Performance Analysis of DTN Routing Protocols

A Thesis Book
Submitted in partial fulfillment of the requirements for the Degree of
Bachelor of Science in Computer Science and Engineering

Submitted by

Zakia Sultana Tanzil	**160204004**
Anika Rahman	**160204012**
Arunima Ayshee	**160204066**
Tahira Salwa Rabbi Nishat	**160204070**

Supervised by

Md. Tanvir Ahmed
Assistant Professor

Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Dhaka, Bangladesh

July 2021

CANDIDATES' DECLARATION

We, hereby, declare that the Thesis Book presented in this report is the outcome of the investigation performed by us under the supervision of Md.Tanvir Ahmed, Assistant Professor, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this Thesis Book nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Zakia Sultana Tanzil
160204004

Anika Rahman
160204012

Arunima Ayshee
160204066

Tahira Salwa Rabbi Nishat
160204070

CERTIFICATION

This Thesis Book titled, **“Performance Analysis of DTN Routing Protocols”**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in July 2021.

Group Members:

Zakia Sultana Tanzil	**160204004**
Anika Rahman	**160204012**
Arunima Ayshee	**160204066**
Tahira Salwa Rabbi Nishat	**160204070**

Md. Tanvir Ahmed
Assistant Professor & Supervisor
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

Prof. Dr. Mohammad Shafiul Alam
Professor & Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

First of all, we would like to thank Almighty Allah for successfully completing our thesis work. We take this opportunity to thank Ahsanullah University of Science and Technology for giving us such an elegant opportunity to become part of this educational institution . It has been a privilege for us to receive the Degree of Bachelor of Science in Computer Science and Engineering here.

We would like to thank our research supervisors, Mr. Tanvir Ahmed, Assistant Professor, Department of Computer Science Engineering, AUST. Without his help and your dedicated involvement in every step of the process, this document would never have been finalized. We would like to thank him very much for your support and understanding over the past years.

We would also like to thank all the faculty members of the department who have helped us in various ways during the period. Finally we would like to thank everyone who is directly or indirectly involved with the thesis.

Dhaka
July 2021

Zakia Sultana Tanzil

Anika Rahman

Arunima Ayshee

Tahira Salwa Rabbi Nishat

ABSTRACT

Delay-Tolerant Networks are used to enpower communication in difficult environments where nodes are intermittently connected and there is no end-to-end path between the source and destination, for example ICMN.DTN uses store-carry-forward approach where message is moved from source to destination node through intermediate nodes by opportunistic contacts because of node movement. Therefore, network environments in which nodes are designated by opportunistic connectivity are modeled accordingly as DelayTolerant Networks (DTN) . A key application of DTN is being explored to deliver emergency messages to key personnel in the event of a disaster. The main challenge facing DTN is to adopt an appropriate routing technique to deliver messages in the most efficient way . In this paper, we have explored the performance of DTN routing protocols, namely Epidemic, PRoPHET, and Spray-and-Wait (Binary version) in a given scenario.Our aim is to evaluate the performance of DTN routing techniques for message circulation in the event of disaster when structured communications networks are not available. Their performances are evaluated in terms of delivery probability, average latency, and overhead ratio of varying simulation time, buffer size and transmit range, respectively. Opportunistic Network Environment (ONE) simulator is used as the simulation tool for analyzing these performance metrics. The result of this work shows that for the given simulation configuration, the best DTN routing algorithm is MaxProp and Binary Spray-and-Wait, while epidemic routing has the worst performance in terms of all of the metrics considered here.

Keywords — Delay-tolerant network (DTN), intermittently connected mobile network (ICMN), opportunistic network environment (ONE) simulator, Epidemic, PRoPHET, Binary spray-and-wait, routing, simulation, Delivery Probability, Average Latency, Overhead Ratio.

Contents

CANDIDATES' DECLARATION i

CERTIFICATION ii

ACKNOWLEDGEMENT iii

ABSTRACT iv

List of Figures viii

List of Tables x

1 Introduction 1
- 1.1 Overview 1
- 1.2 Motivation 2
- 1.3 Thesis Contribution 3
- 1.4 Thesis Structure 3
- 1.5 Summary 4

2 Literature Review 5
- 2.1 Overview 5
- 2.2 Related Work 6
- 2.3 Summary of Literature Review 7

3 Background Study 8
- 3.1 Overview 8
- 3.2 Background Related Information 9
  - 3.2.1 Epidemic 9
  - 3.2.2 Prophet 11
  - 3.2.3 Spray and Wait 13
  - 3.2.4 MaxProp 15

4 Proposal 17
- 4.1 Overview 17

4.2 Proposed Method 19
4.3 Summary of Proposed Method 19

5 Implementation 21
5.1 Simulation Environment Setup 21
5.1.1 Simulation Setup for varying Transmit Range : 21
5.1.2 Simulation Setup for varying Simulation Time: 22
5.1.3 Simulation Setup for varying Buffer Size: 23
5.1.4 Simulation Setup for varying Message Generation Rate (message/min): 23
5.1.5 Simulation Setup for varying TTL (In minutes): 24
5.1.6 Simulation Setup for varying Number of Nodes: 24
5.2 Parameters of Routing Algorithms 25

6 Experiment Result 26
6.1 Impact of Varying Transmit Range 26
6.1.1 Impact on Average Latency 26
6.1.2 Impact on Delivery probability 28
6.1.3 Impact on Overhead Ratio 29
6.2 Impact of Varying Simulation Time 30
6.2.1 Impact on Average Latency 30
6.2.2 Impact on Delivery Probability 31
6.2.3 Impact on Overhead Ratio 33
6.3 Impact of Varying Buffer Size 34
6.3.1 Impact on Average Latency 34
6.3.2 Impact on Delivery Probability 36
6.3.3 Impact on Overhead Ratio 37
6.4 Impact of Varying Message Generation Rate 39
6.4.1 Impact on Average Latency 39
6.4.2 Impact on Delivery Probability 40
6.4.3 Impact on Overhead Ratio 41
6.5 Impact of Varying TTL 43
6.5.1 Impact on Average Latency 43
6.5.2 Impact on Delivery Probability 44
6.5.3 Impact on Overhead Ratio 45
6.6 Impact of Varying Number of Nodes 46
6.6.1 Impact on Average Latency 47
6.6.2 Impact on Delivery Probability 48
6.6.3 Impact on Overhead Ratio 49

7 Conclusion and Future Work 51

7.1 Limitation 51
7.2 Future Work 51
7.3 Conclusion 51

References **53**

List of Figures

1.1 DTN Architecture . . . 1
1.2 Store and Carry Forward Approach . . . 2
1.3 Terrorist Attack . . . 3
1.4 Tsunami . . . 3
1.5 Normal and During Disaster Communication System . . . 3

3.1 Single Copy Case . . . 9
3.2 Epidemic . . . 9
3.3 Flowchart of Epidemic [1] . . . 10
3.4 PRoPHET . . . 11
3.5 Flowchart of PRoPHET [1] . . . 13
3.6 Flowchart of Spray and Wait Protocol [1] . . . 14
3.7 The MaxProp Routing Strategy . . . 15
3.8 Flowchart of MaxProp [1] . . . 16

4.1 Overview of One Simulator . . . 18
4.2 Screenshot of One Simulator . . . 18
4.3 Methodology . . . 20
4.4 Specification Settings . . . 20
4.5 Java Class . . . 20

6.1 Average Latency vs Transmit Range . . . 27
6.2 Avg Latency vs Transmit Range Graph Value . . . 27
6.3 Delivery Probability vs Transmit Range . . . 28
6.4 Delivery Probability vs Transmit Range Graph Value . . . 28
6.5 Overhead Ratio vs Transmit Range . . . 29
6.6 Overhead Ratio vs Transmit Range Graph Value . . . 30
6.7 Average Latency vs Simulation Time . . . 31
6.8 Average Latency vs Simulation Time Graph Value . . . 31
6.9 Delivery Probability vs Simulation Time . . . 32
6.10 Delivery Probability vs Simulation Time Graph Value . . . 32
6.11 Overhead Ratio vs Simulation Time . . . 34
6.12 Overhead Ratio vs Simulation Time Graph Value . . . 34

6.13 Average Latency vs Buffer Size 35
6.14 Average Latency vs Buffer Size Graph Value 35
6.15 Delivery Probability vs Buffer Size 36
6.16 Delivery Probability vs Buffer Size Graph Value 37
6.17 Overhead Ratio vs Buffer Size 38
6.18 Overhead Ratio vs Buffer Size Graph Value 38
6.19 Average Latency vs Message Generation Rate 39
6.20 Average Latency vs Message Generation Rate Value 40
6.21 Delivery Probability vs Message Generation Rate 41
6.22 Delivery Probability vs Message Generation Rate Value 41
6.23 Overhead Ratio vs Message Generation Rate 42
6.24 Overhead Ratio vs Message Generation Rate Value 42
6.25 Average Latency vs TTL 43
6.26 Average Latency vs TTL Value 44
6.27 Delivery Probability vs TTL 45
6.28 Delivery Probability vs TTL Value 45
6.29 Overhead Ratio vs TTL 46
6.30 Overhead Ratio vs TTL Value 46
6.31 Average Latency vs Number of Nodes 47
6.32 Average Latency vs Number of Nodes Value 48
6.33 Delivery Probability vs Number of Nodes 49
6.34 Delivery Probability vs Number of Nodes Value 49
6.35 Overhead Ratio vs Number of Nodes 50
6.36 Overhead Ratio vs Number of Nodes Value 50

List of Tables

5.1 Simulation Parameters and Values for Varying Transmit Range 22
5.2 Simulation Parameters and Values for Varying Simulation Time 22
5.3 Simulation Parameters and Values for varying Buffer Size 23
5.4 Simulation Parameters and Values for varying Message Generation Rate . . . 23
5.5 Simulation Parameters and Values for varying TTL 24
5.6 Simulation Parameters and Values for varying Number of Nodes 24
5.7 Parameters of Routing Algorithms 25

Chapter 1

Introduction

1.1 Overview

Delay tolerant Network is a set of protocols which act together executing store carry and forward approach. It provides a network architecture independent messaging service. Figure 1.1 shows DTN Architecture. The application areas of DTN are challenging circumstances like military battlefields, deep underwater communication, natural disasters affected areas or remote area social communication etc. In DTN, similar and different protocols are used to send and receive messages between nodes. That is why, it is called a heterogeneous network. DTN introduces a new layer called bundle layer on the top of TCP/IP layer. This layer is liable for node to node authenticity. Most of the DTN routing protocols share a similar pattern, that is, “store and forward” fashion.

