Performance Analysis of Pocket Switched Network Algorithms

Rupa Barua **12201056**

Shauvik Shadman **12201093**

Supervisor: **Dr. Amitabha Chakrabarty**

Department of Computer Science and Engineering
BRAC University

DECLARATION

We hereby declare that this thesis is a presentation of our original work. Materials of work found by other researchers are mentioned with due reference to the literature and acknowledgement of collaborative research and discussions.

The work is done under the guidance of Dr. Amitabha Chakrabarty, at the Department of Computer Science and Engineering, BRAC University, Dhaka.

Date of Submission: 17th August, 2016

Signature of the supervisor

____

Dr. Amitabha Chakrabarty

Assistant professor

Department of Computer Science and Engineering

BRAC University

Dhaka, Bangladesh

Signature of Author

____

Rupa Barua

____

Shauvik Shadman

Abstract:

Pocket switch network (PSN) is a type of delay tolerant network which is a suitable process in areas where there is no Internet connection. In a PSN, end-to-end connectivity is not assured. There are several routing protocols available in PSN. This is a new and very attractive research domain. Since its inception PSN has seen various proposals for efficient routing in an infrastructure-less scenarios where human mobility is the only way to transfer information. Our main objective is to develop a new routing protocol which will outperform the existing routing protocols in terms of delivery ratio, latency average, number of forwarded messages, and the number of messages dropped.

Keywords: human mobility, nodes, packet forwarding, delay tolerant network, PSN.

Acknowledgement

We have been supported by numerous people during our research work. A few words' mention here cannot adequately capture all our appreciation.

We are very thankful to our thesis coordinator Dr.Amitabha Chakrabarty, Assistant Professor, Department of Computer Science and Engineering, BRAC University for guiding us throughout our thesis work. The main idea of our algorithm was provided by our supervisor. Due to his endless support and patience, we were able to achieve our goals. We will always be grateful to him, for his kind words, his belief in us, his immense support and guidance.

We are also grateful to Talha Siddique, Sadia Intesar, Shakil Bin Karim, A.K.M Nivrito and Ratna Sarkar, Students of BRAC University for being with us throughout the thesis work.

Last but not least, we would like to thank our parents, our brothers and sisters, for all their enduring support and always believing in us.

CONTENTS

CHAPTER 1

INTRODUCTION

Pocket Switched Networks are a part of Delay Tolerant Networks(DTN). Delay Tolerant Networks are those networks where a packet of data can handle delays. It may take 1 hour to deliver, it may take 24 hours to deliver or it can take an even more time. DTN is a huge topic of study which is designed for intermittently connected networks. Connectivity, like in the usual internet routing, is not assured in DTN. There is no continuous connectivity in DTN, and so the delivery of a packet uses a "store and forward" method. In this method nodes store the packet and holds it for some time dictated by rules and forwards it to the next node. Pocket Switched Networks are a sub division of DTN which specializes on using human mobility as a basis of packet or message transfer. It uses humans and their movement to transfer data. The term "pocket" here is used instead of packet is to specify that the data is being exchanged without notice from one cell phone in the "pocket" to the next. Cell phones are everywhere and since they are mostly kept in pockets hence the term PSN.

We have tried to provide a routing algorithm in this paper after providing the existing literature and have worked to make routing better than before.

1.1 Motivation

Pocket Switched Networks (PSN) is moderately a new field to explore and research, it can work without a specific infrastructure. Recent study shows that human based network connections are less inconsistent and long-termed than connections based solely on node mobility [3]. Therefore, for extreme situations social based DTN was more emphasized than any other form of DTN. For that reason PSN was introduced because it is the only form of DTN which concerns human behavior. Moreover, the current trend to opportunistically route in PSN is to base routing on human behavior and to take up a social approach which are less volatile and may lead to better routing[24]. It is suitable for circumstances including natural disasters, rural areas, deep forests, etc.

In our thesis, we propose a routing algorithm called Popular Node Gateway Protocol for transferring messages between mobile device users which provides better performance in terms of delivery ratio, average delay, transmission cost and the amount of packet dropped. (The following metrics have been elaborated later in this paper).

1.2 Thesis Contribution

This thesis contributes to the society a new routing algorithm for Pocket Switched Networks. Prior to this work, there have been works done on the algorithms for this field but they have failed to achieve results which can be said to be satisfactory and on an acceptable level.

1.2.1 Problem Statement

One of the main impediments of pocket switched networks is the routing of them. Since PSN uses human mobility to forward messages to and from one person to another using everyone as nodes, therefore, the routing is very important. This is because it is impractical to give the message to everyone and use everyone's buffer to forward messages. So, the problem here is to decide a way to forward messages so that a higher delivery is possible and also it gets delivered in a very small amount of time.

1.2.2 Solutions

There were some algorithms proposed prior to our algorithm which have been mentioned in the literature review, which provide a possible solution to the problem of routing. We have just tried to provide an algorithm which is a better routing solution than the rest of the present routing algorithms. We have tried to solve it in a different way than the rest.

1.2.3 Methodology

In the literature, we will see that the methods used by certain algorithms makes the use of the community concept, which we have used as well. We have assigned communities to each experimental nodes and have tried to route them accordingly. Every community is very small and hence if there is flooding in the network with a message then the delivery is very high and also the delay is very less.

1.3 Summary of Contributions

The work presented in this thesis contributes to improving the quality of Pocket Switched Networking algorithms through simulating using our own simulator and running tests on our own algorithm as well as the algorithm already present in the literature.

There are four primary modes of comparing the algorithm such as delivery ratio, average latency, transmission cost, drop rate. We have improved the delivery ratio substantially and also have improved the average latency so that messages can get across faster and gets delivered.

In this thesis, we have shown that it is indeed possible to deliver messages faster without dropping a huge number of messages including the guarantee of delivering every message by a fair percentage.

1.4 Goals

In our thesis, our main objective is to come up with an algorithm that gives better performance than Bubble Rap. We considered the results of Bubble Rap to be the standard while doing comparison with our own algorithm, Popular Node Gateway Protocol (PNGP), because given that its performance in terms of delivery ratio and delay are better than the rest of the existing recent routing algorithms for PSN.

1.5 Thesis Outline

Chapter 1 gives a brief overview of what we plan to do, what we planned to achieve and what we have gained.

Chapter 2 discusses related work done in this field and the background study. We have tried to provide all the algorithms present in the literature and have provided a brief comparison only using the results simulated by others.

Chapter 3 explains our simulation setup and our proposed algorithm.

Chapter 4 discusses results and evaluation.

In chapter 5 concludes this paper with a brief summary and future work.

CHAPTER 2

Literature Review

We have known TCI/IP protocol, for as long as internet existed. Although, it has been with us for quite some time now, it has its limitations. It does not work where there is a lack of end-to end connectivity. If there is a lack of proper infrastructure, the message (or the packet) has a high chance of taking more time to reach the destination. Also this poses a risk for the packet to get lost (it can get dropped before it reaches the destination). This leads us to Delay Tolerant Networking (DTN). It consists of changing the computer network architecture in such a way so that the messages (or packets) have a way to tolerate delay when there is loss of end-to-end connectivity. [20] has categorized DTN as: flooding based, forwarding power based and social based. In flooding according to [20] messages are passed to everyone regardless of probability, whereas in forwarding a lot of metrics are taking into accord before routing.

Similar projects related to this type of networking which have been developed include mobile ad-hoc networks (MANETS) and vehicular ad- hoc networking. MANETS talk about a infrastructure-less network of mobile devices connected without wires.

