a community belonging to the destination and the relay node; therefore maximum Nodal Contribution Factor (NCF) with the destination community is used for deciding the next forwarding node [31].

2.1.10 HERO

Huang et al. have exploited the concept of home node which was derived from the spatial regularity of human mobility meaning humans visits few places frequently and those are called home. Using this basis they have proposed two HomE based Relay selectiOn (HERO) algorithms in [15] such as basic HERO algorithm and enhanced HERO algorithm. These algorithms need only the relay nodes to pass the messages therefore they only need local information and do not have necessity of any global routing information which eventually results in low communication overhead and minimum buffer size. Moreover, since most of the nodes in the home or local are comparatively stable hence it requires less information updates. Because of having very simple relay node selection approach its computational overhead is very low as well.

• Basic HERO: This is one of the single copy and single hop algorithms of PSN. If the source node cannot transfer the message directly to the destination node then a relay node will be chosen to act a home node which will store the data until it has first-hand communication with the destination node.

• Enhanced HERO: This algorithm is a single copy schemes similar to the basic HERO but it is enhanced with the concept of multiple relays. It helps to find more effective and efficient relays nodes with greater possibility of reaching the destination node. Here the intensity of node vising zones or homes is used for determining relay nodes. In case of direct contact failure between the source node and the destination node; the source node will chose a home node as a relay node among its neighboring nodes having the highest visiting intensity and handover the message to it. Then the relay node will forward the message to other encountered home nodes among the neighboring nodes having higher visiting intensity than itself. Thereupon the newly selected node will work as a relay node to forward the message to destination node and this process of choosing optimal relay nodes will continue until all messages reach destination.

In large scale sparse networks HERO algorithms suffer from slow start phase which results in low data propagation and eventually leading to inefficacious data forwarding [15].

2.1.11 Friendship Based Routing

In [7], Bulut et al. have used the quality of the friendship to determine the close friends (forwarding nodes) within various friendship communities depending on the period of the day. According to them high frequency, longevity, regularity are the behavioral features of the close friendship. Thus, using these three features they have proposed two metrics which helped them to understand the strength of friendship between two nodes and those metrics are social pressure metric (SPM) and conditional SPM (CSPM). Moreover, similar to the human friendship, friendship between the nodes show different variations as well and there are of two types such as direct friend and close indirect friends. Direct friends are the nodes that have direct connection or communication with the other node and this kind of friendship could be determined from their contact history which will eventually help to calculate PSM. On the other hand, close indirect friends are the nodes that have contact frequency though mutual friends and CSPM is used to calculate their friendship link quality. With the both SPM and CSMP a node can form its friendship community using equation- 2.4 where $w_{i,j,k} = 1/(SPM_{i,j} + CSPM_{j,k|i})$. These communities are the temporal friendship communities which demonstrate the differentiation in the strength of friendships in the diverse time periods [7].

$$
\begin{aligned}
F_i = \{j \mid w_{i,j} > \tau \textbf{ and } i \neq j\} \quad \cup \\
\{k \mid w_{i,j,k} > \tau \textbf{ and } w_{i,j} > \tau \textbf{ and } i \neq j \neq k\}
\end{aligned}
\tag{2.4}
$$

Message Forwarding Strategy: If a node A wants to forward a message destined to a close indirect friend C and their mutual friend is node B. Then for node B to work as an intermediate node, it needs to be in the same friendship community with the destination node C and needs have stronger friendship with node B than node A has. On the contrary, if the node B is not in the same friendship community it cannot work as a forwarding node [7].

2.1.12 Sane

The SANE (Social Aware NEtworking) is a social-aware, stateless routing protocol which exploits the characteristics of mobility of the hand held devices and social relationship among their owners with minimum storage requirements. Hence, they have adopted the store-and-forward method of communication in this protocol [24].

Sane provides two communication services such as Unicast and Interest-cast. All the messages forwarded have a target interest profile, $N_{replicas}$ and TTL in their heads. During the Unicast, it chooses the forwarding nodes showing similar interest profile with the destination. It controls its buffer size and number of replicas with a threshold value. If $N_{replicas} = \infty$ it works as Epidemic [32], [25]; otherwise as Spay and Wait [25]. On the other hand, in the interest cast, it uses the interest profile which is sometimes known as message relevance profile to forward messages.

Sane have been able to show better performance than other competitors [25]. Thus, it has been used as the basis for several proposed protocols for the pocket switched network such as Chit-Chat [22], Social Circle [10].

2.1.13 Social Circle

M. Fida et al. have provided a dynamic and distributed clustered based routing protocol exploiting the social patterns and social relationships [10]. According to [10], human mobility is very predictable which gradually leads to a social pattern. People tend to communicate with friends frequently than other people with whom they share strong relationship and rarely or never communicate with unknown people. They frequently visit few specific locations (e.g. - students attending classes at the university following the same routine every week, spending a specific amount of time in the office) which shows their mobility patterns with time. Figure- 2.3 is showing the mobility patterns of nodes. Hence, the Social Circle Protocol has been created with the idea of human social circles and their communicating nature.

In the Social Circle, they have maintained a Temporal Community Table which keeps track of the human mobility and their timings, a Friend List showing regular communicating or connecting neighbors or friends and an Acquaintance List observing the nodes known from neighboring nodes. Here, the acquaintances nodes are used for forwarding messages in neighboring community or distant destination node. For determining the strongest acquaintance set for forwarding the data to distant node Sociability Table (ST) is created which is built with the number of acquaintances of each friend. Moreover, for the early delivery of messages they have also given a

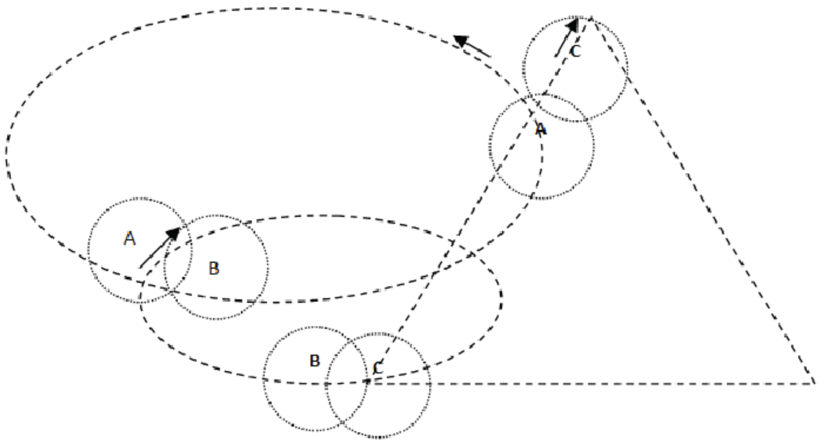

Figure 2.3: Motion Path of the nodes

solution by choosing a friend node or neighboring node from the Temporal Community Table to speed up the delivery process.

The message is routed using the inter-community and intra-community routing. During the inter-community routing, if the destination is found in the friend list it is forwarded directly depending on the desirability of early delivery. On the contrary, for the intra-community routing, if the destination is unknown, then the message is forwarded with the help of Scalability Table; since we need to transfer the message to maximum number of acquaintances with a optimism to find the destination node. On the other hand, if the destination is known, then the message is flooded to corresponding friends node generating multiple copies of the source message.

Social circle has not been able to prove their competency with the disjoint clustered networks. They have used an overlapping clustered network to demonstrate that their proposed algorithm is better than Prophet [21] in all the performance measurements and it is also showing slightly better performance than the Bubble-Rap [17] and Simbet [9].

2.1.14 EPCWH

E. Wang et al. have proposed an Energy efficient Phone-to-phone Communication method based on Wi-Fi Hotspot (EPCWH) in [36] aiming a find a solution which achieves both energy conservation and communication efficiency by scheduling the phone's switching between hotspot mode and client mode in the Pocket Switched Network(PSN). Moreover two phones in opposite mode can establish connection whenever they are in the communication range and form a pocket switched network; on the other hand, two phones in the identical mode cannot establish connection. They have used EPCWH in case of multiples messages and single messages communication under different problem scenarios.