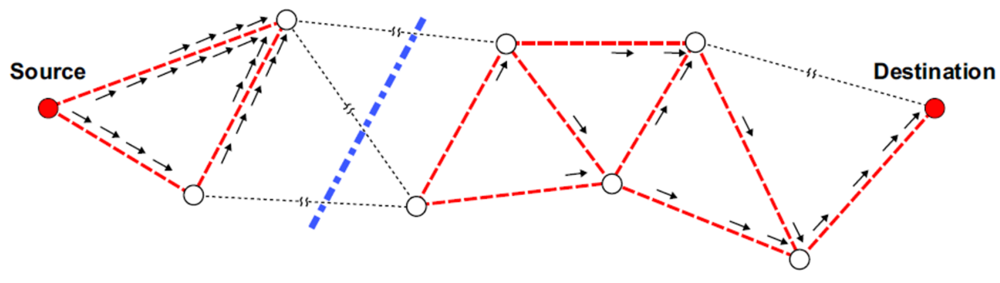


Figure 1.1: DTN Architecture

Figure 1.2 explains that DTN uses the store-and-carry forwarding approach. Messages may be sent to unavailable end hosts. Hop-to-Hop re-transmission happens in case of errors. Store and Forward Approach, complete messages or a chunk of it is transferred and stored in nodes till it reaches the destination. Every node is associated with a persistent storage device (like harddisk) wherever it can store the messages. Here, a communication link may

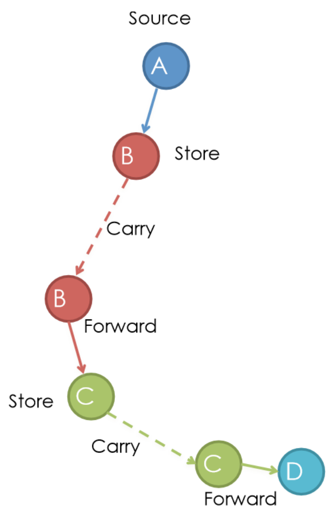


Figure 1.2: Store and Carry Forward Approach

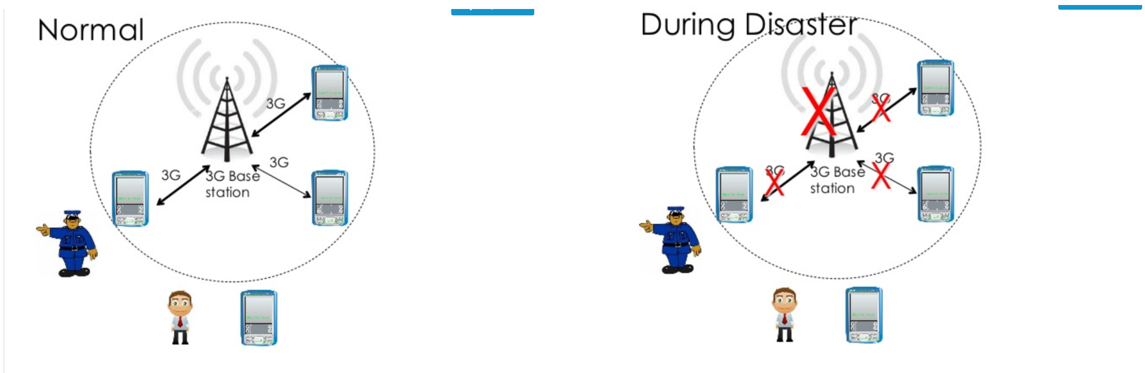
not be available for a long time. One node could send or receive information a lot of quicker or more definitely than the other nodes. A message, once transmitted, may need to be re-transmitted.

1.2 Motivation

Many kinds of disasters like tsunami, hurricane, terrorist attack etc. occurs. Then our as usual communication system has broken. People will not be able to communicate at that time. Packets are simply discarded. Figure 1.3 and 1.4 show different disasters where communication system break down. DTN applies routing nodes with storage to preserve data when links are broken. Data bundles are stored till the next hop is re-established, then they are forwarded on. This indicates, end destinations need not be constantly connected. DTN enables automatic data communication management in short range mobile applications. DTN has been used in many environments where networks have long delays or disruption. The main motivation of DTN is to enable communication between source and destination without the support of a fixed network infrastructure. Figure 1.5 shows the difference between Normal Communication System and During Disaster Communication System.

Figure 1.3: Terrorist Attack

Figure 1.4: Tsunami



(a) Normal Communication System

(b) During Disaster Communication System

Figure 1.5: Normal and During Disaster Communication System

1.3 Thesis Contribution

We've studied few papers related to analyse the performance of DTN Routing Protocols where some of them focused only Message Generation Rate,TTL;some of them discussed about only Buffer Size,Number of Nodes etc.But we didn't find any work where six different simulation parameters are focused.

In this paper,we've focused six different simulation parameters such as- Transmit Range, Simulation Time, Buffer Size, Message Generation Rate(message/min), Time to Live(TTL), Number of Nodes and we successfully analyzed the performance of four different routing protocols- Epidemic,PRoPHET,Spray-and-Wait and MaxProp based on three performance metrics which will be described in later chapter.That is the uniqueness of our work.

1.4 Thesis Structure

In chapter 1, we've discussed about Delay Tolerant Network (DTN),why it is important;motivation of choosing this work.We've also talked about thesis contribution that means the uniqueness of our work and finally we summarized it.

In chapter 2, we've talked over related work and summarized the literature review by which we can mark what kind of work is already done.

In chapter 3,we've explored background study.We focused four different routing protocols-Epidemic, PRoPHET, Spray-and-Wait and MaxProp.In this chapter, we've discussed in detail about these routing protocols,how they work,their flowchart etc.

In chapter 4, we've explained about our proposal.Here, we've talked about our simulator by which we can analyze performance of DTN Routing Protocols.We also discussed about performance metrics and methodology.

In chapter 5, we've described about our implementation setup.We've considered on varying six different simulation parameters which are described in detail here.Mainly simulation environment setup and parameters of routing algorithm are explained in this chapter.

In chapter 6, we've showed our experiment result on varying simulation parameters among four different routing protocols based on three performance metrics.We've summarized experiment result after every experiment and took a decision which routing protocol is the best.

In chapter 7, we've talked about our limitation and future plan to fulfil our work.

1.5 Summary

Delay Tolerant networking (DTN) is a viewpoint to computer network architecture that seeks to deal with technical problems in heterogeneous networks that may lack continuous network connectivity.

- To observe DTN routing protocol
- To review ONE Simulator
- To inspect Delivery Probability, Average Latency and Overhead Ratio
- To evaluate performance of DTN routing protocols using the ONE Simulator

Chapter 2

Literature Review

2.1 Overview

In this chapter, we present an evaluation of DTN routing protocols as related to intermittently connected mobile networks, focused on Epidemic [2], PROPHET [3], and Spray-and-Wait [4]. Epidemic routing is a clear example of the "store-carry-and-forward" approach, which forwards each copy of the messages to each node. This method is flooding-based in nature, and is exhaustive of network resources. The Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) tries to exploit the likelihood of real-world encounters by managing a set of probabilities for successful delivery to acquainted destinations in the DTN. Then it replicates messages during opportunistic encounters only if the mobile node in contact that does not retain the message seems to have a better chance of delivering it to the destination. Spray-and-Wait approach is proposed by Spryropoulos et al to limit the utilization of network resources, in which only a limited number of message copies are replicated among nodes. It parts routing into two phases: a spray phase, where message replicas are spread, and a wait phase, where nodes with single-copy messages wait till a direct encounter with the respective destinations. The rest of this paper is arranged as follows-

Section II explains the DTN routing protocols under investigation in this paper. Section III describes the simulation setup and the analysis of obtained results. Section IV presents the concluding remarks about this research endeavor.

2.2 Related Work

In [5] in this paper, they have dealt with the problem of single copy routing in intermittently connected mobile networks. They have showed a number of increasingly sophisticated single-copy strategies, and used theory and simulations to extensively evaluate their performance.

In [6] they recommended two efficient multi-copy schemes, called Spray and Wait and Spray and Focus that manage to beat the shortcomings of flooding-based and other existing schemes.

In [7] Delay and Disruption-Tolerant Networking (DTN) is also a replacement communication model which will span across multiple networks and affect harsh conditions not visualized in the web model. Disregard the wide variation of operative conditions. The DTN implementations gain in providing enough performance for several situation.

In [8],the document analyzed the Delay Tolerant Network (DTN) service and protocol stack and enabled an Android implementation called Bytewalla. It permits the employment of phones for the physical transportation of information between network nodes in areas where there are not any other links is offered for sending the info.

In [9] of this article they argue that it is not so much the selection or complication of social metrics and algorithms that carries the greatest weight in performance, but rather the mapping of the mobility process that generates contacts to the aggregated social graph. Proposed algorithm that uses ideas of unsupervised learning and graph theory to educe this “correct” graph structure. This algorithm allows every node to locally identify and balance to the optimal operating point, and achieves good performance altogether situation considered.

In [10], opportunistic routing techniques were proposed whereby the node can store and transport a message for some time until a new forwarding possibility arises. These algorithms mainly focus on relatively homogeneous node configurations. However, in many future applications, the participating nodes may use portable devices, vehicles, sensors, etc.