Opportunistic Networks [31] also known as Pocket switch network (PSN) [37] state that a human carrying a mobile phone can be both an end-user (destination) and a router (a relay node). It is one form of wireless communication network (independent of end to end connectivity between nodes) which is an instance of DTN [2]. PSN is sort of network which cannot use traditional end to end connectivity or TCP/IP protocols, rather PSN uses the opportunistic meetings of human beings specifically for forwarding messages or packets whereas DTN takes into account all sorts of possible carriers including human beings, to forward data[22]. Here are some routing algorithms, that are used or implemented in PSN in order to transfer data.

There are quite a few routing algorithms present in the literature. We have gone through most of the routing algorithms and have provided below a brief on how they work and a brief comparison basing on the results we have noticed from studying the other algorithms.

2.1 Epidemic Routing Protocol

The main idea of **Epidemic routing** [3] is: whenever the packet-carrying (source) node comes across a node that does not have a copy of that packet already, the source node is said to "infect" this new node by passing the copy along and the newly infected node behaves the same when it comes into contact with other susceptible nodes (i.e., nodes without a packet).This routing protocol trades off performance by achieving minimum delivery delay with an increased usage of resources like transmission power, buffer size, bandwidth etc.

There are some recovery schemes associated with this protocol [1]. Firstly, after the packet has been delivered to the destination node, a node can generate an "anti-packet" within itself so that others nodes would not pass along the same packet again. This is known as the "**IMMUNE**" recovery scheme. A more strict approach according to [1] is the forwarding of the "anti-packet" among the infected nodes (which is known as the "**IMMUNE-TX**" scheme) or among both the infected and susceptible nodes (which are known as the "**VACCINE**" scheme) so that the number of copies sent is reduced. Both the "**IMMUNE-TX**" and "**VACCINE**" have similar buffer requirements.

2.2. First-Contact

The concept of first contact routing [4] involves only a single copy of the message available in the whole network. A node forwards the packet only when any single contact is available. If there are none, the message waits till one is available. The source node passes along the copy of the message to the first node it comes in contact with, making that node a relay node if it is not the destination itself. Once the message is passed along, the node deletes that message from its buffer. To make sure two nodes did not exchange the same message back and forth, a node forwards the message only to nodes which did not have the message before. Then after passing along and deleting the copy from its buffer it generates the “anti-packet” so that it does not get re-infected. But since the current node selects the next node randomly, it does not guarantee that the next node has a higher probability of contacting the destination node than the current one, so no high yield (bad-delivery ratio). Moreover, even if the previous node had a greater chance to reach the destination node, it cannot be re-infected. This routing protocol only works if the source and the destination is only one hop away.

2.3 Bubble-Rap

What is a community? It has been a vital concept of sociology and ecology for a long period of time. Community is a term used to assemble people who have common taste or maybe living in the same location [5].

This protocol is based on human behavior. As in, it operates following a trend of popularity (connectivity). All the nodes in the network are grouped into a community and the node passes along a message based on the popularity ‘RANK’, usually if the next node is in a higher rank than the current one[4]. For this protocol to work, every node must belong to a community and have two rankings: local and global.

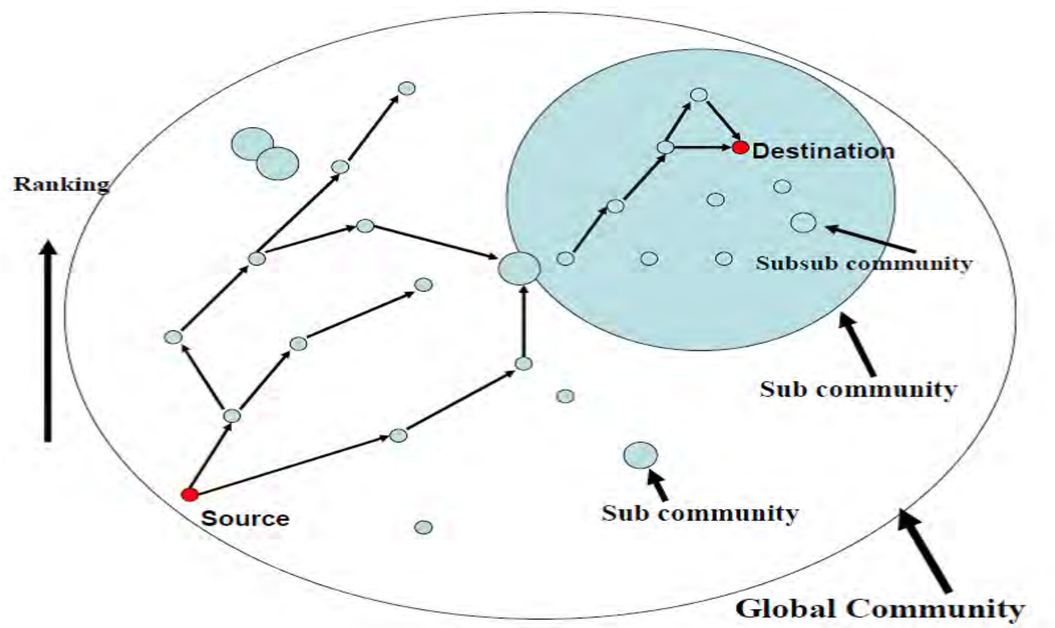

Fig.2.1.1: Working principle of Bubble-Rap [5]

The global ranking labels the node in the entire society whereas local ranking denotes the node’s place in its own community [5]. If the destination node is within the community then the message forwarding will depend on a higher local ranking than the current node. Otherwise, the forwarding of the message will depend on a higher global ranking until it comes in contact with a node in the destination’s community. From then on, higher local ranking is used to forward the message. By doing so, the probability of reaching the destination node will be greater.

But what if one person belongs to multiple communities, that is, what if the communities overlap? It is vital to detect this feature. The K-clique method completes this purpose, and it is designed for binary graphs, so specifying the threshold of the edges is important. WNA (Weighted Network Analysis) can work on weighted graphs directly but cannot detect overlapping communities. We use both as per our needs.

2.4 Spray and Wait

Spray and wait is another routing protocol for Pocket switched networks which was proposed in [11]. This routing protocol is similar to the flooding-based idea. Spray and wait is a modified version of the flooding based protocol. The first definition of Spray and wait is that it has two phases, which are as follows:

1. Spray phase: every message that the source node carries, L number of copies of the message are initially forwarded by the source node and possibly other nodes receiving a copy to L number of different "relays".

2. Wait phase: if the destination is not found in the spraying phase, each of the L nodes carrying a copy of the message forwards it only to its destination (performs direct transmission)

Epidemic routing and flooding has been morphed, resulting to Spray and Wait. Flooding keeps giving out copies of the message to every node it encounters until it reached its destination through nodes who received the copy or through directly passing. But Spray and Wait makes L copies and sends them to L distinct nodes. Those nodes keep the copy until they meet the receiver. Here the number of message copies and how many to share remains open to discussion. In spray and wait, if there are L number of copies, then any node A that has n >1 message copies (source or relay), and encounters another node B (with no copies), it hands over half the amount (n/2) to B and keeps the half for itself; when it is left with only one copy, it switches to direct transmission.

2.5 Lobby Influence

Lobby Influence was proposed in [6]. Lobby Influence works similar to Bubble Rap in the sense that it uses opportunistic ways to pass messages and both are social based forwarding algorithm but it uses a modified version of bubble rap to enhance the delivery ratio and puts less stress on the most popular node.

The basis for Lobby Influence is derived from [7] who presented the metric known as Lobby Influence. They used diplomats dilemma [8] which states that a diplomat has a high influence in the society because he knows a lot of influential people of the society. Therefore a diplomat is as important as the actual influential person and he can reach them with minimum effort and low cost.

The culmination of two criteria gives rise to the Lobby influence routing protocol : Node Popularity(np) and Lobby Index (li).