When a phone is in hotspot mode its energy consumption is high and it will be passing its messages using the epidemic routing protocol and a random way point mobility pattern. Thus it can only be in the hotspot mode for a short period of time and needs to have a tradeoff between the energy consumption and the message dissemination. According to them EPCWH intends to maximize the message dissemination scope within the limited energy consumption [36] .

2.1.15 Chit-Chat

This message routing algorithm or the procedure has been proposed with the primary concepts of the SANE (Social Aware NEtworking) algorithm [25] where messages are transferred exploiting the concepts of PSN (i.e. - social behaviors, social patterns, stateless interest-cast and others). D. McGeehan et al. have tried to improve the SANE algorithm by adding concepts of growth and decay which has improved the performance to a noticeable level minimizing the challenges faced due to the sparsity of the nodes in the real world PSN environment [22], [25].

In this message routing approach [22], a social profile (SP) is maintained for each user consisting of a set of social interests (SI) where each SI represents the keywords of each interests with a unique ID. For ensuring efficient message forwarding decisions of a node transient social relationship (RTSR) is used. It keeps the weight of each social interest provided by the social profile of an individual user or node. It helps the nodes to make the forwarding decisions and update the weight of multi-hop relationships using the decay model and growth model. The weight of TSR starts degrading with the increase of dis-connectivity time with the nodes of similar interest (i.e. Equation 2.5). On the contrary, the weight starts to increase if the nodes stays connected with a node of similar interest for a longer time (i.e. Equation 2.6). In the initial phase, when two nodes comes within the communication range they invoke the RTSR module. Then they automatically exchange the RTSR and adjust or update it in the previous mentioned ways. Afterwards they invoke the message forwarding algorithm to route the messages or interests to their destination based on the analysis of the interest of the connected nodes.

$$w_u(SID_i,t_s) = \begin{cases} \frac{w_u(SID_i,t_{d,i})}{\beta\cdot(t_s-t_{d,i})} & \text{if } SID_i \notin SP_u \\ \frac{(w_u(SID_i,t_{d,i})-0.5)}{\beta\cdot(t_s-t_{d,i})} + 0.5 & \text{if } SID_i \in SP_u \end{cases} \tag{2.5}$$

$$w_u(SID_i,t_c) = min\{1, w_u(SID_i,t_s) + \Delta\} \tag{2.6}$$

$$\Delta = \sum_{v \in v} \frac{w_u(SID_i,t_s)\cdot(t_c - t_s)}{\psi_{i,u,v}} \tag{2.7}$$

The chitchat has considered the problems associated with the sparsely connected nodes and networks. They have tried to provide a solution of the sparse nodes with RTSR by not relying on the communication range which has eventually helped it to achieve 2nd highest delivery ratio after the Epidemic [22], [39]. Since increasing the communication range in sparse network is challenging and enhancing the transmission range is technically challenging. Moreover, it has also minimized the communication overhead using the efficient message forwarding algorithm.

After being connected to the users or the nodes of same interests, it forwards the messages. So it does not need to store the file for longer time which helps it to gain a shortest buffer time for each node [22].

2.2 Application domain

Usage of the hand-held devices including cell phones, tablets, portable gaming devices etc. is ever on rise and it would reach 982 million by 2015 was predicted by the reports of International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker [36], [14]. Keeping in mind, PSN does not require any fixed infra-structure; hence the intermittently connectivity of the devices is not considered as a liability but as an opportunity. Its mechanism works through making opportunistic routing decisions considering the geographic and the social constraints as well as exploiting the human mobility pattern to communicate with other PSN enables devices [10], [5], [36]. Due to not requiring any central communication tower or fixed infrastructure, the applications of PSN are very diverse. It can be used for disaster management, deep space communication, military missions, commercial advertisements, remote communication and many more [22], [3], [4], [2], [13]. In the figure-2.4 illustrates the main structure of Opportunistic Network. Here the nodes can start collaboration of data after discovering the neighboring nodes and among them the most dedicated node works as a sprinkler and shares as well as collects information though data sharing protocol.

Firstly, since almost all of the hand-held devices are battery powered and right after the incident of any kind of natural disasters such as earthquakes, tsunami, cyclone etc. that affected area is prone to suffer from power and infrastructure disruption resulting in no cellular network. PSN is the most beneficial in this kind of situation for managing communication among people and finding survivors [2]. In [34] Uddin et al. have proposed a multi-copy routing protocol for post-disaster operations. It is a disaster-response application which works by making a pattern out of the movements and communication among the rescue workers, volunteers and survivors.

Secondly, PSN can be used for collecting various kinds of data for urban sensing such as measuring traffic, traffic accidents, monitoring user behavior etc. It is a very inexpensive way of acquiring a large range of data [2]. Moreover, data collected though PSN can be used for harnessing terrorist activities by locating their positions though patterned movements in remote areas including dense forests, deserts etc. [32].

Thirdly, PSN helps to form social groups by utilizing the social profiles, relationships and their communication pattern. E-SmallTalker [37], MobiClique [29], ONEChat [8] are social profile based PSN applications. In addition, both the location based data collected for urban sensing and social relations among users help us in modern day commercial advertisements by transmitting sale advertisements to customers according to their specific interest and interest of their acquaintances [22], [2].

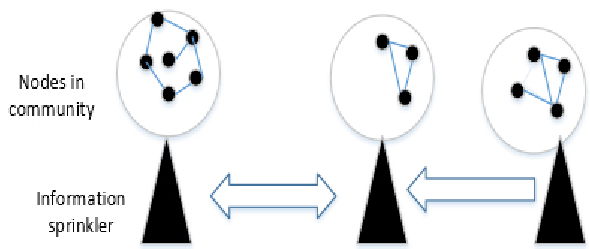

Figure 2.4: PSN is one kind of Opportunistic Networking

Again, as a matter of fact, according the 2015 report of the ITU, over 80% families of the developed countries have access to the internet; on the contrary only 34% of the families in developing countries. PSN can use the information of social relationships and internet to transmit information among users and act as an alternative way of communication.

Not to mention, PSN can be used by the soldiers in the battle field when the infrastructure is damaged resulting in no cellular network and internet. It can be used as a way of communication and collecting essential information about the war zone; for instance position of enemies, the mine-fields, location of the ammunition reserve are transmitted using PSN [22], [3].

Lastly, in space data transmission suffers significantly from intermittent connectivity, long propagation delays and high drop date; PSN which a is special kind of Delay Tolerant Networking (DTN) is suitable for deep-space communication for instance the satellites orbiting the earth in the space [19]. Furthermore since most of scientist community have a preference of working with un-processed, undistorted data for getting more accurate results, it can be beneficial to them as a source of raw data for predicting various events such as weather or environment, natural hazards, global air and water quantity, efficient management of natural resources. Related to this, a network concept of sensor-web built with the satellites orbiting the earth is well mentionable [30].

2.3 Open Issues in PSN

Pocket Switching Network is a unique kind of Delay Tolerant Network (DTN). The number of cell phone or the hand held device users is always on rise according to the reports mentioned in [14]. However, considering DTN faces enormous problems in sparse network, PSN also have to endure few difficulties.

Firstly, for transmitting messages in the PSN, we need to keep the wireless technology (Bluetooth, Wi-Fi) uninterrupted which will eventually lead to fast energy consumption. In addition to this, if excessive numbers of hand-held devices are present in a particular area or

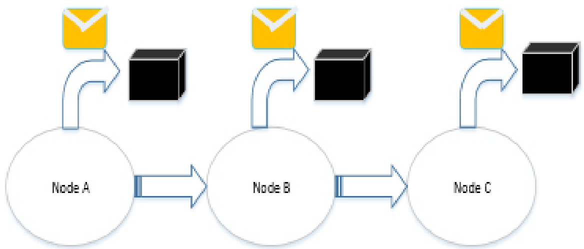

Figure 2.5: Store and forward mechanism used by Pocket Switched Network

network, then the energy consumption gets accelerated which is not beneficial to the users. Hence we need to find a trade-off between the number of data transmission among the cell-phone devices available in a particular community and optimization of energy consumption [35].