In [11] of this paper, they proposed a practical and efficient common scheduling and cancellation policy which will optimize various performance metrics like average delay and delivery probability. At first, they use the idea of combat-based message broadcast to derive the optimal policy supported global knowledge about the network. Then, disclose a technique that estimates all necessary parameters which using locally collected statistics information.

2.3 Summary of Literature Review

From the previous section, we've found that many work on single copy schemes, multi copy schemes(Spray and Wait and Spray and Focus).Some proposed to optimize different performance metrics for-example, average delay and delivery probability.Some discussed about Delay Tolerant Network on Android Platform.

Later,we'll focus on four routing protocols such as Epidemic,PRoPHET,Spray and Wait and MaxProp.We'll try to analysis three performance metrics such as Delivery Probability,Average Latency and Overhead Ratio on different parametric settings.

Chapter 3

Background Study

After analyzing the related works on performance analysis of DTN Routing protocols, it has been found that most of the works are on the same types of protocols and these works need much time for the simulation process. While exploring previous works, some terms have also been found which are explained in the following sections.

3.1 Overview

In this chapter, we give a compact overview on the classification of DTN routing protocols, and summarize the design of the Epidemic, PROPHET, Spray-and-Wait and MaxProp routing protocols. DTN routing protocols are classified into two fundamental schemes: single-copy and multi-copy.

Single-Copy Schemas

In single-copy schemes, there is only one node in the network that conveys a copy of the message at any given time phase. When the intermediate node forwards the copy to the specific next hop, this becomes the message's new tenure-holder. Figure 3.1 shows Single Copy Schemas.

Multi-Copy Schemas

Multi-copy scheme onwards several copies of the same message to the network, i.e., replicates messages which are called replication-based.

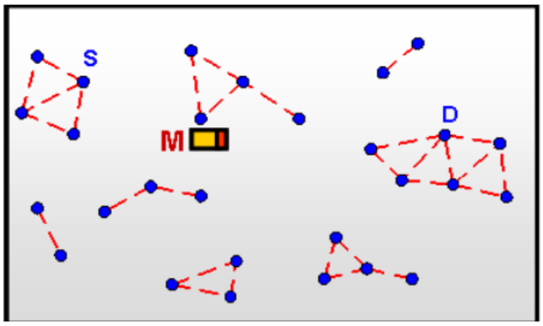


Figure 3.1: Single Copy Case

3.2 Background Related Information

3.2.1 Epidemic

The epidemic routing protocol is used because of its high message delivery rate and latency. It is a flooding-based routing, as nodes continuously replicate and transmit messages to newly found nodes that do not own a copy of the message yet.
In this flooding mechanism, each node replicates the message, and when their adjoining-nodes get nearer to their transmission range containing no replica of that particular message, they transmit the copies to them. Every node transmits the copies in the same way until the destination node is reached. This is called the message swap method in the Epidemic routing strategy. The method is shown in the below figure 3.2:

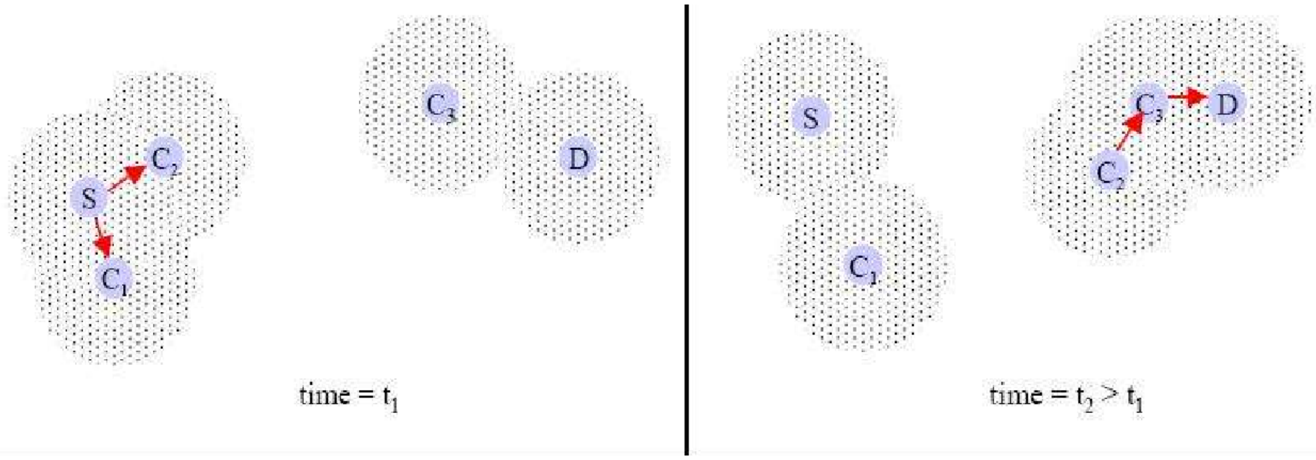


Figure 3.2: Epidemic

Figure 3.2 (In the left): S is a source, wishes to send a message to a destination, D, but no connected path is available between them. So, S transmits its messages to its two neighbors, C1 and C2, within direct communication range. Figure 3.2 (In the right): After a particular time, C2 comes into a direct transmission range with another node, C3, and transmits the message to it. And as C3 is in direct contact with D, the message is finally sent to the destination.

Every node contains a buffer to store the messages received from other nodes and hash tables where messages are listed (keyed by unique identifiers). Each node also stores a bit vector, called "summary vector," to identify the hash tables' entries. When two nodes are in communication, they exchange their summary vectors so that they can determine which messages stored remotely have not been seen by the local node and then requests that copies of the messages. To avoid redundant connections, each host also maintains a cache of hosts spoken with recently. Figure 3.3 explains the flowchart of Epidemic Routing Protocol.

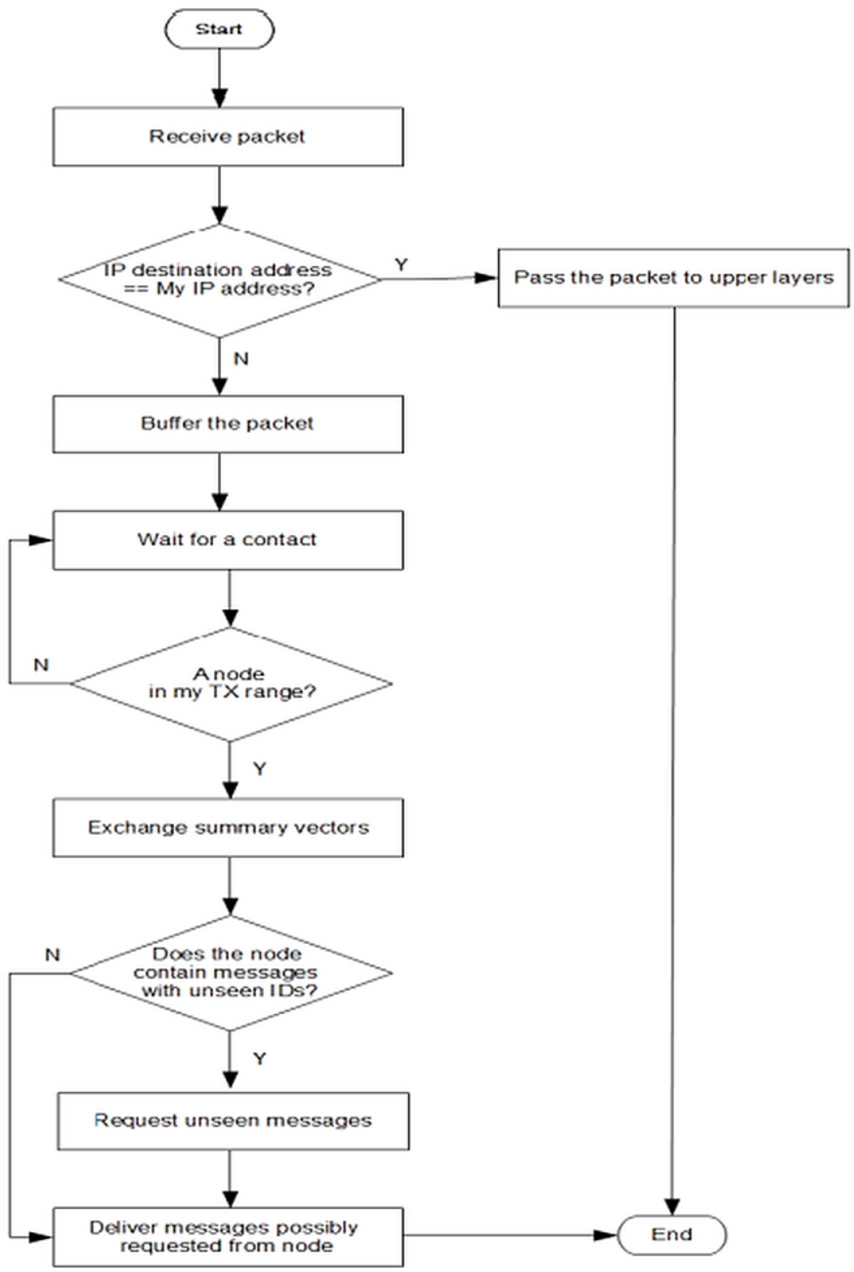


Figure 3.3: Flowchart of Epidemic [1]

The main goal of Epidemic Routing is to maximizing the message delivery rate, minimizing message delivery latency and also, reducing the total system resources consumed in message delivery. It focuses on the forwarding of messages based on the least possibility of network topology and connectivity. Given adequate buffer space and time, it guarantees eventual message delivery through such pair-wise message exchange.

3.2.2 Prophet

The basic functionality of PRoPHET routing is similar to that of Epidemic and seeks to improve the likelihood of message delivery.It is message forwarding which is based on the calculation of probability (also called delivery predictability) by every node to every destination node.When two nodes are encountered, messages are forwarded to a node which has higher delivery predictability.Figure 3.4 shows different stage of PRoPHET Routing Protocol.