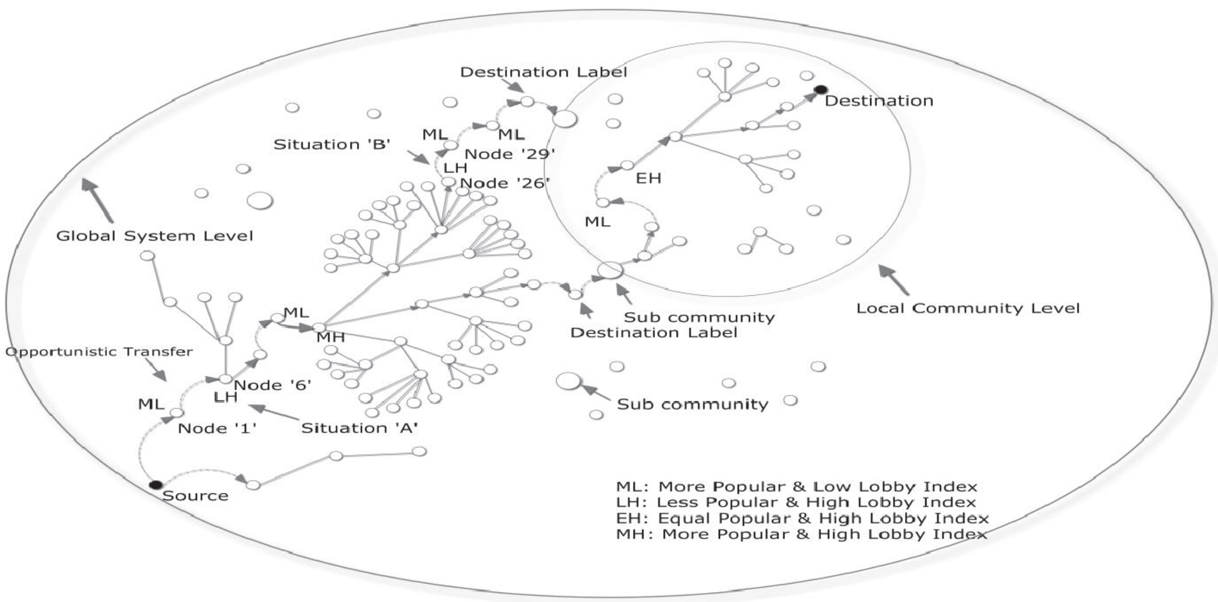

Fig.2.1.2: Working principle of Lobby Influence [6]

In summary Lobby Influence takes up the ideas of Bubble Rap and Lobby Index algorithms and merges them together which results in the overcoming of the shortcomings of both the algorithms. The algorithm proposed by Khan et al may come across two situations on the basis of their algorithm:

1. **Node Within a local community:**

   In this situation if the destination node is within the local community and if the encountered node is part of the local community then local rank and lobby index is used to determine the forwarding decision. If lobby index or rank is higher, it forwards the message.

2. **Node within the global system:**

   In this situation the destination node is part of the global community. If the encountered node is part of the same community as the destination node then the message will be transferred and deleted from the node which it was sent from.

2.6 PROPHET

This algorithm is proposed in [13] which stand for Probabilistic Routing Protocol using History of Encounters and Transitivity. They have hugely relied on the repeated behavioral pattern which means that if a node visits a location several times, it is likely to visit that location again. Delivery Predictability is a probabilistic metric which was used by Lindgren et al which is defined as $P_{(a,\ b)}$at every node a, for each known destination b. This is used to understand the level of probability node a has to deliver messages to node b. Prophet is similar to epidemic routing in the sense that when two nodes meet they exchange messages as well as delivery predictability information stored at the nodes.

They have calculated the delivery predictability of messages which is briefly described below:

Firstly the metric update is taken into consideration. The metric needs to be updated every time a node is encountered so that it is understood which nodes have high delivery predictability. The following calculation has been used:

$$P_{(a,b)} = P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}}) \times P_{init}$$

Secondly, the delivery predictability must age because if one node does not meet another certain node for a while then according to their hypothesis it is less likely that they will meet again. An aging constant, has been used in the following equation:

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

Thirdly, the delivery predictability has a transitive property. This basically means that if node A meets node B frequently and node B meets node C frequently then it is highly probable that a message given to C will be delivered to A. They have used a scaling constant in the following equation:

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

Finally the forwarding strategy is pretty simple and straight forward. When two nodes meet if the delivery predictability of the destination of the message is higher at the encountered node then the message is passed.

2.7 Friendship-based routing algorithm

In PSN, since the people are considered as nodes, this algorithm involves making the decision based on the friendship between two nodes (people). This algorithm introduced a new metric called social pressure metric (SPM), taking into account different sides of social behavior of people. For two people to be considered friends of each other, they have to meet up frequently, make regular and long-lasting contact. In order to ease up the challenges faced during discontinuous end-to-end connectivity, [14] have emphasized on three components of friendship: durability, high frequency and regularity. Friendships can be strong and weak. Two nodes can be good friends directly, other scenarios include two nodes having no direct friendship among them but has a very strong mutual friend. In that case, they can be considered as indirect close friends. To label such indirect connections, [14] suggests to use conditional SPM between. And when it comes to direct friendships, every node can identify them using their own contact history.

This algorithm follows the following forwarding strategy: if node A has to forward a message for node B, and meets node C in the middle, A will forward the packet if and only if C has a stronger friendship with B than A in the current time period. But in [14], it was also mentioned that even if C has a better connection to B than A but does not include B in its current time period (time taken to form a community), A will not forward it to C.

2.8 APPLICATION DOMAIN

The very first application domains that can be noticed are the areas where there are no infrastructures for any end to end connectivity. PSN should be normally applied using cell phones. For example, in a dense forest this can be used since very few people travel through there. If everyone has PSN enabled cell phones then data collection and sharing of data would be very easy (since no connection is required). The people living in very distant rural areas may use PSNs as non-interactive internet [25]. Secondly the work of [15] on the reindeer herd of the Swedish Lapland's is mentionable. They have used DTNs, but they could also use PSN which would perform the same. Thirdly, NASA has a lot of satellites in space and using that we get weather information, short/long range environmental prediction, global air current prediction, and also predicting natural hazards according to [16].A way has been searched such that satellites can share information. In their paper, they have discussed the sharing of such data for earth observing satellites that will support the next generation of space exploration. In [21] they suggest that satellites, which can be subjected to tolerate delays and where long space and satellite communications is needed, we can use DTN. Moreover, they have pointed out that DTN is the way to go to support future space programs and deep space communication which is also supported by [25,26]. Again, since PSN is a part of DTN this can also be applied here. Another application domain is collecting information on animal behavior, ecology, habitat preference of animals, physiology and movement patterns [17]. The use of PSN may greatly improve efficiency. We can use the work done at [18] to know the weather information throughout the park. The military may also use PSN specifically to gather information and to transmit messages about their environment among their own respective sides. Since in a war zone there is absolutely no internet and no cellular networks they may use PSNs[25, 26]. Lastly, in extreme situations like natural disasters, we can use PSN in rescue missions. In a disaster recovery scene we can use PSN to find victims and gather vital information about the rescue workers and also to station them in an efficient manner [25, 21, 26].

2.9 Comparison

Analyzing all the data from the collected research papers, here is a table to show comparison among the some of the existing algorithms over their performance in the following metrics: Delivery Ratio, Drop Rate, Transmission Cost and Average Latency (Average Delay).
(Detailed description of the metrics mentioned here are given later in the paper while comparing our own algorithm with the existing ones).