Again, PSN needs to utilize the cell-phones available in a network to forward data among the users. But as a human nature, people tend to be selfish; and their cell-phone devices working as hops not willing to forward data apart from their own desired destinations may lead to an increase in the drop rate. Because from [32] we know that PSN uses store and forward strategy for effective data dissemination. Therefore, more mechanisms should be proposed for minimizing the selfishness of nodes regarding data transmission. In the figure-2.5 shows a node storing data and forwarding it to another hop node for successful message transmission. Here the circles represents an individual node and black box shows limited storage capacity of each node.

On the contrary, some hops may work as malicious relays resulting in disrupted data [32], [16]. These malicious activities of nodes may include redirection, impersonation, eavesdropping, fabrication, denial of service, poisoning etc. Hence, maintaining security in PSN is of crucial importance and regarding this many policies have been introduced as well as adapted such as distributed naming, authentication, trust, reputation systems, incentive to cooperate and others. For authentication and trust issues assistance from a third party can be used for justifying how good a node is as a forwarder and the degree of trustworthiness.

Thirdly, data forwarding can also be a foremost issue for PSN which is done both in local and global manner. The main intention behind using other nodes as carriers or forwarders is to transmit data to the destination as soon as possible. So, choosing the proper hop node is of great concern. Local forwarding is done though intermittent and mobility based connectivity. When global connectivity is available, global forwarding is done. For this reason, a replicated message copy is kept for the chances of direct encounter to the destination node [16].

Fourthly, PSN mainly works exploiting the human mobility pattern which is at times unpredictable and irregular. So temporal paths between the source to destination is searched and finding an efficient temporal path is challenging [26].

Moreover, increased replication of messages in the network not only improves the message delivery probability but also decreases the delivery time. However, higher number of message copies in the network increases congestion; eventually leading to increase is the drop rate. Therefore it creates a conflict and further mechanisms for resolving this issues needs to be introduced [2].

Lastly, even though PSN enabled hand-held devices have inadequate storage capacity and limited energy, they constantly need to store and forward data for transmission. So researchers need to work for network scheduling and energy conservation methodologies [16].

CHAPTER 3

PROPOSED ROUTING ALGORITHM

In this chapter, we have given a brief description our proposed routing algorithm and for simulating this routing algorithm we have created a custom simulator. This chapter also includes the information about the classes and methods related to our proposed algorithm and its custom simulator and the environment for working our algorithm.

3.1 Design of our Proposed Algorithm

For comprehending the design of our proposed Gossip algorithm, understanding the concept is very important. For that reason we have given a real life example which will help us to comprehend the design of Gossip.

3.1.1 Real Life Example of our proposed algorithm

Our proposed algorithm Gossip is a social interest based algorithm. We have considered each node to have few social interests such as photography, painting, shopping and many more. Since we know humans with the same geological and social interests tend to go to the same place more often and this kind of movements eventually form a human mobility pattern. Nodes having same interests have a greater chance of delivering a particular message to its destination within a very short time than others. Hence, it utilizes the fact that human behavior generally follows a pattern. For example, a student will follow a routine of the classes and usually ends up meeting the similar people on a certain day. Exploiting that nature we can say, certain messages can be sent to the destination faster. Imagine a student of computer science interested in Photography and Food. Each week he attends multiple classes, but each Thursday he meets two people with having the interest on Photography and Food respectively. The base algorithm of ours adjusts

Table 3.1: Variables used in Gossip algorithm

Used Variables	Description of the variables
u	User or node executing the algorithm
v	Neighbors within the communication range
v_k	Neighbors of u within communication range and their buffer are not full
$t_{c,n}$	Current time-stamp of algorithm execution
$t_{s,n}$	When the user u enters the communication range
$t_{d,i,n}$	Last time-stamp when connected with the neighbor of same interest(SID_i)
$\psi_{i,u,v}$	Growth dampening factor
β	Rate of decay
SID_i	Unique i th social interest id (Described in 3.1.2.1)
SP_u	Social profile of user u (Described in 3.1.2.1)

according to the delay of meeting people of common interest. If that is used, when the student meets the same people on the next Thursday the weight of the interest will decrease so much that if the student has any message containing the interests, it will not be passed. So in order to utilize the pattern and enforce the weight of the repeated meetings to be higher, we distributed the weight based on the day of the week. Now if the same student meets the same people on Thursday the weights will be healthy enough to transfer the messages. So it is just simulating that the same student is meeting the same people on the next day, where in reality they are meeting after one week.

3.1.2 Design of Gossip algorithm

$w_u(SID_i, t_{s,n})$ is the weight of the SID_i on the time $t_{s,n}$ on a certain day n of a week for node u. This is calculated by using the previous weight of the SID_i and the time difference of meeting $(t_{s,n} - t_{d,i,n})$ where $t_{s,n}$ is the time when a node u enters the communication range and $t_{d,i,n}$ is the latest time-stamp of connectivity with node of same social interest.

$$w_u(SID_i, t_{s,n}) = \begin{cases} \dfrac{w_u(SID_i, t_{d,i,n})}{\beta \cdot (t_{s,n} - t_{d,i,n})} & \text{if } SID_i \notin SP_u \\ \dfrac{(w_u(SID_i, t_{d,i,n}) - 0.5)}{\beta \cdot (t_{s,n} - t_{d,i,n})} + 0.5 & \text{if } SID_i \in SP_u \end{cases} \tag{3.1}$$

$$w_u(SID_i, t_{c,n}) = min\{1, w_u(SID_i, t_{s,n}) + \Delta\} \tag{3.2}$$

$$\Delta = \sum_{v \in V} \frac{w_v(SID_i, t_{s,n}) \cdot (t_{c,n} - t_{s,n})}{\psi_{i,u,v}} \tag{3.3}$$

3.1.2.1 Social Profile

Social profile of a user is characterized as SP_u. Each user has a social profile which consists of that user or nodes social interests (SID_i). E.g. $SP_u(SID_1,\ SID_2\ ,\ \dots,\ SID_n)$. TSR_u of a user u consists of social interests and weight of those corresponding social interests. E.g. $TSR_u(SID_i, w(SID_i, t))$. For example, X is a user and her social interests are photography and cooking. If photography has SID_i of 001 and cooking has SID_i of 002, then her social profile is $SP_X(001, 002)$ and her initial TSR would be $TSR_X((001, 0.5), (002, 0.5))$.

3.1.2.2 Decay Function

β acts as a dampening factor for the decay. Then the decay factor becomes $1/\left(\beta \cdot (t_{s,n} - t_{d,i,n})\right)$. When the SID_i belongs to the Social Profile of u (SP_u), we subtract 0.5 from the previous weight, while adding 0.5 after the decay factor is applied so that the value never goes below 0.5. When the SID_i does not belong to the Social Profile of u (SP_u), we calculate the weight by diving previous weight with the decay factor.

3.1.2.3 Growth Function

The weight of the SID_i is updated using the function-3.2. Here the weight is first adjusted with a growth factor Δ. Then we take minimum between the value 1 and the adjusted value of the weight, making sure the value never goes beyond 1. The dampening factor is $\psi_{i,u,v}$. Its value is determined on the basis of whether a particular user node, u and its encountered node, v have a particular social interest in his social profile or not.

- $\psi_{i,u,v} = 1$: When both of the users u and v have the social interest(SID_i) in their social profiles(SP_u). So it is their direct interest.

- $\psi_{i,u,v} = 2$: When user u have the SID_i in his social profile (direct interest) but user v has obtained it from other encountered users or nodes though TSR.

- $\psi_{i,u,v} = 3$: When user v have the SID_iin his social profile (direct interest) but user u has obtained it from other encountered users or nodes though TSR.

- $\psi_{i,u,v} = 4$: When both of the users u and v do not have SID_i in their social profiles. But they have received the SID_i though TSR of other encountered nodes. Hence they still have possibility of meeting people with the same SID_i.

- $\psi_{i,u,v} = 5$: When user u have a weightless TSR. Then it will include any TSR from user v even if it has a very low weight.

- $\psi_{i,u,v} = 6$: When the user u does not have the SID_i but the user v has the SID_i from its TSR and its weight is not zero. Hence it has the lowest chance of delivering the message to the destination and has the highest growth dampening factor which is 6.