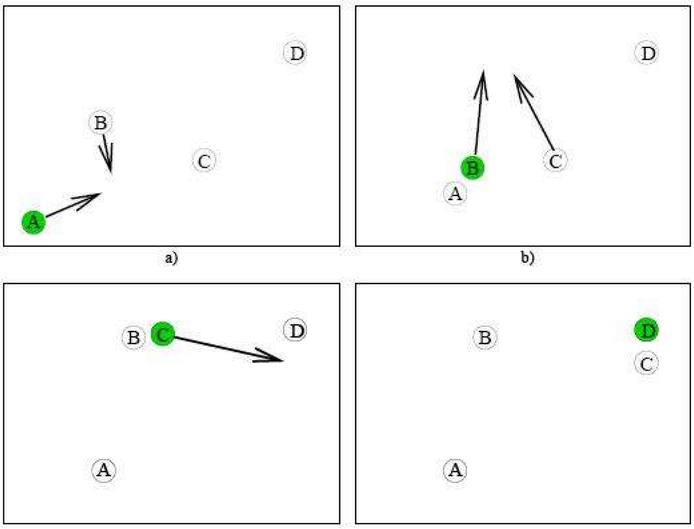


Figure 3.4: PRoPHET

The delivery predictability P (a, b) is stored in the internal delivery vector and updated every time the nodes meet. The delivery predictability used by each node are recalculated at each opportunistic encounter according to the following three rules:

- When node A meets another node B, the predictability for B is increased. Eq. 1 shows this calculation.

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

Where P_{init} is an initialization constant.

- The delivery predictability must age because if two nodes do not meet each other in a while, then they are less likely to relay messages to each other. Eq. 2 shows ageing equation.

$$P_{(a,b)} = P_{(a,b)old} * \gamma^k \quad (2)$$

Where γ^k is an aging constant.

- The delivery predictability also follows the transitive property, that is, if a node A frequently encounters node B and node B frequently encounters node D, then node D likely is a reliable node to forward message intended for node A. Eq. 3 shows the effect of transitivity on the delivery predictability.

$$P_{(a,d)} = P_{(a,d)old} + (1 - P_{(a,d)old}) * P_{(a,b)} * P_{(b,d)} * \beta \quad (3)$$

Where β is a scaling constant which decides what influence transitivity should have on the predictability of delivery.

A new type of routing protocol called PRoPHET has been proposed so that it can be improved the delivery probability and reduced the wastage of network resources in Epidemic routing. [12]

If a node in PROPHET has visited a location multiple times, there is a chance that the Pattern will repeat itself in the future. In PROPHET, each node uses the probability metric , known as delivery predictability, to deliver messages to a trusted node. The higher delivery predictability for a node indicates that it is more reliable node than other nodes to forward message to the destination. PROPHET outplays Epidemic routing.Figure 3.5 explains the flowchart of PRoPHET Routing Protocol.

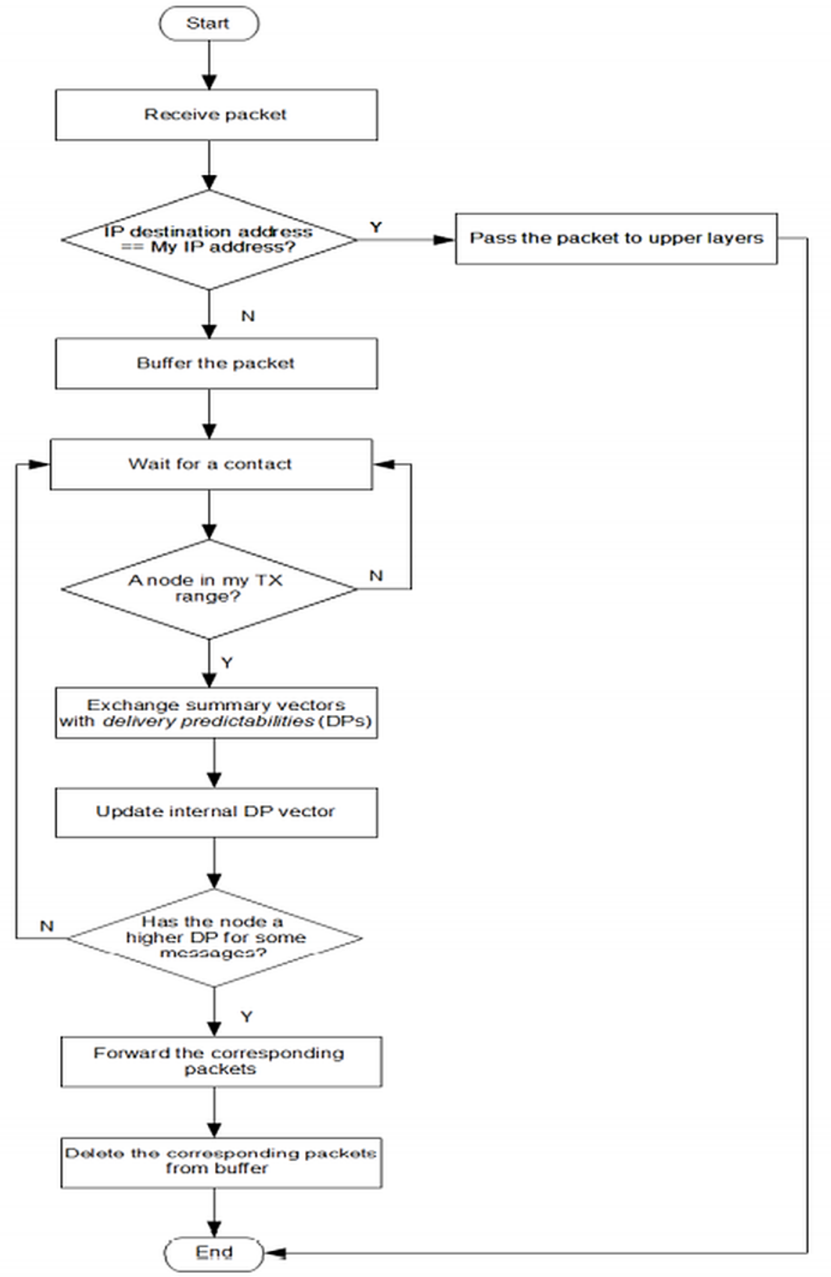


Figure 3.5: Flowchart of PRoPHET [1]

3.2.3 Spray and Wait

Spray and Wait is a protocol that combines the speed of epidemic routing with the simplicity of direct broadcast. It limits the number of copies distributed on the network and thus differs from the epidemic protocol. Messages are delivered through two phases in this protocol: the spray phase and the wait phase. Figure 3.6 explains the flowchart of Spray and Wait Routing Protocol.

Spray Phase: After generating a new message, the source node sprays a certain number (suppose L) of message copies to the same number of intermediate nodes it detects. The intermediate nodes receive the copies, and the system goes into the second phase, which is the Wait phase.

Wait Phase: The sprayed messages are stored in the buffers of the intermediate nodes. In the wait phase, the nodes wait for the destination to be encountered and finally deliver the message directly.

There are two modes of this protocol.

Source/Vanilla mode: In this mode, the source node transmits L-1 copies to the L-1 distinct nodes that it encounters first. All the nodes get one copy each. The source node also keeps one copy for itself. Now, these nodes are in the wait phase until one of them encounters the destination node.

Binary mode: In Binary mode, the source node starts with L copies of the original message. Now its task is to transfer half of the copies(L/2) to the nodes that are encountered with no copies. This process continues among all the nodes (source and relay) until they are left with only one copy. Whenever they have only one copy to themselves, they switch to direct transmission, which means they will be forwarded only to the destination.

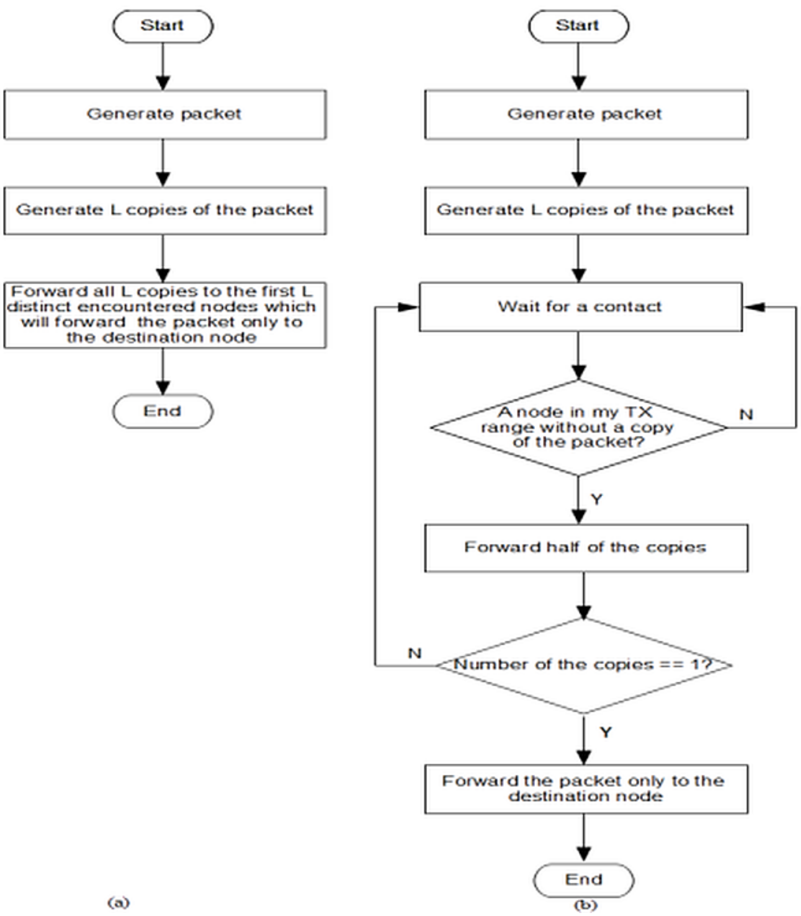


Figure 3.6: Flowchart of Spray and Wait Protocol [1]

3.2.4 MaxProp

MaxProp is a flood-based routing protocol, which means that each node distributes copies of every message, although they may not have any information about the network [13]. MaxProp solves one of the challenges of DTN routing protocols, which is buffer overload. Here, packets are sorted by highest to lowest priority based on the knowledge of previous encounter of the nodes which is mentioned as Delivery likelihood. The highest prioritized packets will be transmitted first during a transfer opportunity and lowest prioritized packets will be dropped first. In Figure 3.7 MaxProp Routing Strategy is described.