	Epidemic	**First-Contact**	**Spray and wait**	**Prophet**	**Bubble Rap**
Number of messages delivered (Delivery Ratio)	Excellent	Poor (Very low)	Adequate	Good	Very Good
Number of messages dropped (Drop Rate)	Poor (very high)	Excellent (very low)	Adequate	Good	Very Good
Number of messages forwarded (Transmission Cost)	Poor (very high)	Excellent (very low)	Good	Adequate	Very Good
Average Latency (Average Delay)	Poor (very high)	Excellent (very low)	Good	Adequate	Very Good

Table 2.9.1: Comparison of different routing algorithms

So, from the above table we can clearly see that, Bubble Rap stands out of having a better performance on average than the rest of the algorithms (except Epidemic). The reason Epidemic routing protocol is not considered as a standard to compare other algorithms to, is because of it's extremely high transmission cost. Since it spreads out copies of the message to every node in the network, this can lead to a network congestion when implemented in a large heterogeneous network. So, our objective was to develop an algorithm that will outperform Bubble Rap and produce an increased Delivery ratio, delay, reduced the transmission and the number of packets dropped.

2.10 Open issues in pocket switched networks:

Reports in [27] point to an estimate of 982 million cell phone users in the world and that PSN will be very useful and popular in the near future. So if we can implement PSN using all the 982 million cell phones we can have almost an infallible network. There is still a lot of work that needs to be done to implement pocket switched networks. Work on reliability of the messages and to control the huge amount of data(which can lead to congestion in the network) is done in [19] but they are based on MANETS, which can easily be implemented on PSNs. In the near future, space programs may need to implement PSNs which is still open for research as mentioned in [21]. Work is done in [27, 28, 29] to show that Wi-Fi connections of the cell phones may be used to create a pocket switched network to send and receive messages from a distant location. There is still room for improvement in this and the routing protocols for better delivery ratio and lower drop rate. A very big issue in PSN is: there is very little or almost no work done in the security measures. One of the attacks is known as flood attacks which is an issue for DTNs mostly, but may also be an issue in PSNs. In [23] they address this issue by implementing rate limiting [30] but, then again, this is for DTNs. The basic idea is very simple, which is to limit the source node to transmitting a ceiling value of messages. Therefore in the future, the security of PSN may still be researched upon to create a full-proof system of passing messages.

CHAPTER 3

Popular Node Gateway Protocol

In this chapter we have explained our proposed algorithms in great details and have shown the simulation results later on which were done using our own simulator.

3.1 Design of our algorithm:

We have named our algorithm Popular Node Gateway Protocol. The main idea of our algorithm is derived from the idea of the most popular node in a community. We thought about the person who we meet the most people in our own community, for example, if a village is considered a community then the chairman of that village may be the most popular person in that community and he can convey this message to everyone in his community. This algorithm takes into account the destination community of a message. If the node met is the destination community then a check is done to see if that node is the most popular node at his community, if it is so, then transmission occurs. If that node is the destination then, it is delivered. But if it is not is the destination then that node floods its own community with that message. This is why the delivery is very high here. Moreover, due to the fact that a community is formed considering the nodes who meet often, the delivery is much more.

The most popular node is a gateway to that community. Hence, we have used the name Popular Node Gateway because it provides a gateway and then it floods its own community.

3.2. Pseudo code:

The pseudocode of our algorithm is given below.

Algorithm 1: Popular Node Gateway Protocol

// Directly Delivering

ForEach encountered $node_i$ **do**

ForEach $message_k$ **do**

If $node_i = message_k.destination$ **and** $message_i.getDeliveryStatus = false$ **then**

Deliver $message_k$ to $node_i$

transmission++

delivery++

Delete($sourceNode_i$, $message_k$)

EndIf

EndForMessage

EndForEncounter

//Possible transfer check

ForEach encountered $node_i$ **do**

ForEach $message_k$ **do**

If $node_i.group = message_k.destinationGroup$ **and** $message_k.getDeliveryStatus = false$ **then**

If $node_i.mostPopular = true$ **and** $sourceNode_i.group != node_i.group$ **then**

transmit($node_i$,$message_k$)

transmission++

Delete($sourceNode_i$, $message_k$)

else

transmit($node_i$,$message_k$)

transmission++

EndIf

EndForMessage

EndFor

3.3 Dataset

We used the SASSYdataset [35] for the simulation of our algorithm. In this dataset twenty-five participants were equipped with 802.15.4 Tmote Invent sensors, and they are tracked for 79 days. We augmented the trace as detailed in [36], resulting in a dense trace of encounters between participants.

We also used a new dataset named as Synthetic SASSY by following SASSY dataset [2]. Synthetic SASSY is made by tracing the probability of a node present in the contact pattern. We used this dataset in order to evaluate the performance of our algorithm for a larger system. Synthetic SASSY dataset is 210 days long with 25 participants.

Dataset Details			Input of Experiments	
SASSY	Days	participants	Metrics	Ranges
	79	25	TTL (s)	86400-518400
Synthetic SASSY	210	25	Messages	1200-3000
			Buffer Size	10

Table 3.3.1: Dataset Analysis

3.4 Simulation Setup:

To simulate the different algorithms mentioned above we have created our own simulator. We have simulated Flooding, First contact, Bubble Rap and our own algorithm through similar scenarios and came up with results. Since we have simulated them with the same scenario that means the base is the same with different algorithms which makes the results comparable. The detailed description of our simulator is given below.

3.4.1. Simulator:

The simulator was written in Java and using the IDE Netbeans. The simulator consisted of a Node class and a Message class. These classes have been used to simulate and are similar in all the algorithms. The Node class emulates the different nodes and the Message class emulates the messages in the whole network.

3.4.2. The Message class:

The message class contains methods and variables, which each message might have in the real world. The number of times this message class is initiated is the same number of messages we needed in the simulation environment. Each message class objects have several get and set methods along with a set of variables. It has a constuctor which sets some of the variables in this class. Among the variables we have included a messageID, TTL(Time to Live), sourceNodeID, destinationNodeID, destinationGroup, generationTime, deliveryTime, delayMetric, deliveryStatus, dropStatus, currentNodeID. The methods that are used are basic set and get methods which are used to manipulate the mentioned variables and calculate several metrics such as Delay, Drop and Delivery.

3.4.3. The Node class:

The Node class contains variables and methods which easily emulates the real life Nodes which in case of PSN are cell phones. We have included the variables nodeID, bufferMessageID, destinationNodeID, bufferMessages, localRank, globalRank, groupID. The Node class has a constructor which initiates and sets some of the values to the variables. The bufferMessageID is an ArrayList which contains the messageIDs of all the messages contained by the Node, the destinationNodeID is also an ArrayList which has all the nodeID of the destination of the messages in integer value. The bufferMessage is also an ArrayList which has the message class objects of the messages it contains.

3.4.4. The Algorithm Classes:

The algorithms that we simulated are Flooding, First Contact, Bubble Rap and our own algorithm. Each algorithm has been simulated using their own classes. The in-depth description of the inner workings of each algorithm will be described in the next section.

3.4.5. The Data:

The data consists the information of the nodes who met and at which time. A total of 25 Nodes and 1200 messages were created beforehand for each algorithm. The source of a message and the destination of each message were set in each message and the source node was issued the message. The source node was also fed the destinationNodeID of each of the messages in an array. The messages were created in different times and its generation time was also given to each message object created.

3.5. Settings of the Different Algorithms:

The settings of the different has been explained in brief in the following sections.

3.5.1. Flooding:

In flooding based routing algorithm, each node transfers a copy of the messages it contains to everyone that node meets irrespective of any metric. If the node it transfers to, has its buffer full it drops the messages. As a result drop increases. We have iterated the sassy dataset and the synthetic sassy dataset, and in each pass we have transferred all the messages one node has to the other nodes it met. Flooding has a very large amount of transfers and so is its drop rate. Due to its transfer to everyone, it is natural that its delivery rate is more than algorithms.