3.1.2.4 Time calculation: (From Algorithm 1)

In Gossip algorithm, the time difference is calculated as if a day has been passed, because the weight of a particular social interest is specific to the day of the week. So even if the calculation is happening one week apart, the actual time difference is calculated as a day apart. The time differences increase according to weeks. If it is two weeks apart, the calculation will be two days apart. Again, if it is three weeks, it will calculate three days and so forth. The calculation is done using algorithm 1. We first calculate how many days have passed, then using those parameters we decide how many same days have passed.

3.1.2.5 Real-time Transient Social Relationship (RTSR) Calculation: (From Algorithm 2)

In this algorithm, weight of a social interest SID_i for a user u is calculated when it has $v_{t_{c,n}}$ connected neighboring nodes within the communication range. When two users (u and v) or nodes are within the communication range of each other they first check if they are still connected to each other, then they will not go through any decay, otherwise weight is manipulated using the decay function-3.1. Following the exchange of TSRs between them, growth of the weight of SID_i is calculated using the function-3.2 among all the received TSRs.

3.1.2.6 Routing Decision Calculation: (From Algorithm 3)

After the exchange and the adjustment of the TSRs, message forwarding is done. User u is the node willing to transfer message and v is the neighboring nodes within communication range among them v_k are the nodes who do not have buffer full. User u will try to forward the message though unicast to any of the nodes among v_k, if they match with the message destination. Otherwise, user u will calculate the sum of the weights of the weights of its own TSR and compare it with the sum of the weights of the neighboring nodes (each neighbor at a time). If the neighboring node v_k have greater sum of weights then user u would forward the message to node v_k.

3.2 Pseudo Code of Gossip

Algorithm 1 Pseudocode to find the total duration elapsed after the last connection

Determine n ▷ Current day of the week
get $t_{s,n}$ ▷ Most recent connection time
Time difference, $h = t_{s,n} - t_{d,i,n}$ ▷ Calculating the total seconds elapsed
▷ including other weekdays
Total days elapsed, $TotalDays = h/86400$ ▷ In integer
Total weeks, $TotalWeeks = (TotalDays/7)$
▷ How many Mondays (the current weekday) has elapsed
Adjusted time, $TotalTimeElapsed = (TotalDays_b \times 86400) + (h \bmod 86400)$
return $TotalTimeElapsed$

Algorithm 2 Real-time Transient Social Relationship Modeling for Gossip: A new connection is established between node u and nodes $v_{t_{c,n}} = v_1, \ldots, v_k$ *at time* $t_{c,n}$.

procedure CONNECT($u, v_{t_{c,n}}$) ▷ Decay of TSRs
    for *each* $SID_i \in SP_u$ **do**
        if $t_{d,i,n} == t_{c,n}$ **then** ▷ No decay to $w_u(SID_i, t_{c,n})$
            $w_u(SID_i, t_{c,n}) = w_u(SID_i, t_{d,i,n})$
        else if $SID_i \in SP_u$ **then**
            $w_u(SID_i, t_{c,n}) = \frac{(w_u(SID_i, t_{d,i,n}) - 0.5)}{\beta \cdot (t_{s,n} - t_{d,i,n})} + 0.5$
        else
            $w_u(SID_i, t_{c,n}) = \frac{(w_u(SID_i, t_{d,i,n}))}{\beta \cdot (t_{s,n} - t_{d,i,n})}$
        end if
    end for
▷ Exchange current TSRs with new neighbors.
    for *each node* $v \in v_{t_{c,n}}$ **do**
        Send TSR_u *to* v
        Receive TSR_v *from* v
        $t_{s,n} = t_{c,n}$
    end for
▷ Growth of TSRs
    for *each* $SID_i \in SP_u$ **do**
        $\Delta = \sum_{v \in v} \frac{w_u(SID_i, t_{s,n}) \cdot (t_{c,n} - t_{s,n})}{\psi_{i,u,v}}$
        $w_u(SID_i, t_{c,n} + 1) = min\{1, w_u(SID_i, t_{c,n}) + \Delta\}$
    end for
end procedure

Algorithm 3 Gossip Routing Algorithm: Node u decides to which connected neighbors in v to forward the message msg at time t

```
procedure SENDMESSAGE(u, msg, v)
    for v_k ∈ v do
        if msg.destination == v_k then
            Forward msg to v_k
            u.messages = u.messages - msg
        end if

        if msg ∉ v_k.messages then
            S_u = Σ_{SID_i ∈ msg.SID_s} w_u(SID_i, t)
            S_{v_k} = Σ_{SID_i ∈ msg.SID_s} w_{v_k}(SID_i, t)

            if S_{v_k} > S_u then
                Forward msg to v_k
            end if
        end if
    end for
end procedure
```

3.3 Dataset

PSN mainly works by opportunistically transferring messages though human mobility pattern which is formed by human behavior. For our simulation we have used the dataset Sassy [6] which is well related to our PSN criteria. In [6] , 25 participants were equipped with 802.15.4 Tmote Invent sensors and they had been tracked for 79 days. From their obtained data the [6] was created and people or nodes used in the data were in frequent communication. We expanded the tracing detailed in [27] which results in a dense trace of encounters among the participants. By understanding the contact patterns of [6] , we have created Synthetic Sassy dataset of 3 months (17 weeks) for 25 nodes. Because we know larger dataset provides more accurate result. The details of the dataset is shown in the table- 3.2.

3.4 Simulation Setup

We have created a custom simulator for simulating our provided Gossip algorithm and comparing it with other routing algorithms. For the comparison we have chosen both the traditional (First

Table 3.2: Dataset Analysis

Dataset	Days	**Participants**
Sassy	79	25
Synthetic Sassy	119	25

contact, Bubble Rap) and the modern routing algorithms(Chichat). We have maintained same scenario and environment by using the same dataset for simulating all the algorithms. Thus they are same in base and comparable. The detailed description of our custom scenario is given below:

3.4.1 Simulator

Our custom simulator had been written in java using the IDE Netbeans. The Simulator consisted of several classes but among them Node class and Message class were used in all the algorithms. Few special classed had been written for social interest based algorithm Chitchat and our proposed algorithm Gossip among them `social_interest`, socialProfile are well mentionable.

3.4.2 Node Class

The node class is the class to describe participants of the simulation. Each entity has a message buffer equipped with variables bufferMsgID, DesitnationNodeId, bufferMsg and destinationMessages. Other than the buffer we have a TSRlist and a socialProfile. The method hasThisMessage(int m) returns a Boolean value based on the availability of the message with id m in the buffer. The hasInTSR(int sid) finds out if the sid is in the TSR or not. The clearBuffer() method clears the existing message buffer. There is also another variable encounterTime which notes down the last encounter with other nodes on each day.

3.4.3 Message Class

The message class consists of several methods and variables which each message might have in the real world. Every time we need a new message in the simulation environment this message class is initiated. Each message class has several get and set methods and few constructors to set variables for the simulation purpose and for calculating several metrices such as delay rate, drop rate and delivery ratio. The variables used in the message class are messageID, TTL (Time to Live), sourceNodeID, destinationNodeID, destinationGroup, generationTime, deliveryTime, delayMetric, deliveryStatus, dropStatus, currentNodeID. Moreover, the number of messages Message class has the integer dSID to keep the destination social ID of the message. This class has two constructors, one can create object with on SID and other one can create object with multiple SIDs.

3.4.4 Simulator Class

Simulator class is the hub of simulation. It has TTL to note the time to leave of the simulated messages, messageSize to note how many messages will be created per week, nodes as a list of nodes, msgs as a list of messages, siList as a list of social interests, totalDelivered to count how much messages has been delivered, totalDeliveryTime to note the latency, totalWeek is the simulation length. The generateMessages() creates all the messages. The generateNodes() creates all the nodes. PrintAllMessages() prints all the messages, and printAllNodes() prints the details of all the nodes. Simulate() simulates the algo, calculateDelivers() performs the delivery related calculations. ClearAllBuffers() clears out all the message buffers.