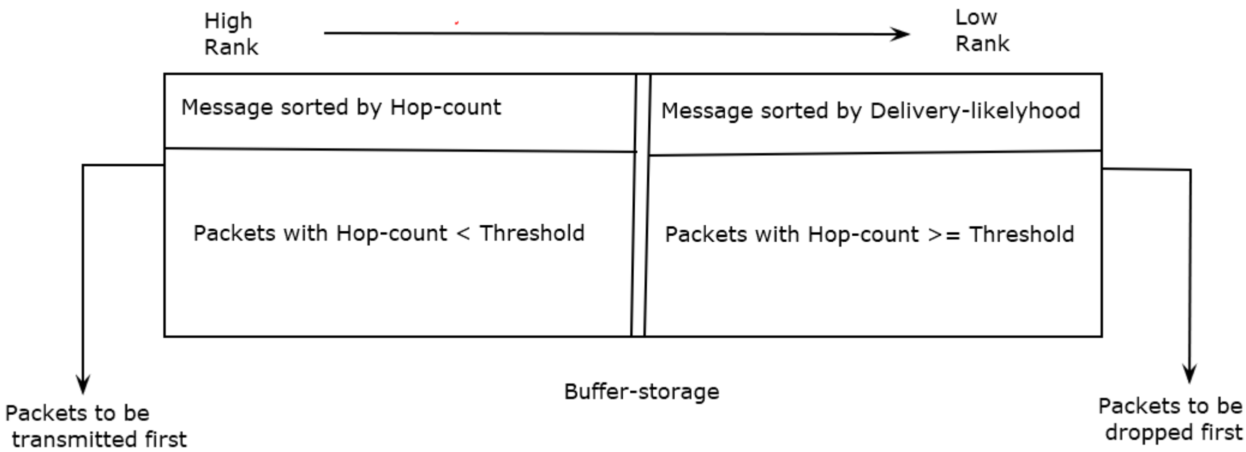


Figure 3.7: The MaxProp Routing Strategy

In MaxProp, each node is initialized with a probability of meeting all the other nodes in the network. When an encounter occurs between two nodes, their probability increases and the gets normalized alongside all the nodes in that network and they exchange a copy of these information aka delivery likelihood. This delivery likelihood is later used to find the shortest path among nodes using a variation of Dijkstra's algorithm.

Figure 3.8 explains the flowchart of MaxProp Routing Protocol.The lowest costed path among the possible paths is considered to be the destination path. If two or more packet's destination cost becomes equal, the packet with fewer hop-count gets priority.MaxProp uses acknowledgements to notify all the peers about packet deliveries and deletes messages that have been delivered. It assigns higher priority to new packets. It also ensures that the same packet is not received twice.

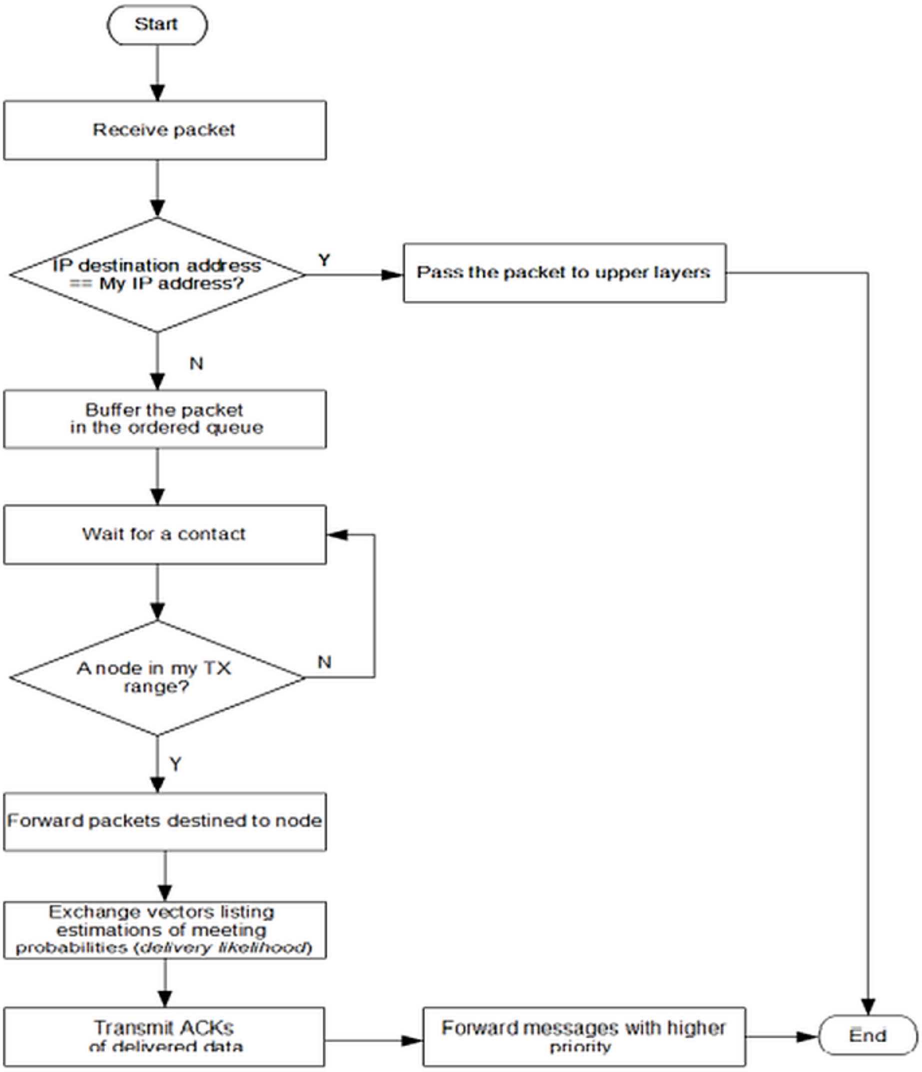


Figure 3.8: Flowchart of MaxProp [1]

.

Chapter 4

Proposal

In this chapter, we emphasis on the performance analysis of Epidemic, PRoPHET, Spray-and-Wait and MaxProp routing protocols in an intermittently connected mobile network. All routing protocols are simulated using Opportunistic Network Environment (ONE) [8] with version of 1.5.1. This section describes the environment modeling parameters, performance metrics, and the analysis of outcome of the simulations.

4.1 Overview

At its core, ONE is an agent-based distinct event simulation engine. At every simulation step the engine updates a number of modules that implement the main simulation functions. The major tasks of the ONE simulator are the modeling of node movement, inter-node contacts using several interfaces, routing, message handling and application interactions. The result collection and analysis are done through visualization, reports and post-processing tools. A detailed explanation of the simulator is available in [10] and the ONE simulator project page [32] where the source code is also available.

Figure 4.1 and 4.2 describe the overview concept of ONE Simulator and Screenshot of ONE Simulator respectively. Node movement is implemented by the movement models. There is connectivity between the nodes is based on their location, communication range and the bit-rate. The routing functions implemented by routing modules resolve which messages to forward over existing contacts. At last, the messages themselves are generated either through event generators that generate random traffic between the nodes, or through applications that produce traffic based on application interactions. The messages are also unicast, having a single source and destination host inside the simulation world. Simulation results are gathered primarily through reports generated by report modules during the simulation run. Report modules receive events from the simulation engine and generate results based on them. The results developed may be logs of events that are then further processed by the external post-processing tools, or they may be aggregate statistics calculated in the simulator. Secondarily, the graphical user interface (GUI) displays a visualization of the simulation state to presentation the locations, active contacts and messages carried by the nodes.

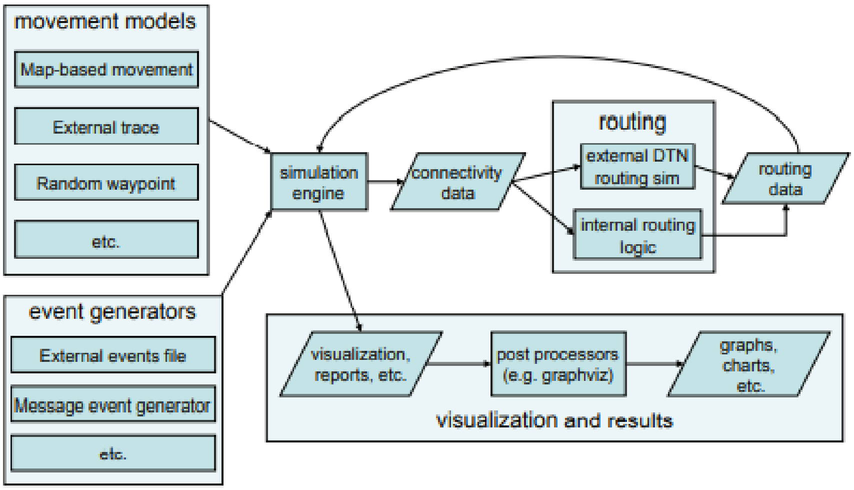


Figure 4.1: Overview of One Simulator

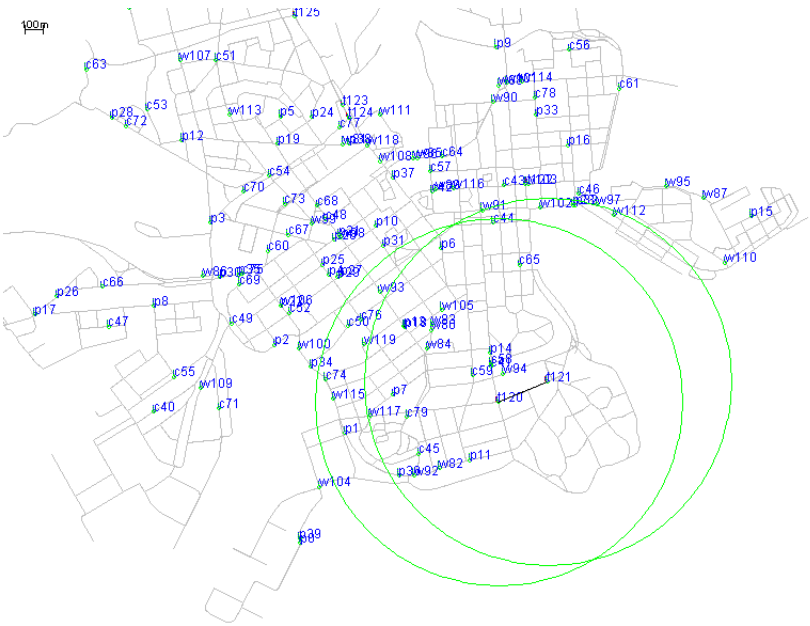


Figure 4.2: Screenshot of One Simulator

4.2 Proposed Method

The three metrics to measure the performance of the different protocols are described below.