3.5.2. First Contact:

In first contact routing algorithm, the node only transfers the messages to the destination node. Each node checks the destination node list to check for the node it met. If the node found, is a match it transfers that message and it gets delivered. But this algorithm is very inefficient because it only transfers messages to destination node, so there is no real routing present. As a result the delivery ratio is very small and so is the transmission. The delivery ratio is so minimal that this is not a real life solution to the problem of routing in the Pocket Switched Network Routing.

3.5.3. Bubble Rap:

In this routing algorithm there are some metrics that need to be understood. The first one is Local Ranking and the other is Global Ranking. We have set up the simulation in such a way that each node belongs to a random group and they have an initial localRank and a globalRank. As node A meets node B, node A checks its buffer for any possible messages destined for node B, if found, delivery occurs, else node A checks the group of node B, if it is the same group then it checks the localRank. If the localRank is greater than node A then it transfers the messages whose in that group. On the hand, if node B is foreign then node A checks its destination group list. If it has any messages destined for that group then transmission occurs. Lastly if there are no messages destined for that group then it checks the globalRank of the node B. If it is higher than node A then messages transfer. This is how we designed Bubble Rap, and iterated over the datasets.

CHAPTER 4

SIMULATION RESULTS AND EVALUATION

In this section, we simulated our algorithm using the datasets SASSY and SYNTHETIC SASSY to generate our results. We compared our simulation results of the existing algorithms with the results gathered from the research papers. Then we simulated our own algorithm PNGP, and compared it with the data obtained from our simulator of the existing ones.

4.1 SETTING OF PARAMETERS

For the purpose of our simulation we have created a network consisting of 26 nodes and a 100 messages per week. That means a total number of 1200 messages are generated in the network when we are using the SASSY dataset and a total number of 3000 messages are generated in the network when we are using the SYNTHETIC SASSY dataset. The default values of parameters used in all the simulation are given below:

Parameter	Default Value
Nodes	26
Messages	100 per week
TTL	84600 s - 518400 s
Delivery Status	False
Drop Status	False
Buffer	10 messages per node

Table 2: List of parameters and their default values

4.2 Results

4.2.1 Delivery Ratio:

4.2.1(a) Delivery ratio when simulated with Sassy Dataset

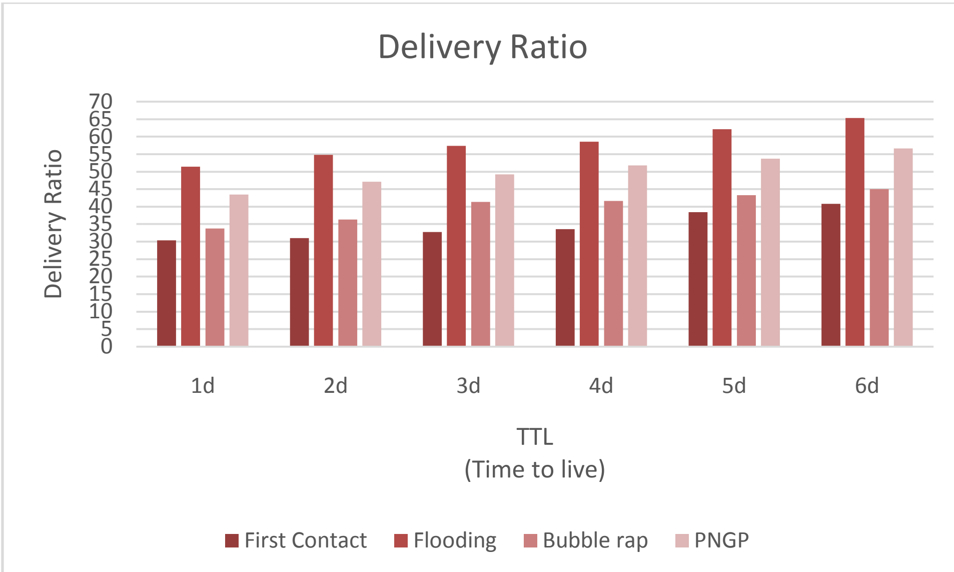

Fig.4.1: Delivery Ratio vs. TTL (SASSY)

In Figure 4.1, we used the Sassy dataset, which sets the time for simulation for 12 weeks . Here, Epidemic gives the highest Delivery Ratio since, it transmits the messages to the whole network . So, the number of messages is very high in the network, hence, it has a higher chance of delivery. First Contact performs the worst in this field, because first contact essentially works on the concept of giving the message directly to another node. If the node carrying the message does not meet the destination then the message is not delivered. From our observation before simulating PNGP we noticed that Bubble Rap perform the best among all the other routing protocols. But still Bubble Rap had some limitations because it cannot deliver more than 45-50 percent on average. Our algorithm PNGP has a 10% higher Delivery Ratio than Bubble Rap.

4.2.1(b) Delivery ratio when simulated with Synthetic Sassy Dataset

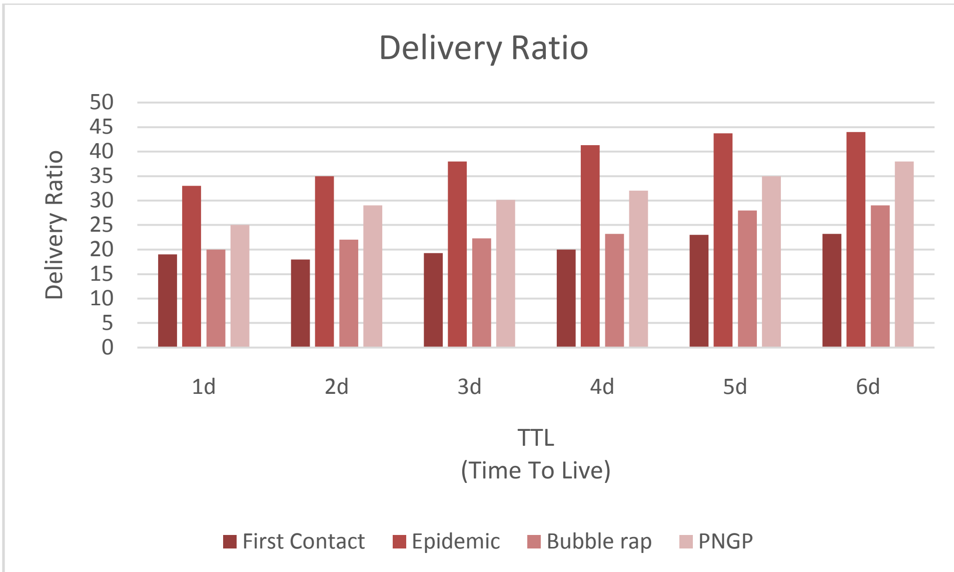

Fig.4.2: Delivery Ratio vs. TTL (SYNTHETIC SASSY)

In Figure 4.2, we used the Synthetic Sassy dataset, which sets the time for simulation for 30 weeks . Here, Epidemic also gives the highest Delivery Ratio since, it transmits the messages to the whole network. First Contact performs the worst in this field, because first contact essentially works on the concept of giving the message directly to another node. If the node carrying the message does not meet the destination then the message is not delivered. From our observation before simulating PNGP we noticed that Bubble Rap perform the best among all the other routing protocols. Our algorithm PNGP has a 30-35% higher Delivery Ratio than Bubble Rap. Since the dataset is larger, the simulations produce higher yield.