3.4.5 Gossip Class

Gossip class holds the main algorithm performed by the simulation. We have variable beta that holds decay factor, currentDay and currentTime. We have to kinds of decay method based on the algorithm decay(..) and decaySP(..) that calculates the decay. We have growthCalc(..) to calculate growth, findDamp(..) to find the dampening factor. Delta(..) to calculate delta. tsrExchange(..) performs exchange of TSR between two nodes. Connect(..) performs the connection between two nodes. sendMessage(..) performs sending of the messages.

3.4.6 Algorithm Classes

we have used Chitchat, First Contact and Bubble Rap for comparing with our algorithm Gossip. Each of the algorithms had been simulated using their own classes but their base, environment, scenarios and dataset have been the same.

3.4.6.1 First Contact

The First Contact routing algorithm is a single message copy routing algorithm and the message is only transferred to the destination node. Thus it has low transmission cost, delivery ratio and high drop rate. This algorithm is not suitable for PSN environment and scenarios and will not solve real life problems related to PSN.

3.4.6.2 Bubble Rap

The Bubble Rap Routing algorithm works by using two metrics which are global ranking and local ranking. Our bubble rap class is created in such a way that each node belongs to a randomly assigned community and has initial localRank and globalRank. When two random local nodes of a community node A and node B meets, both of them check whether they have any buffered messages for each other and transfer if have. Otherwise they check the local ranking to transmit messages to work as relay node. On the contrary, if one of them is a foreign community node and encountered node is not the desired destination node, then they check the globalRank. If the globalRank is higher than its own globalRank then transmit the message.

3.4.6.3 Chitchat Class

Chitchat class is similar to our Gossip class having a variable beta that holds decay factor and time. We have two kinds of decay method based on the algorithm decay(..) and decaySP(..) that calculates the decay. We have growthCalc(..) to calculate growth, findDamp(..) to find the dampening factor. Delta(..) to calculate delta. tsrExchange(..) performs exchange of TSR between two nodes. Connect(..) performs the connection between two nodes. sendMessage(..) performs sending of the messages.

3.4.7 Data

The dataset [6] consists of contact information of 25 nodes for 79 days. Using this dataset 's pattern we have created a Synthetic Sassy dataset of 3 months or 17 weeks and for this reason total 1700 messages have been created; 100 messages per week for each of the algorithms. These messages had been generated in the different times within these 17 weeks. When the source node had been assigned with the message, the source and the destination of each of the messages were set.

CHAPTER 4

SIMULATION RESULTS AND ANALYSIS

In this chapter, we have given a brief description of our custom simulator and compared the simulation results with existing traditional algorithms First Contact [12] and Bubble Rap [17] and modern algorithm Chitchat [22]. For the simulation process, we have used the datasets Sassy [6] and Synthetic Sassy .

4.1 Setting of Default Experimental Parameters

With a view to simulate our proposed algorithm we have created a network consisting of 25 nodes and we have generated 100 messages per week. For getting greater accuracy with our simulation results we have used the dataset Sassy [6] and created a dataset Synthetic Sassy which is for 3 months or 17 weeks. For creating this dataset, we have followed the tracing probability of a random node present in the contact pattern. The default experimental values used in the simulation are mentioned below in table 4.1 and 4.2.

Table 4.1: List of Parameters and their default values for First Contact and Bubble Rap

Parameter	Default Experimental value
Nodes	25
Messages	100 per week
Drop status	False
Delivery status	False
TTL	1 week
Buffer	10 messages per node
Simulated Time	17 weeks

Table 4.2: List of Parameters and their default values for Chitchat and Gossip

Parameter	Default Experimental value
Number of participants or Nodes	25
Pool of Social Interest Keywords	6
TTL	1 week
Buffer	10 messages per node
Simulated Time	17 weeks

We have used two types of algorithms from the field of PSN. So the default values for each of algorithms have minor difference from each other. For the Chitchat [22] and our algorithm Gossip we have used defined number of social interests and list of social interest keywords because these are social interest based algorithms which uses people's behavior to form their social profile and route messages. Otherwise all the parameters used are almost similar in all the algorithms.

4.2 Results

We have illustrated our simulation results by comparing it with other traditional (First Contact, Bubble Rap) and modern algorithms (Chitchat) from the field of PSN in the efficiency measuring parameters of message delivery rate, drop rate, average latency or delay, message replication overhead and transmission cost and their definitions are mentioned below.

- Delivery Ratio: Percentage of successful delivery of messages within TTL.
- Drop rate: Percentage of unsuccessful message delivery because of expiration of TTL.
- Message replication overhead: Percentage of message duplicates in the network in order to increase the delivery ratio.
- Transmission cost: Percentage of message transmission though each node.
- Average delivery latency: Average latency is the time between the message generation and the successful delivery of the message.

4.2.1 Delivery ratio

From the figure- 4.1 we can see that the First Contact provides the lowest delivery ratio because from [12] we know that it is a single copy message forwarding protocol which forwards the message to the first node available in its transmission range and thus often fails to deliver messages successfully. On the other hand, Bubble Rap [17] has successfully surpassed the First Contact in most of the cases because it mainly works by identifying communities and our provided network or dataset is not enormous. Moreover, Chitchat [22] and our algorithm Gossip both are the social interest based algorithm and have exceed the traditional algorithms (First Contact and

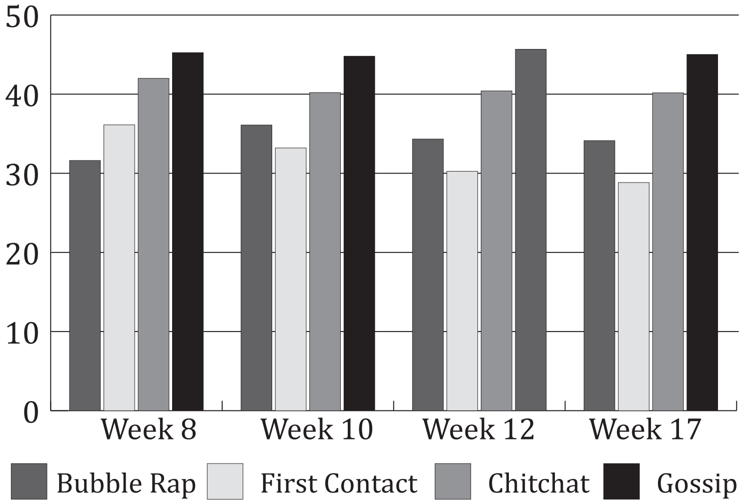

Figure 4.1: Delivery Ratio vs. Simulation Length

Bubble Rap) in the Delivery ratio metric. As we can see our algorithm Gossip has shown best results in delivery ratio which is higher than all other algorithms in our comparison. Throughout the whole process with the increase of dataset our algorithm Gossip has been able to outshine compared to other algorithms.

4.2.2 Drop Rate

From the figure- 4.2 we can see that with the increase of the number of weeks or simulation length, First Contact [12] have the maximum drop rate because only a single copy of message is available in the network which eventually in the most cases do not reach the destination node within the TTL. Both the Bubble Rap [17] and the Chitchat [22] have the drop rates which is greater than our algorithm Gossip. So algorithm Gossip has successfully reduced the message drop rate because our algorithm transmits messages using node's social interests which helps in decreasing unnecessary duplication of messages and thus reduces drop rate.