- **Delivery Probability:** The delivery probability is the ratio of total number of messages delivered to their destination to total number of generated messages at source node.

  $$\text{Delivery ratio} = \frac{D}{G} \quad (1)$$

  where **D** is a number of messages delivered to destination, and **G** is a number of created messages.

- **Overhead Ratio:** The overhead ratio indicates how many redundant packets are relayed to deliver one packet. It simply discloses transmission cost in a network.

  $$\text{Overhead Ratio} = \frac{R-D}{D} \quad (2)$$

  where **R** is a number of messages forwarded by relay nodes, and **D** is a number of messages delivered or conveyed to their destination.

- **Average Latency:** The average latency is the time between messages is created and the messages are received at a destination.

  $$\text{Average Latency} = \frac{\sum_{i=1}^{n}(R_i - G_i)}{n} \quad (3)$$

  where **n** is a number of messages delivered to their destinations, is the time when a message **i** reaches to its destination, and is the time when a message i is created.

4.3 Summary of Proposed Method

It is one kind of simulation tool which is developed in Java and used for analyzing DTN routing protocol. The entire system is developed in 3 steps as described below-

- Initially we write the implementation of DTN routing protocol in the form of JAVA classes which are provided into ONE Simulator, which itself is a collection of java classes and packages
- Then ONE simulates the protocol as per our specifications
- As a consequence, it generates a detailed report on performance evaluation of the protocol as per simulation

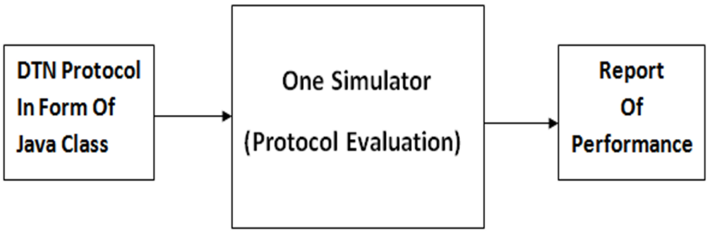


Figure 4.3: Methodology

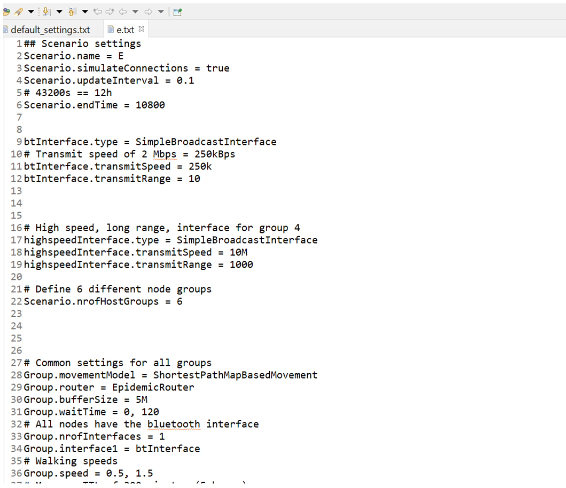


Figure 4.4: Specification Settings

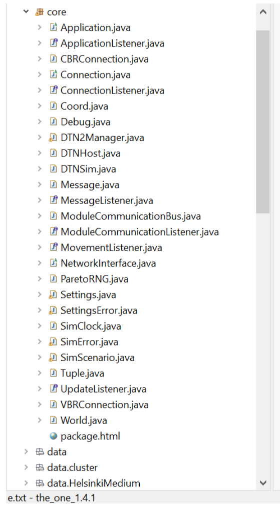


Figure 4.5: Java Class

In Figure 4.3, our proposed methodology is shown above. Figure 4.4 and 4.5 represent some figures of our Specification Settings and Java Class which are mentioned above.

Chapter 5

Implementation

5.1 Simulation Environment Setup

Parameters of simulation setup and routing algorithms are defined in Table 5.1, Table 5.2, Table 5.3, Table 5.4,Table 5.5,Table 5.6 and Table 5.7 respectively. For varying simulation time, the buffer size is 5MB and the total number of nodes is 126, where 80 pedestrians (two groups with 40 nodes in each group), 40 cars (single group of 40 nodes) and 6 trams (3 groups with 2 nodes in every group). Again, for varying the buffer size, simulation time is 10800s or 3 hrs. Again, for varying the transmit range, buffer size is 5MB and total simulation time is 43200s or 12 hrs. For varying TTL and number of nodes, buffer size is 5MB, transmit range is 10m and simulation time is 43200s. Table 5.7 summarizes the simulation configuration for routing algorithms.For all the Simulation Setup, Mobility is **Random Way Point**, Movement Model is **Shortest Path Map Based**, Message Size is **500KB to 1MB**, Simulation Area Size is **4500m*3400m**.

5.1.1 Simulation Setup for varying Transmit Range :

Here, we vary Transmit Range with respect to all other Simulation Parameters.The constant Simulation Time is 43200 seconds and Buffer Size is 5MB.

Parameters	Values
Simulation Time (In Seconds)	43200
Update Interval	0.1 second
Number of Nodes in Group	126
Interface	Bluetooth Interface
Interface Type	Simple Broadcast Interface
Transmit Speed	250 kbps
Transmit Range	10m, 12m, 14m, 15m, 16m, 18m, 20m
Routing Protocols	Epidemic, PRoPHET, Spray-and-wait, MaxProp
Buffer Size	5MB
Message Generation Rate (message/min.)	2 (One new message in every 25 to 35 seconds)
Message TTL	300 minutes
Mobility	Random Way Point
Movement model	Shortest Path Map Based
Message Size	500 KB – 1 MB
Simulation Area Size	4500m * 3400m

Table 5.1: Simulation Parameters and Values for Varying Transmit Range

5.1.2 Simulation Setup for varying Simulation Time:

Here, we vary Simulation Time with respect to all other Simulation Parameters.The constant Transmit Range is 10m and Buffer Size is 5MB.

Parameters	Values
Simulation Time (In Seconds)	5000, 7500, 10000, 12500, 15000, 17500, 20000
Update Interval	0.1 second
Number of Nodes in Group	126
Interface	Bluetooth Interface
Interface Type	Simple Broadcast Interface
Transmit Speed	250 kbps
Transmit Range	10m
Routing Protocols	Epidemic, PRoPHET, Spray-and-wait, MaxProp
Buffer Size	5MB
Message Generation Rate (message/min.)	2 (One new message in every 25 to 35 seconds)
Message TTL	300 minutes

Table 5.2: Simulation Parameters and Values for Varying Simulation Time

5.1.3 Simulation Setup for varying Buffer Size:

Here, we vary Buffer Size with respect to all other Simulation Parameters.The constant Simulation Time is 10800 seconds and Transmit Range is 10m.

Parameters	Values
Simulation Time (In Seconds)	10800
Update Interval	0.1 second
Number of Nodes in Group	126
Interface	Bluetooth Interface
Interface Type	Simple Broadcast Interface
Transmit Speed	250 kbps
Transmit Range	10m
Routing Protocols	Epidemic, PRoPHET, Spray-and-wait, MaxProp
Buffer Size	5, 10, 15, 20, 25, 30,35, 40, 45, 50 (In MB)
Message Generation Rate (message/min.)	2 (One new message in every 25 to 35 seconds)
Message TTL	300 minutes

Table 5.3: Simulation Parameters and Values for varying Buffer Size

5.1.4 Simulation Setup for varying Message Generation Rate (message/min):

Here, we vary Message Generation Rate with respect to all other Simulation Parameters.The constant Simulation Time is 43200 seconds, Transmit Range is 10m and Buffer Size is 5M.

Parameters	Values
Simulation Time (In Seconds)	43200
Update Interval	0.1 second
Number of Nodes in Group	126
Interface	Bluetooth Interface
Interface Type	Simple Broadcast Interface
Transmit Speed	250 kbps
Transmit Range	10m
Routing Protocols	Epidemic, PRoPHET, Spray-and-wait, MaxProp
Buffer Size	5MB
Message Generation Rate (message/min.)	1,2,3,4,5,6,10 (in Minute(s))
Message TTL	300 minutes

Table 5.4: Simulation Parameters and Values for varying Message Generation Rate

5.1.5 Simulation Setup for varying TTL (In minutes):

Here, we vary TTL (in minutes) with respect to all other Simulation Parameters.The constant Simulation Time is 43200 seconds, Transmit Range is 10m and Buffer Size is 5M.