4.2.2 Transmission Cost:

4.2.2 (a) Transmission Cost when simulated with Sassy Dataset

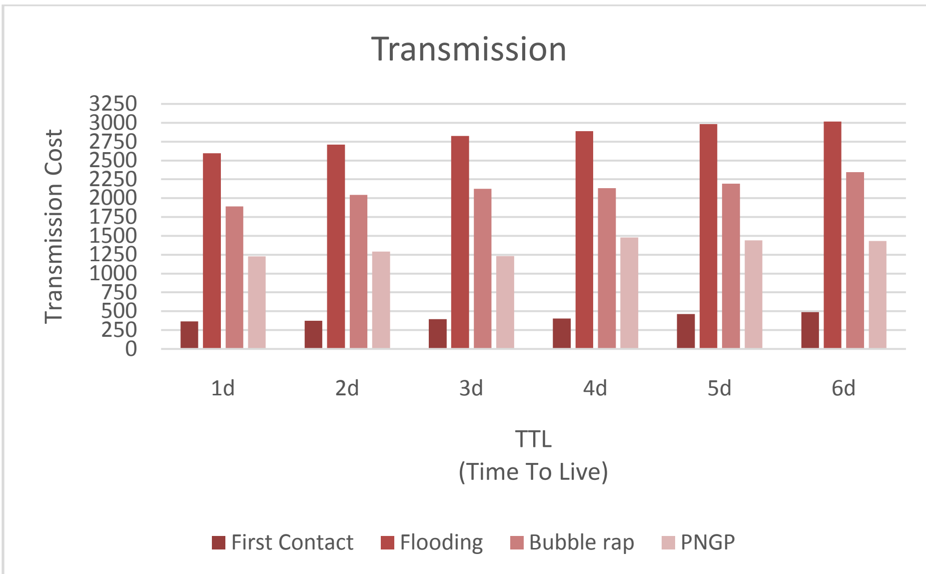

Fig.4.3: Transmission Cost vs. TTL (SASSY)

In Figure 4.3, Epidemic has the highest Transmission Cost as the node sends the message to every other node present in the network. This causes Epidemic to always have the highest amount of transmission. First Contact always has the lowest transmission cost as it passes the message only to one node at a time, giving the best performance in this scenario. Before our simulation Bubble Rap gave next to best performance, as it also passes the message along only to one node at a time according to the community the destination node belongs to. Our algorithm PNGP has a 30% lower transmission cost than Bubble Rap, placing it to be the second best in this scenario as the most popular node in the destination's group is one hop count away.

4.2.2 (b) Transmission Cost when simulated with Synthetic Sassy Dataset

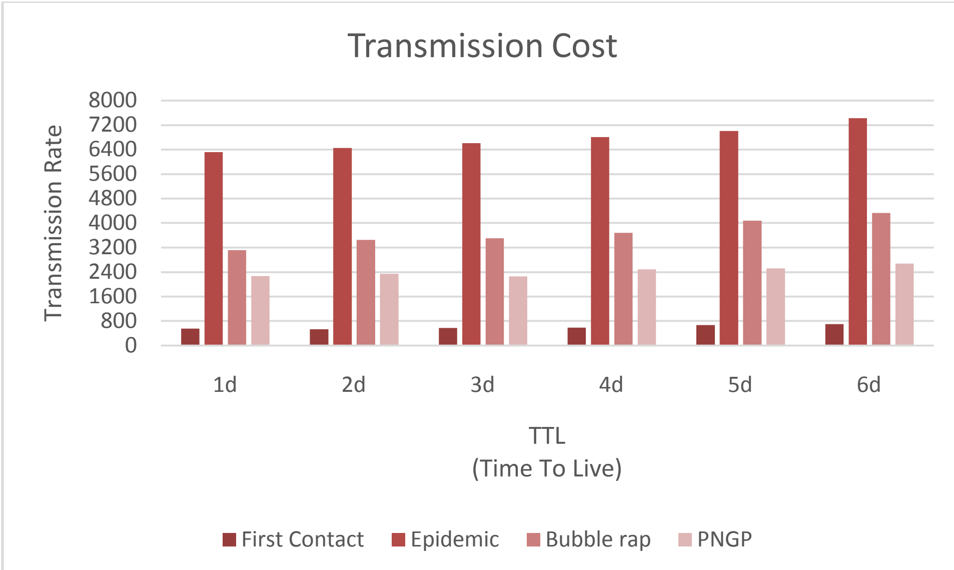

Fig.4.4: Transmission Cost vs. TTL (SYNTHETIC SASSY)

In Figure4.4, we used Synthetic Sassy dataset for the simulation. Here, Epidemic also has the highest Transmission Cost as the node sends the message to every other node present in the network. This causes Epidemic to always have the highest amount of transmission. First Contact always has the lowest transmission cost as it passes the message only to one node at a time, giving the best performance in this scenario. Before our simulation Bubble Rap gave next to best performance,as it also passes the message along only to one node at a time according to the community the destination node belongs to. Our algorithm PNGP has a 25-30% lower transmission cost than Bubble Rap, placing it to be the second best in this scenario as the most popular node in the destination's group is one hop count away. Since the dataset is larger, the simulations produce higher yield.

4.2.3 Drop Rate:

4.2.3 (a) Drop Rate when simulated with Sassy Dataset

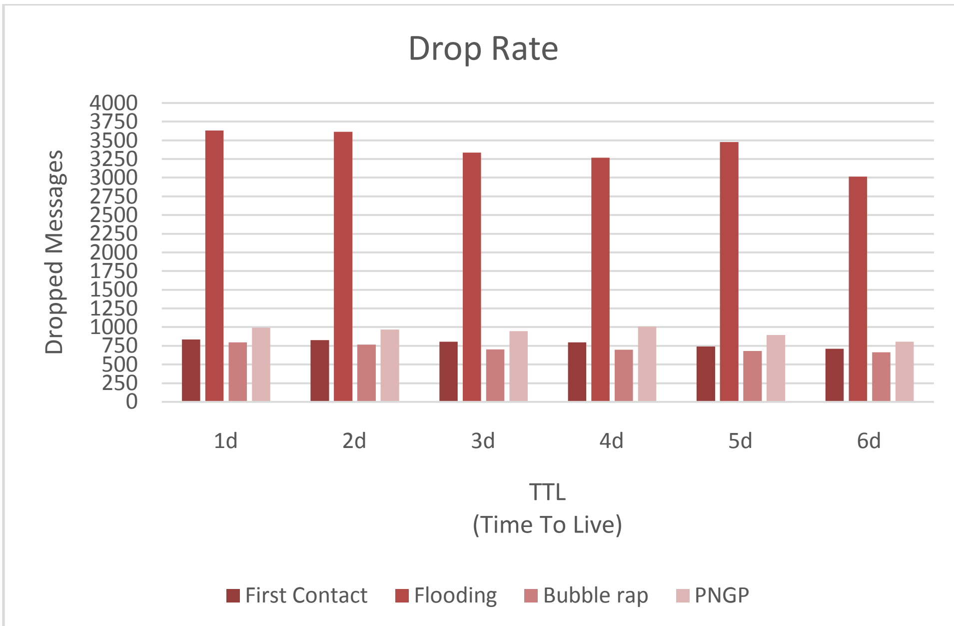

Fig.4.5: Drop Rate vs. TTL (SASSY)

In Figure 4.5, Epidemic has the highest amount of packets dropped as it has multiple number of copies of the message present in the whole network. First contact and Bubble Rap both have single copies of the message present in the network, so their drop rates are much lower than both Epidemic and PNGP. PNGP has a drop rate 25% higher than Bubble Rap, as in PNGP, when the popular node of the destination group receives the message, it transmits the message to everybody in the group (including the destination).