4.2.3 Message Replication Rate

In the figure-4.3, we can see that First Contact [12] have the lowest message replication rate and it is zero. Because only a single copy of the message is routed throughout the network with a view to successfully deliver the message. On the other hand, Bubble Rap [17] has the highest message replication rate because when the destination node is residing in a different community, the source node forwards the message to the node with the highest global ranking from the destination node's community and that node floods the message among its local community members or nodes. Chitchat [22] shows the second lowest message replication rate. Because it is

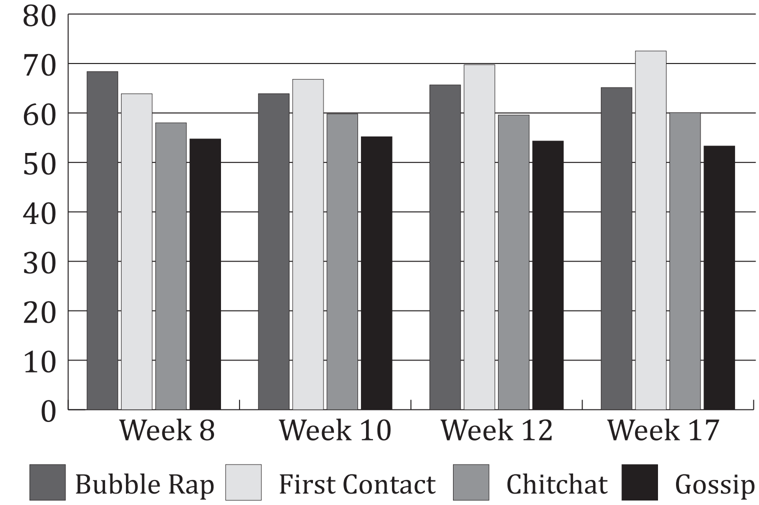

Figure 4.2: Drop Rate vs. Simulation Length

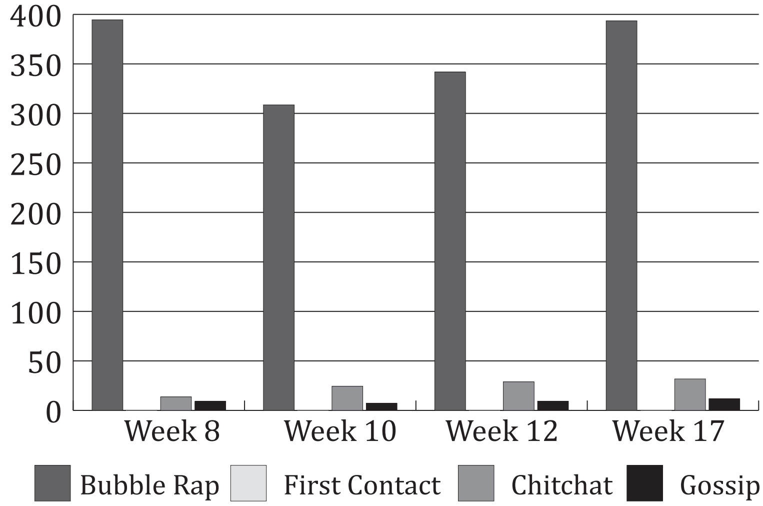

Figure 4.3: Message Replication Rate vs. Simulation Length

a modern social interest based algorithm. However our proposed algorithm Gossip has achieved the lowest message replication rate because it ensures no unnecessary buffer usage and does not create congestion in the network with huge number of replicated message which eventually decreases the drop rate and helps to achieve higher delivery ratio.

4.2.4 Transmission Cost

Here from this figure-4.4, we can see that Bubble Rap [17] has the highest transmission cost because it floods the message in the destination node's local community to deliver the message

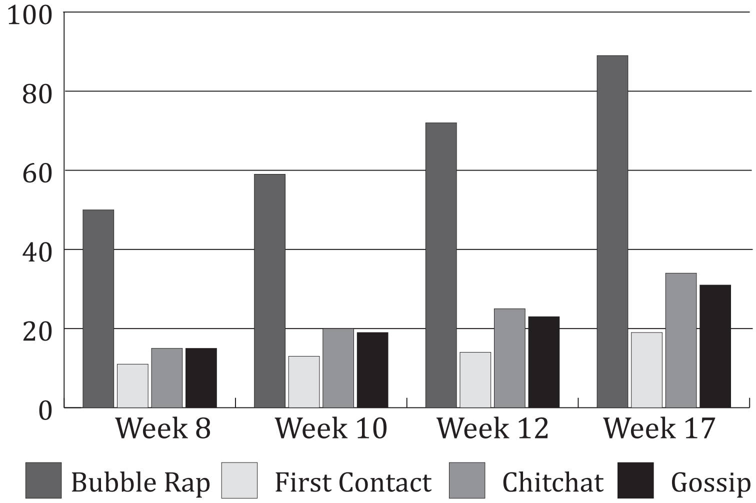

Figure 4.4: Message Transmission Rate vs. Simulation Length

to the destination node. While First Contact [12] has the lowest transmission cost because only a single copy of message is routed. Chitchat [22] have the second highest transmission cost nevertheless our proposed Gossip has showed the lowest transmission than both algorithms-Bubble Rap and Chitchat.

4.2.5 Average Latency or Delay

In the figure-4.5, First Contact [12] have revealed the lowest average latency because it does not spend time to chose efficient relay nodes and forwards the message to the first encountered node which eventually negatively contributes in high drop rate and low delivery rate. While Bubble Rap [17] is the second lowest because it floods the message in destination node's whole local community and often creates congestion in the network while contributing to higher drop rate. However Chitchat [22] and Gossip have comparatively higher average delay than the traditional routing algorithms because the message forwarding decision process of these two is complicated and they select efficient relay nodes which improves the message dissemination process. Besides in proposed algorithm Gossip, we have considered only 3 interests per message and the used dataset is not that large and for this reason Gossip has displayed higher average latency. On the contrary, in Gossip higher the numbers of social interests in a single message, lower the probability of having higher average latency. Hence this two have inversely proportional relation. In the figures- 4.6, 4.7, 4.8, 4.9 and 4.10 we have varied the number of social interests for algorithm Gossip while keeping the simulation length to 12 weeks. The figure- 4.10 shows that with the increase of number of social interests the latency is decreasing. We have come up with a trade-off among the high delivery rate, low drop rate and average latency. Additionally, if achieving the lowest latency was our goal then Epidemic [20] would have been the best routing

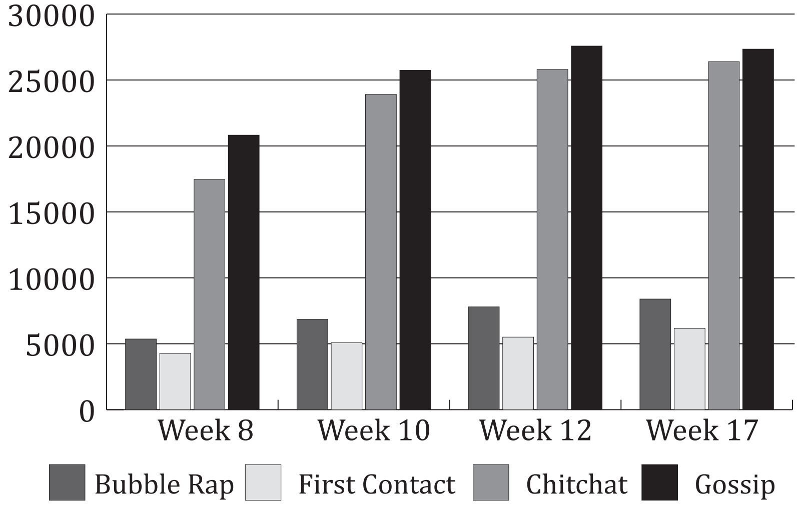

Figure 4.5: Average Latency vs. Simulation Length

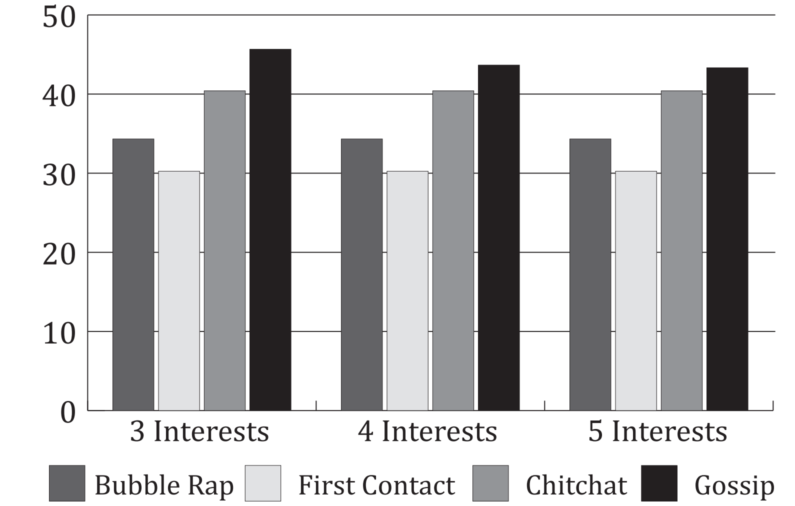

Figure 4.6: Delivery ratio vs. No of social interests used in Gossip

algorithm. But it is not suitable for large and sparse networks like PSN [10]. However, from the Figures - 4.6, 4.7, 4.8 and 4.9 , we can say that even though the number of social interests are varying per messages but our algorithm Gossip has still surpassed the other algorithms in the efficiency measuring parameters of high delivery rate, low drop rate, minimum message replication and transmission cost.