Parameters	Values
Simulation Time (In Seconds)	43200
Update Interval	0.1 second
Number of Nodes in Group	126
Interface	Bluetooth Interface
Interface Type	Simple Broadcast Interface
Transmit Speed	250 kbps
Transmit Range	10m
Routing Protocols	Epidemic, PRoPHET, Spray-and-wait, MaxProp
Buffer Size	5MB
Message Generation Rate (message/min.)	2 (One new message in every 25 to 35 seconds)
Message TTL	60, 120, 180, 240, 300, 360, 420 (in minutes)

Table 5.5: Simulation Parameters and Values for varying TTL

5.1.6 Simulation Setup for varying Number of Nodes:

Here, we vary Number of Nodes with respect to all other Simulation Parameters.The constant Simulation Time is 43200 seconds, Transmit Range is 10m and Buffer Size is 5M.

Parameters	Values
Simulation Time (In Seconds)	43200
Update Interval	0.1 second
Number of Nodes in Group	126,156,186,216,246,276,306,336
Interface	Bluetooth Interface
Interface Type	Simple Broadcast Interface
Transmit Speed	250 kbps
Transmit Range	10m
Routing Protocols	Epidemic, PRoPHET, Spray-and-wait, MaxProp
Buffer Size	5MB
Message Generation Rate (message/min.)	2 (One new message in every 25 to 35 seconds)
Message TTL	300(in minutes)

Table 5.6: Simulation Parameters and Values for varying Number of Nodes

5.2 Parameters of Routing Algorithms

Routing Algorithms	Parameters	Values
Epidemic	-	-
PRoPHET	Seconds in Time Unit	30 second
Spray and Wait	No. of Copies (L)	6
MaxProp	-	-

Table 5.7: Parameters of Routing Algorithms

Chapter 6

Experiment Result

6.1 Impact of Varying Transmit Range

Here, the routing algorithms are evaluated by varying the transmit range from 10m to 20m. We have tried on a scenario containing 126 nodes, messages are generated in every 25 to 35 seconds and total simulation time is 43200s or 12 hrs.

6.1.1 Impact on Average Latency

Decreasing the delay is one of the challenges in DTN. It is the time between message generations to its reception. The algorithm whose average latency decreases the most with transmission range performs better.

As the transmit range increases, the average latency of all the 4 routing algorithms decreases slowly. It is mainly because as the transmit range increases, it takes fewer hops to the base station. Though latency decreases with transmit range for all 4 algorithms, it decreases the most for the MaxProp algorithm. MaxProp's average latency decreased more than 50% with transmit range, But as the Latency was already lower for Spray and Wait, it outperforms the other 3 algorithms. The graph of Average Latency vs Transmit Range is shown in Figure 6.1.

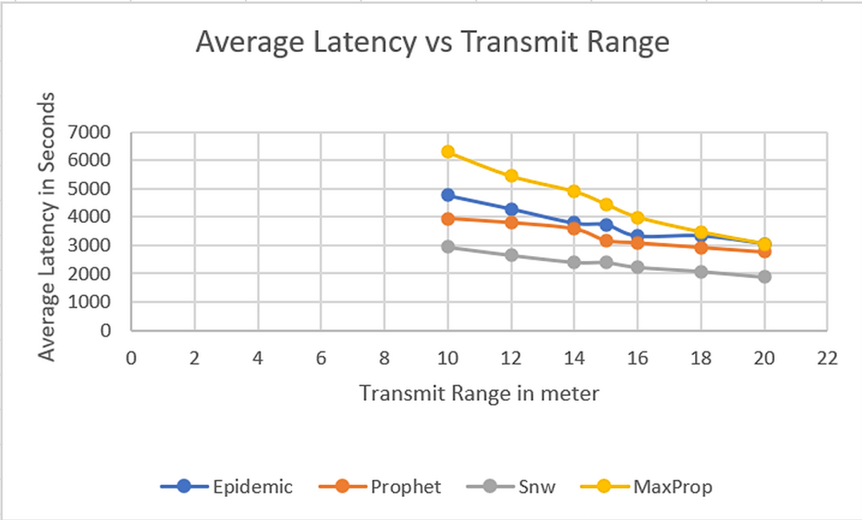


Figure 6.1: Average Latency vs Transmit Range

Summary :

Transmit Range	Average_Latency			
	Epidemic	Prophet	Snw	MaxProp
10	4775.294	3950.935	2937.356	6300.788
12	4294.167	3805.063	2650.814	5450.779
14	3785.389	3586.363	2406.577	4903.43
15	3731.238	3179.728	2396.106	4451.834
16	3321.994	3091.096	2231.252	3989.701
18	3333.231	2931.187	2082.968	3462.829
20	3049.37	2770.107	1885.184	3044.195

Figure 6.2: Avg Latency vs Transmit Range Graph Value

From Figure 6.2 we see that the value of Epidemic starts from 4775.294 at Transmit Range 10 and it gradually decreases. At last, it becomes 3049.37 at Transmit Range 20. For PRoPHET, we see that the value of avg latency starts from 3950.935 at Transmit Range 10 and it also decreases gradually. Finally, it becomes 2770.107 at Transmit Range 20. For Spray and Wait, we found that it starts from 2937.356 and ends in 1885.184 at our specified settings. For MaxProp, it starts from 6300.788 and finishes at 3044.195 for Transmit Range 20. In Addition, we can see that the value of Average Latency is the lowest for Spray and Wait Routing Protocol for our specified settings.

6.1.2 Impact on Delivery probability

Delivery probability generally increases with increased transmission range. Because increased transmission range results in increased number of neighbors, as a result, nodes can deliver the message as close as possible by choosing the most appropriate hop. So, Delivery probability increases with the increasing transmit range for all 4 routing algorithms.

The delivery probability of Epidemic and PRoPHET algorithms increases moderately with increased transmission range. But for MaxProp, it increases almost 5 times of that increment of the previous two algorithms. Spray and wait also gave increased delivery probability, it stood just behind MaxProp. So, in this case, MaxProp gives better output among other algorithms. The graph of Delivery Probability vs Transmit Range is shown in Figure 6.3.

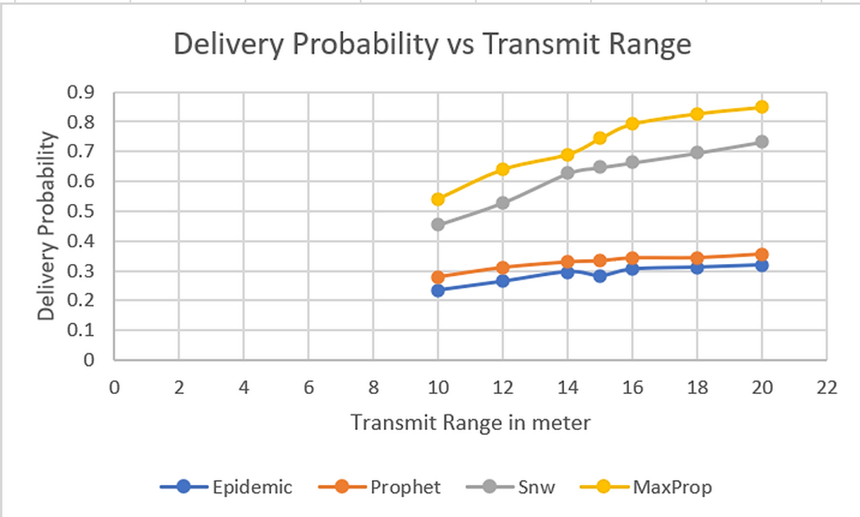


Figure 6.3: Delivery Probability vs Transmit Range

Summary :

Transmit Range	Delivery-Probability			
	Epidemic	Prophet	Snw	MaxProp
10	0.2348	0.2786	0.4545	0.54
12	0.2642	0.3101	0.5291	0.6393
14	0.2957	0.3285	0.627	0.69
15	0.2827	0.3326	0.6468	0.744
16	0.3046	0.342	0.6632	0.7927
18	0.3107	0.3429	0.6961	0.8275
20	0.3183	0.3552	0.7331	0.8494

Figure 6.4: Delivery Probability vs Transmit Range Graph Value

From Figure 6.4 we see that the value of Epidemic starts from 0.2348 at Transmit Range 10 and it gradually increases. At last, it becomes 0.3183 at Transmit Range 20. For PRoPHET, we see that the value of Delivery Probability starts from 0.2786 at Transmit Range 10 and it also increases gradually. Finally, it becomes 0.3552 at Transmit Range 20. For Spray and Wait, we found that it starts from 0.4545 and ends in 0.7331 at our specified settings. For MaxProp, it starts from 0.54 and finishes at 0.8494 for Transmit Range 20. In Addition, we can see that the value of Delivery Probability is the highest for MaxProp Routing Protocol for our specified settings.

6.1.3 Impact on Overhead Ratio

Overhead ratio reflects the cost of a network. Lesser Overhead defines the efficient protocol. Overhead ratio increases a lot for PROPHET and EPIDEMIC protocols as the transmit range increases.

It also increased for the MaxProp algorithm. But its initial overhead ratio was already lower than the other 2 algorithms and, it also increased lesser then the previous 2 algorithms. So, it wasn't the worst.But surprisingly Overhead ratio which was already low for Spray and Wait algorithm, instead of increasing like other 2 algorithms, it decreases. It is undoubtedly a plus point for Spray and Wait routing algorithm. The graph of Overhead Ratio vs Transmit Range is shown in Figure 6.5.