4.2.2 (b) Transmission Cost when simulated with Synthetic Sassy Dataset

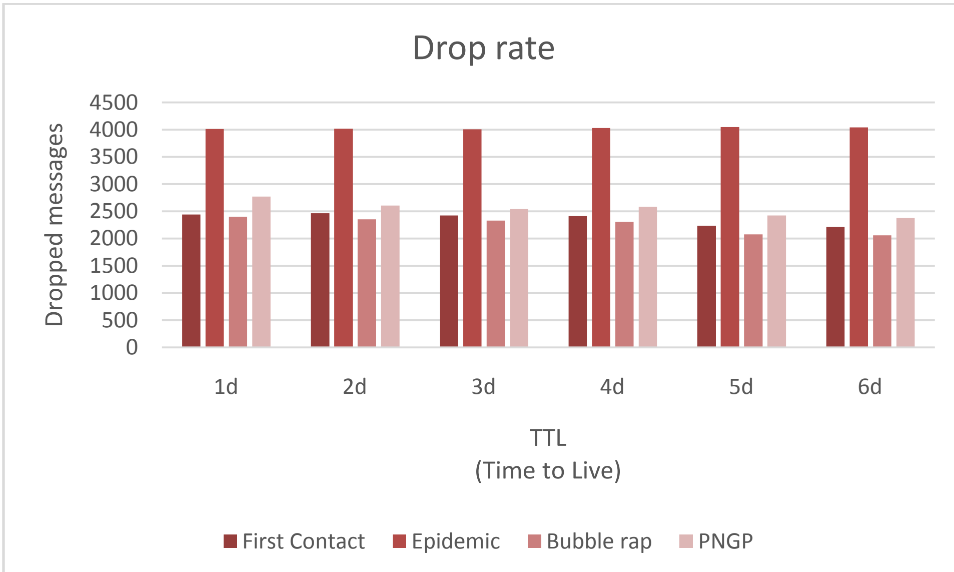

Fig.4.6: Drop Rate vs. TTL (SYNTHETIC SASSY)

In Figure 4.6, we used Synthetic Sassy to simulate the results. Epidemic has the highest amount of packets dropped as it has multiple number of copies of the message present in the whole network. First contact and Bubble Rap both have single copies of the message present in the network, so their drop rates are much lower than both Epidemic and PNGP. PNGP has a drop rate 10-15% higher than Bubble Rap, as in PNGP, when the popular node of the destination group receives the message, it transmits the message to everybody in the group (including the destination). This is a larger dataset than Sassy, so the difference or improvement over Bubble Rap is shown by a larger margin.

4.2.4 Average Latency (Delay):

4.2.4 (a) Delay when simulated with Sassy Dataset

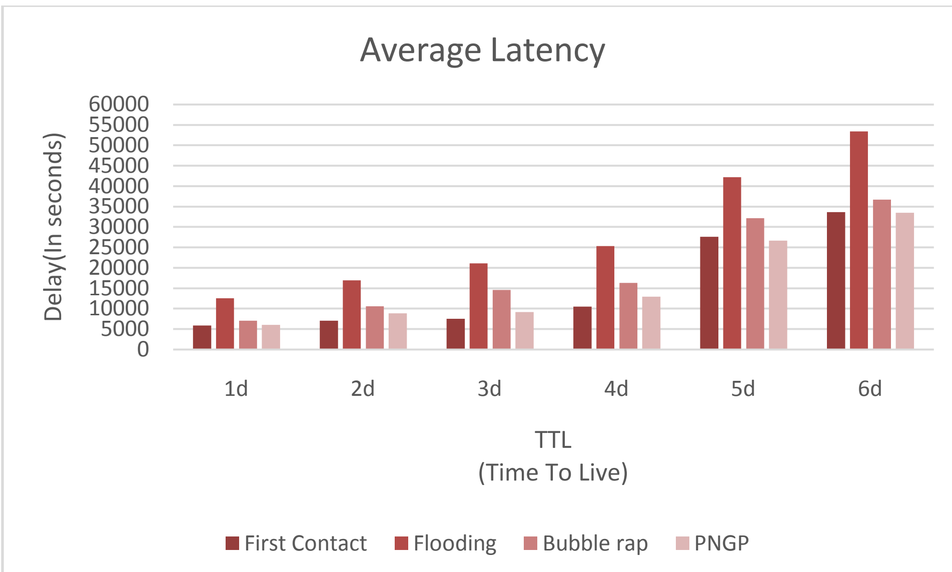

Fig.4.7: Average Latency vs. TTL (SASSY)

In Figure 4.7, we found the value average latency by calculating the time between the message created/generated and delivered. Epidemic has the highest average latency, as the message carrying node transmits the message to every node until it finds the destination. Bubble Rap has the second highest average latency in this simulation. The rest of the two algorithms come very close to having the lowest delay. In two of the cases, First Contact performs better than PNGP. However, even though in some cases, it has a lower average latency, its delivery ratio is still the lowest. PNGP has a delay rate 15% lower than Bubble Rap, as the most popular node of the destination node is just one hop away.

4.2.2 (b) Transmission Cost when simulated with Synthetic Sassy Dataset

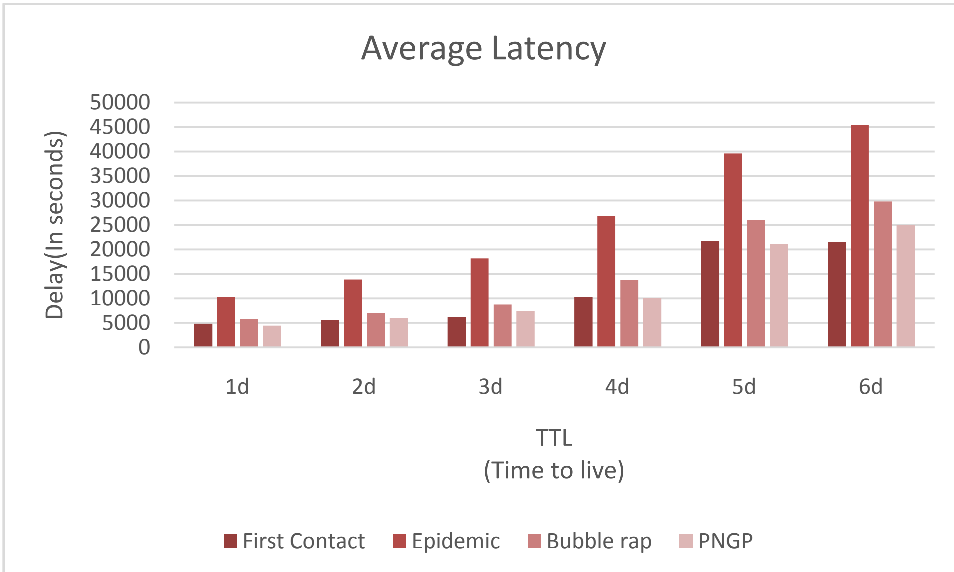

Fig.4.8: Average Latency vs. TTL (SYNTHETIC SASSY)

In Figure 4.8, we used Synthetic Sassy dataset for our simulation. Epidemic has the highest average latency, as the message carrying node transmits the message to every node until it finds the destination. Bubble Rap has the second highest average latency in this simulation. We found the value average latency by calculating the time between the message created/generated and delivered. The rest of the two algorithms come very close to having the lowest delay. In two of the cases, First Contact performs better than PNGP. However, even though in some cases, it has a lower average latency, its delivery ratio is still the lowest. PNGP has a delay rate 22%-25% lower than Bubble Rap, as the most popular node of the destination node is just one hop count away. This is a larger dataset than Sassy, so the difference or improvement over Bubble Rap is shown by a larger margin.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

To conclude we are providing a brief conclusion citing the important aspects of this paper and also have provided the future works that can be done here.

5.1 CONCLUSION

PSN is important for a lot of aspects in real life. As this works without any proper infrastructure, when countries (especially the one which are still developing) gets hit by a tsunami or an earthquake or any other major natural disaster, PSN can help the people who are trapped inside houses or buildings be found by rescuers. Our motive was to come up with an algorithm that outperforms Bubble RAP, which shows good performance on average in all other metrics used in these experiments. Although Epidemic provides a better delivery ratio, due to its extremely high transmission cost, it cannot be implemented in a large heterogeneous network. PNGP showed better performance than Bubble Rap in terms of transmission cost, average latency and delivery ratio.