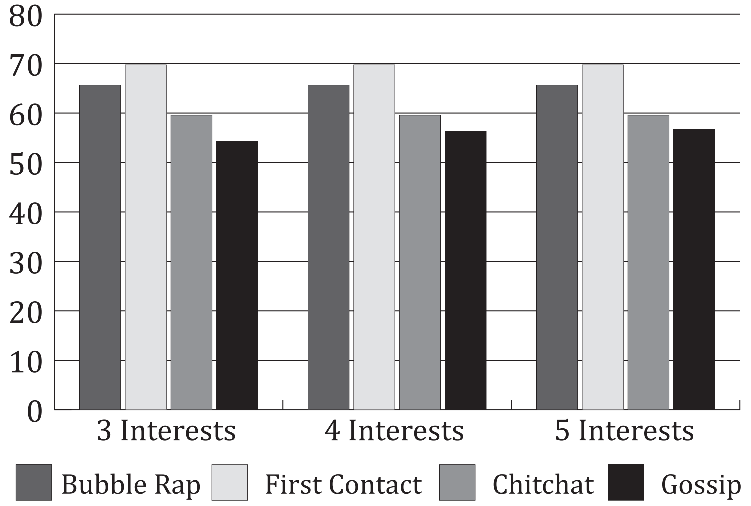

Figure 4.7: Drop rate vs. No of social interests used in Gossip

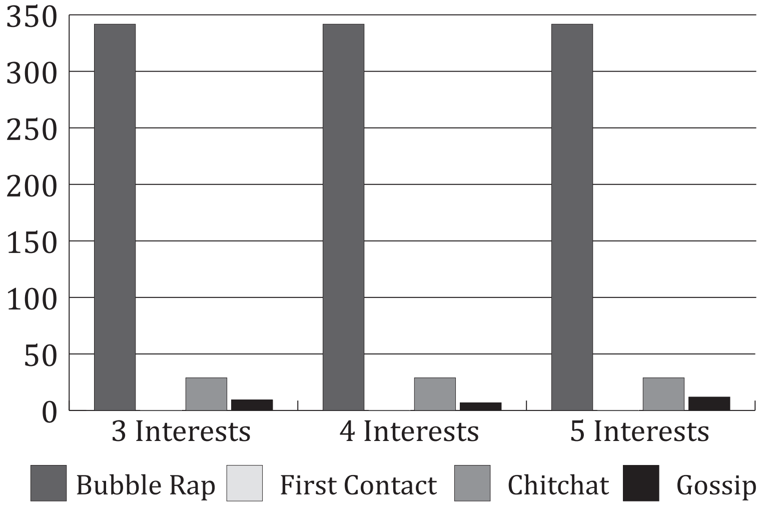

Figure 4.8: Message replication rate vs. No of social interests used in Gossip

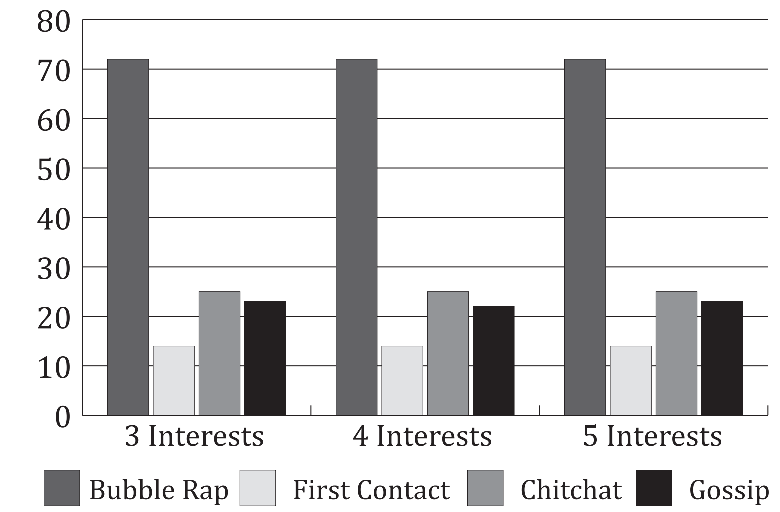

Figure 4.9: Message transmission Rate vs. No of social interests used in Gossip

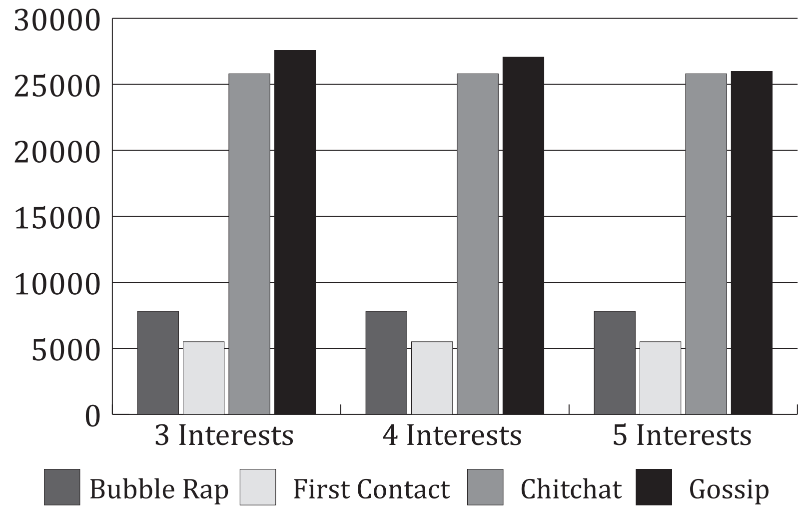

Figure 4.10: Average latency vs. No of social interests used in Gossip

CHAPTER 5

CONCLUSION & FUTURE WORK

In this chapter, we have concluded this thesis report with a discussion of our thesis contribution as well as illustrated about our further work related to thesis.

5.1 Conclusion

Pocket Switched Network (PSN) is a new paradigm in the field of Mobile Ad hoc Network (MANET) and Delay-tolerant Network (DTN). PSN can work without any fixed infrastructure whereas other networks fail to achieve this unique quality. PSN can opportunistically route messages among people or nodes using the human mobility evolving into a routing pattern which is based on various parameters such as history of movements, social interest, geographical locations and many others.

Our provided novel routing algorithm called Gossip has been able to outperform First Contact, Bubble Rap and Chitchat in the performance measuring parameters of delivery ratio and low drop rate maintaining a short buffer size. Considering Gossip algorithm works by using the social interests of people through growth and decay function for successful message routing, larger dataset consisting of more people with various interests will increase the success rates to a great extent.

5.2 Future Work

In this thesis, we have presented a novel social interest based routing algorithm called "Gossip". It has been able to surpass traditional and modern routing protocols of PSN in the metrics of delivery ratio, low drop rate, message replication and transmission cost.

Our future work includes simulating our algorithm using larger datasets and comparing its result with more algorithms from the field of PSN. We have discussed few major problems related with PSN network, we will to provide solutions using our algorithm to solve as many as challenges of PSN network as possible. Finally, as we know the upcoming generation is being more and more depended on hand-held mobile devices, hence we will try to provide a new topology of PSN which will help them to communicate with each other more efficiently, efficiency and conveniently.

Bibliography

[1] M. Ali, M.-R. Fida, A. S. Arsalaan, and A. Adnan, *Implementation and analysis of clustering techniques applied on pocket switched network*, International Journal of Distributed Sensor Networks, 2015 (2015), p. 2.

[2] T. E. Amah, M. Kamat, W. Moreira, K. A. Bakar, S. Mandala, and M. A. Batista, *Towards next-generation routing protocols for pocket switched networks*, Journal of Network and Computer Applications, (2016).

[3] R. Barua and S. Shadman, *Performance analysis of pocket switched network algorithms*, PhD thesis, BRAC University, 2016.

[4] R. Barua, S. Shadman, and A. Chakrabarty, *Pngp : A social relationship based routing algorithm for pocket switched network*, in 2016 19th International Conference on Computer and Information Technology (ICCIT), Dec 2016, pp. 25–30.

[5] M. A. Bayir and M. Demirbas, *On the fly learning of mobility profiles for routing in pocket switched networks*, Ad Hoc Networks, 16 (2014), pp. 13–27.

[6] G. Bigwood, T. Henderson, D. Rehunathan, M. Bateman, and S. Bhatti, *CRAWDAD dataset st_andrews/sassy (v. 2011-06-03)*.
Downloaded from `http://crawdad.org/st_andrews/sassy/20110603`, June 2011.

[7] E. Bulut and B. K. Szymanski, *Friendship based routing in delay tolerant mobile social networks*, in Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, IEEE, 2010, pp. 1–5.

[8] H. Cui, S. Srinivasan, and H. Schulzrinne, *Onechat: Enabling group chat and messaging in opportunistic networks*, in Eleventh Workshop on Mobile Computing Systems and Applications (HotMobile), Citeseer, 2010.

[9] E. M. Daly and M. Haahr, *Social network analysis for routing in disconnected delay-tolerant manets*, in Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing, ACM, 2007, pp. 32–40.

[10] M. FIDA, M. ALI, AND A. ADNAN, *Socialcircle: A message forwarding technique for pocket switched networks*, in 2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2015, pp. 1–7.

[11] A. GALATI, *Delay tolerant network*, (2010).

[12] V. GAMIT AND M. H. PATEL, *Evaluation of dtn routing protocols*.

[13] S. GANGELE AND N. BODNE, *Adaptive multicast for dtn for reducing delay*, Global Journal of Engineering Science and Research Management (April, 2015).

[14] S. HU, H. LIU, L. SU, H. WANG, T. F. ABDELZAHER, P. HUI, W. ZHENG, Z. XIE, AND J. A. STANKOVIC, *Towards automatic phone-to-phone communication for vehicular networking applications*, in INFOCOM, 2014 Proceedings IEEE, IEEE, 2014, pp. 1752–1760.

[15] J. HUANG, X. CHENG, J. BI, AND B. CHEN, *Wireless relay selection in pocket switched networks based on spatial regularity of human mobility*, Sensors, 16 (2016), p. 94.

[16] P. HUI, A. CHAINTREAU, R. GASS, J. SCOTT, J. CROWCROFT, AND C. DIOT, *Pocket switched networking: Challenges, feasibility and implementation issues*, Autonomic Communication, (2006), pp. 1–12.

[17] P. HUI, J. CROWCROFT, AND E. YONEKI, *Bubble rap: Social-based forwarding in delay-tolerant networks*, IEEE Transactions on Mobile Computing, 10 (2011), pp. 1576–1589.

[18] A. KLEMM, C. LINDEMANN, AND O. P. WALDHORST, *A special-purpose peer-to-peer file sharing system for mobile ad hoc networks*, in Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th, vol. 4, IEEE, 2003, pp. 2758–2763.

[19] S. KUMARI, P. YADAV, AND M. YADAV, *Review of efficient routing in delay tolerant network*, International Journal Of Engineering And Computer Science ISSN, pp. 2319–7242.

[20] Y. LIN, B. LI, AND B. LIANG, *Stochastic analysis of network coding in epidemic routing*, IEEE Journal on Selected Areas in Communications, 26 (2008), pp. 794–808.

[21] A. LINDGREN, A. DORIA, AND O. SCHELÉN, *Probabilistic routing in intermittently connected networks*, ACM SIGMOBILE mobile computing and communications review, 7 (2003), pp. 19–20.

[22] D. MCGEEHAN, D. LIN, AND S. MADRIA, *Chitchat: An effective message delivery method in sparse pocket-switched networks*, in Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on, IEEE, 2016, pp. 457–466.

[23] N. MEHTA AND M. SHAH, *Human-mobility-based spray and wait: Efficient routing protocol for pocket switched networks*, International Journal of Future Generation Communication and Networking, 9 (2016), pp. 11–22.

[24] A. MEI, G. MORABITO, P. SANTI, AND J. STEFA, *Show me your friends and i‚Äôll tell you what you like*, Proceedings of Extremecomm, (2010).

[25] A. MEI, G. MORABITO, P. SANTI, AND J. STEFA, *Social-aware stateless forwarding in pocket switched networks*, in 2011 Proceedings IEEE INFOCOM, April 2011, pp. 251–255.

[26] A. D. NGUYEN, P. SENAC, AND M. DIAZ, *How disorder impacts routing in human-centric disruption tolerant networks*, in Proceedings of the 2013 ACM SIGCOMM workshop on Future human-centric multimedia networking, ACM, 2013, pp. 47–52.

[27] I. PARRIS AND T. HENDERSON, *Privacy-enhanced social-network routing*, Computer communications, 35 (2012), pp. 62–74.

[28] C. M. PATEL AND N. GONDALIYA, *Enhancement of social based routing protocol in delay tolerant networks*, International Journal of Computer Applications, 122 (2015).

[29] A.-K. PIETILÄINEN, E. OLIVER, J. LEBRUN, G. VARGHESE, AND C. DIOT, *Mobiclique: middleware for mobile social networking*, in Proceedings of the 2nd ACM workshop on Online social networks, ACM, 2009, pp. 49–54.

[30] G. E. PRESCOTT, S. A. SMITH, AND K. MOE, *Real-time information system technology challenges for nasa‚Äôs earth science enterprise*, in Proceedings of The 20th IEEE Real-Time Systems Symposium, Phoenix, Arizona, 1999.

[31] K. RASUL, D. MAKAROFF, AND K. G. STANLEY, *Hybrid community-based forwarding: A complete energy efficient algorithm for pocket switched networks*, in Local Computer Networks Conference Workshops (LCN Workshops), 2015 IEEE 40th, IEEE, 2015, pp. 760–768.

[32] R. R. SARKAR, K. RASUL, AND A. CHAKRABARTY, *Survey on routing in pocket switched network*, Wireless Sensor Network, 7 (2015), p. 113.

[33] T. SPYROPOULOS, K. PSOUNIS, AND C. S. RAGHAVENDRA, *Spray and wait: an efficient routing scheme for intermittently connected mobile networks*, in Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, ACM, 2005, pp. 252–259.

[34] M. Y. S. UDDIN, H. AHMADI, T. ABDELZAHER, AND R. KRAVETS, *Intercontact routing for energy constrained disaster response networks*, IEEE transactions on mobile computing, 12 (2013), pp. 1986–1998.

[35] E. WANG, Y. YANG, AND J. WU, *Energy efficient phone-to-phone communication based on wifi hotspot in psn*, in 2015 24th International Conference on Computer Communication and Networks (ICCCN), IEEE, 2015, pp. 1–8.

[36] E. WANG, Y. YANG, J. WU, AND W. LIU, *Phone-to-phone communication utilizing wifi hotspot in energy-constrained pocket switched networks*, (2016).

[37] Z. YANG, B. ZHANG, J. DAI, A. C. CHAMPION, D. XUAN, AND D. LI, *E-smalltalker: A distributed mobile system for social networking in physical proximity*, in Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on, IEEE, 2010, pp. 468–477.

[38] F. ZHANG, I. JOE, D. GAO, AND Y. LIU, *An efficient multiple-copy routing in intermittently connected mobile networks*, International Journal of Future Generation Communication and Networking, 9 (2016), pp. 207–218.

[39] X. ZHANG, G. NEGLIA, J. KUROSE, AND D. TOWSLEY, *Performance modeling of epidemic routing*, Computer Networks, 51 (2007), pp. 2867–2891.