5.2 FUTURE WORK

In this thesis, we have studied that PNGP gives a better performance than Bubble Rap in almost all the scenarios. We ran our simulation on IDE Netbeans using the datasets: SASSY and SYTHETIC SASSY. Our future work includes, running our algorithm using other datasets. Also forming communities based on social behavior of human beings on social media (Facebook, Twitter, Instagram, etc) or based on how many times a person (node) talks (makes contact) to another node during a specified period of time (a day or a week).

References:

[1] Hossmann, T., Spyropoulos, T., & Legendre, F. (2011, May). Putting contacts into context: Mobility modeling beyond inter-contact times. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing* (p. 18). ACM.

[2] Sarkar, R. R., Rasul, K., &Chakrabarty, A. (2015). Survey on Routing in Pocket Switched Network. *Wireless Sensor Network*, *7*(9), 113.

[3] Zhang, X., Neglia, G., Kurose, J., &Towsley, D. (2007). Performance modeling of epidemic routing. *Computer Networks*, *51*(10), 2867-2891.

[4] Gamit, V., & Patel, M. H. Evaluation of DTN Routing Protocols.

[5] Hui, P., Crowcroft, J., &Yoneki, E. (2011). Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, *10*(11), 1576-1589.

[6] Khan, S. K. A., Mondragon, R. J., &Tokarchuk, L. N. (2012, March). Lobby Influence: Opportunistic forwarding algorithm based on human social relationship patterns. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2012 IEEE International Conference on*(pp. 211-216). IEEE.

[7] Korn, A., Schubert, A., &Telcs, A. (2009). Lobby index in networks. *Physica A: Statistical Mechanics and its Applications*, *388*(11), 2221-2226.

[8] Holme, P., &Ghoshal, G. (2009). The diplomat's dilemma: Maximal power for minimal effort in social networks. In *Adaptive Networks* (pp. 269-288). Springer Berlin Heidelberg.

[9] Dillon, P. J. (2013). Software contribution to the ONE simulator.

[10] Mangrulkar, R. S., &Atique, M. (2011, February). Performance evaluation of flooding based delay tolerant routing protocols. In *Proceedings of the National Conference on Emerging Trends in Computer Science and Information Technology (ETCSIT), Punjab* (pp. 25-26).

[11] Spyropoulos, T., Psounis, K., &Raghavendra, C. S. (2005, August). Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* (pp. 252-259). ACM.

[12] Keränen, A., Ott, J., &Kärkkäinen, T. (2009, March). The ONE simulator for DTN protocol evaluation. In *Proceedings of the 2nd international conference on simulation tools and techniques* (p. 55). ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[13] Lindgren, A., Doria, A., &Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *ACM SIGMOBILE mobile computing and communications review*, *7*(3), 19-20.

[14] Bulut, E., & Szymanski, B. K. (2010, December). Friendship based routing in delay tolerant mobile social networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE* (pp. 1-5). IEEE.

[15] Doria, A., Uden, M., & Pandey, D. (2009). Providing connectivity to the saami nomadic community. *generations*, *1*(2), 3.

[16] Prescott, G. E., Smith, S. A., & Moe, K. (1999, December). Real-time information system technology challenges for NASA’s earth science enterprise. In *Proceedings of The 20th IEEE Real-Time Systems Symposium, Phoenix, Arizona*.

[17] Boehlert, G. W., Costa, D. P., Crocker, D. E., Green, P., O'Brien, T., Levitus, S., & Le Boeuf, B. J. (2001). Autonomous pinniped environmental samplers: using instrumented animals

as oceanographic data collectors. *Journal of atmospheric and oceanic technology*, *18*(11), 1882-1893.

[18] Beaufour, A., Leopold, M., & Bonnet, P. (2002, September). Smart-tag based data dissemination. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (pp. 68-77). ACM.

[19] Nagmote, P. S., &Hande, K. Data Replication To Increase The Reliability And To Control Congestion In Opportunistic Network.

[20] Patel, C. M., &Gondaliya, N. (2015). Enhancement of Social based Routing Protocol in Delay Tolerant Networks. *International Journal of Computer Applications*, *122*(4).

[21] Kumari, S., Yadav, P., &Yadav, M. Review of Efficient Routing In Delay Tolerant Network.

[22] Ali, M., Fida, M. R., Arsalaan, A. S., & Adnan, A. (2015). Implementation and analysis of clustering techniques applied on Pocket Switched Network.*International Journal of Distributed Sensor Networks*, *2015*, 2.

[23] NIHARIKA, M., & SIVAJI, U. (2015). Defending Against Flooding Attacks in Mobile Networks.

[24] Basith, A. (2015*)*. A Framework for Routing Performance Analysis in Packet Delivery Delay Tolerant Networks with Application to Non Cooperative Networks *International Journal of Computer Science information and Engg., Technologies.*

[25] Gangele, S., &Bodne, N. ADAPTIVE MULTICAST FOR DTN FOR REDUCING DELAY.

[26] Wang, E., Yang, Y., & Wu, J. (2015, June). Dynamic beaconing control in energy-constrained Delay Tolerant Networks. In *2015 IEEE International Conference on Communications (ICC)* (pp. 3813-3818). IEEE.

[27] Hu, S., Liu, H., Su, L., Wang, H., Abdelzaher, T. F., Hui, P., ...&Stankovic, J. A. (2014, April). Towards automatic phone-to-phone communication for vehicular networking applications. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications* (pp. 1752-1760). IEEE.

[28] Sharma, A., Navda, V., Ramjee, R., Padmanabhan, V. N., & Belding, E. M. (2009, December). Cool-tether: energy efficient on-the-fly wifi hot-spots using mobile phones. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies* (pp. 109-120). ACM.

[29] Wang, E., Yang, Y., & Wu, J. (2015, August). Energy Efficient Phone-to-Phone Communication Based on WiFi Hotspot in PSN. In *2015 24th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1-8). IEEE.

[30] Li, Q., Gao, W., Zhu, S., & Cao, G. (2013). To lie or to comply: defending against flood attacks in disruption tolerant networks. *IEEE Transactions on Dependable and Secure Computing*, *10*(3), 168-182.

[31] Boldrini, C., Lee, K., Ott, J., Pagani, E. (2014,July).Opportunistic Networks. *Computer Communications* .

[32] You, L., Li, J., Wei, C., Dai, C., Xu, J., & Hu, L. (2014). A hop count based heuristic routing protocol for mobile delay tolerant networks. *The Scientific World Journal*, *2014*.

[33] Jeon, M., Kim, S. K., Yoon, J. H., Lee, J., & Yang, S. B. (2014). A Direction Entropy-Based Forwarding Scheme in an Opportunistic Network. *JCSE*, *8*(3), 173-179.

[34] Hyytiä, E., Bayhan, S., Ott, J., &Kangasharju, J. (2014, September). Searching a needle in (linear) opportunistic networks. In *Proceedings of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems* (pp. 187-196). ACM.

[35] G. Bigwood, D. Rehunathan, M.Bateman, T. Henderson, and S. Bhatti. CRAWDAD data set standrews/sassy (v. 2011-06-03). Downloaded from http://crawdad.org/st andrews/sassy, June 2011.

[36] I. Parris and T. Henderson. Privacy-enhanced socialnetwork routing. *Comput. Commun.*, 35(1):62–74, Jan. 2012. doi:10.10.

[37] Hui, P., Chaintreau, A., Scott, J., Gass, R., Crowcroft, J., &Diot, C. (2005, August). Pocket switched networks and human mobility in conference environments. In *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* (pp. 244-251). ACM.