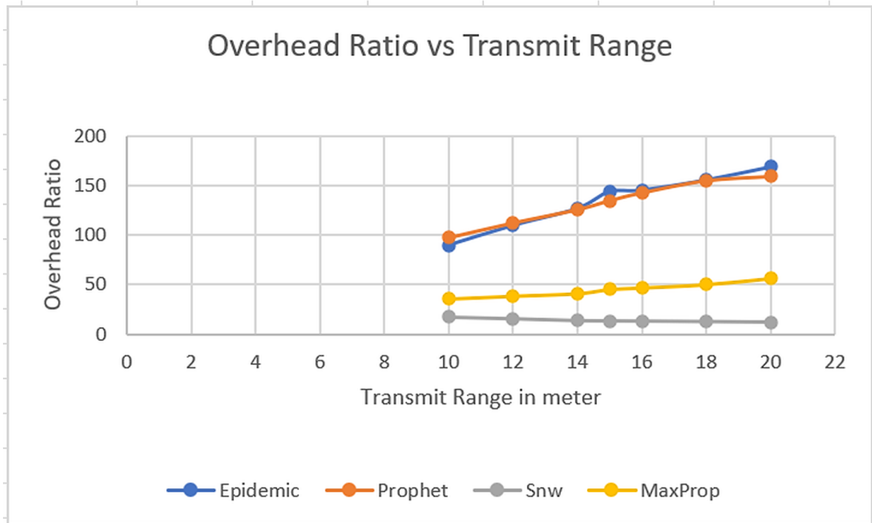


Figure 6.5: Overhead Ratio vs Transmit Range

Summary :

From Figure 6.6 we see that the value of Epidemic starts from 89.3965 at Transmit Range 10 and it gradually increases. At last, it becomes 169.2989 at Transmit Range 20. For PRoPHET, we see that the value of Overhead Ratio starts from 97.5061 at Transmit Range 10 and it also increases gradually. Finally, it becomes 159.2408 at Transmit Range 20. For Spray and Wait, we found that it starts from 17.4729 and ends in 11.8637 at our specified settings. For MaxProp, it starts from 35.7136 and finishes at 56.0048 for Transmit Range

Transmit Range	Overhead_ratio			
	Epidemic	Prophet	Snw	MaxProp
10	89.3965	97.5061	17.4729	35.7136
12	109.5311	112.3245	15.6688	38.4893
14	126.8519	125.7938	13.5622	41.0322
15	144.0339	134.8045	13.2148	45.0745
16	145.6427	142.7237	12.9763	46.7072
18	155.9383	154.9681	12.4661	49.9785
20	169.2989	159.2408	11.8637	56.0048

Figure 6.6: Overhead Ratio vs Transmit Range Graph Value

20. In Addition, we can see that the value of Overhead Ratio is the lowest for Spray and Routing Protocol for our specified settings.

6.2 Impact of Varying Simulation Time

Here, we observed the algorithms by varying the simulation time from 5000 seconds to 20000 seconds. We have tried on a scenario containing 126 nodes and messages are generated in every 25 to 35 seconds.

6.2.1 Impact on Average Latency

For Spray and Wait, average latency increases a bit as the simulation time progresses. Then it stays almost consistent. Average latency increases for PROPHET, Epidemic and MaxProp too. For MaxProp, the average latency increases more than 4 times than Spray and Wait while for Epidemic, latency increases more than 3 times of Spray and Wait, and almost 2 times of PROPHET with the simulation time. In this case, MaxProp performs the worst and Spray and Wait performs better than other three algorithms. The graph of Average Latency vs Simulation Time is shown in Figure 6.7.

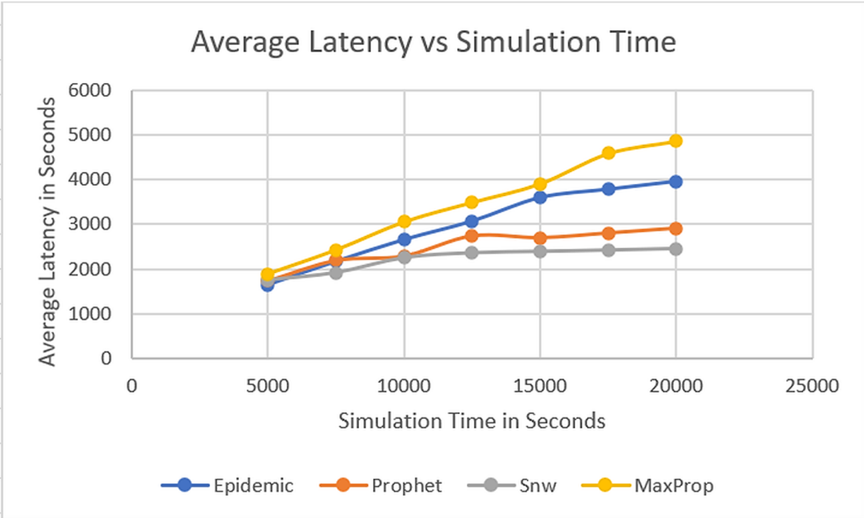


Figure 6.7: Average Latency vs Simulation Time

Summary :

Simulation Time	Average_Latency			
	Epidemic	Prophet	Snw	MaxProp
5000	1647.271	1740.748	1746.776	1891.709
7500	2169.251	2189.74	1911.433	2427.723
10000	2655.567	2288.303	2247.598	3053.475
12500	3068.962	2731.083	2356.344	3487.733
15000	3595.742	2691.71	2386.939	3904.567
17500	3780.767	2795.092	2416.835	4581.655
20000	3959.106	2901.445	2449.989	4856.036

Figure 6.8: Average Latency vs Simulation Time Graph Value

From Figure 6.8 we see that the value of Epidemic starts from 1647.271 at Simulation Time 5000s and it gradually increases. At last, it becomes 3959.106 at Simulation Time 20000s. For PRoPHET, we see that the value of Average Latency starts from 1740.748 at Simulation Time 5000s and it also increases gradually. Finally, it becomes 2901.445 at Simulation Time 20000s. For Spray and Wait, we found that it starts from 1746.776 and ends in 2449.989 at our specified settings. For MaxProp, it starts from 1891.709 and finishes at 4856.036 for Simulation Time 20000s. In Addition, we can see that the value of Average Latency is the lowest for Spray and Wait Routing Protocol for our specified settings.

6.2.2 Impact on Delivery Probability

When simulation times increased, the delivery probability of MaxProp, Epidemic and Spray and Wait increased. For MaxProp, the delivery probability increased the most. The initial delivery probability of MaxProp and Spray and Wait was almost same. But as time progresses, MaxProp's delivery ratio increased and become almost double than its initial probability.

For Spray and Wait, delivery probability increased nearly 3 times than what increased for Epidemic.

For PROPHET, delivery probability was increasing at first, but after a little while, it started to decrease. Finally, it decreased than what it was at the beginning. But its delivery probability is still higher than Epidemic. As node forwards messages in a flooding way in Epidemic, when message buffer is limited, the delivery ratio will be low.

So, in this case, MaxProp showed the best result, Spray and Wait was also good, but PROPHET did the worst comparing with other algorithms.The graph of Delivery Probability vs Simulation Time is shown in Figure 6.9.

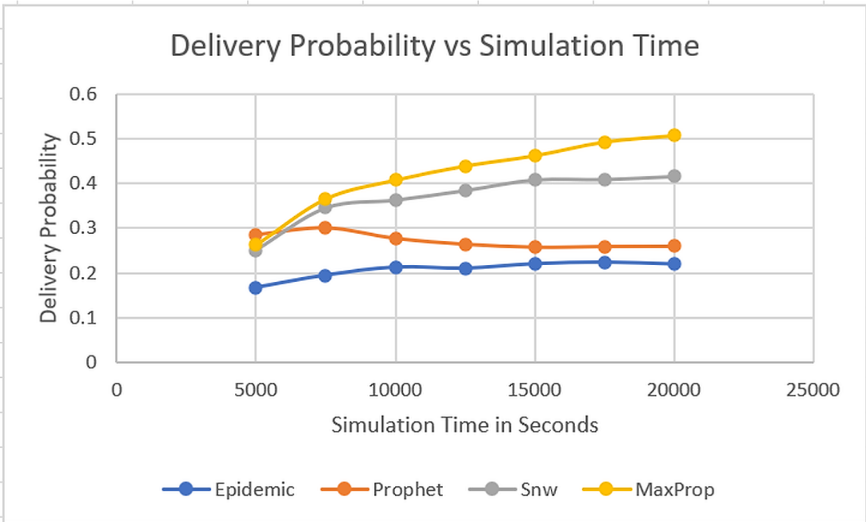


Figure 6.9: Delivery Probability vs Simulation Time

Summary :

Simulation Time	Delivery-Probability			
	Epidemic	Prophet	Snw	MaxProp
5000	0.1667	0.2857	0.25	0.2619
7500	0.1944	0.3016	0.3452	0.3651
10000	0.2124	0.2773	0.3628	0.4071
12500	0.2099	0.2642	0.3844	0.4387
15000	0.22	0.2574	0.4086	0.4617
17500	0.2234	0.2589	0.4095	0.4924
20000	0.2193	0.2593	0.4163	0.5067

Figure 6.10: Delivery Probability vs Simulation Time Graph Value

From Figure 6.10 we see that the value of Epidemic starts from 0.1667 at Simulation Time 5000s and it gradually increases. At last, it becomes 0.2193 at Simulation Time 20000s.

For PRoPHET, we see that the value of Delivery Probability starts from 0.2857 at Simulation Time 5000s and then it increases first and later decreases. Finally, it becomes 0.2593 at Simulation Time 20000s. For Spray and Wait, we found that it starts from 0.25 and ends in 0.4163 at our specified settings. For MaxProp, it starts from 0.2619 and finishes at 0.5067 for Simulation Time 20000s. In Addition, we can see that the value of Delivery Probability is the highest for MaxProp Routing Protocol for our specified settings.

6.2.3 Impact on Overhead Ratio

Under the default settings, Spray and Wait had the lowest overhead ratio which decreased a little bit as the simulation time increased at first. Then it remained almost constant. This is mainly because Spray and Wait controls flooding by limiting the number of bundle copies created per bundle.

For MaxProp, the overhead ratio decreased to about half of its initial overhead ratio. It is a good sign for MaxProp. But as at initial, it had greater overhead ratio than Spray and Wait, it couldn't outperform Spray and Wait.

The overhead ratio of the Epidemic was higher than the other two as flooding causes high overhead because of the number of copied messages in the network, which can congest the network. It shows some decrement as the simulation time progressed at first, but after longer period, the overhead ratio becomes almost the same, what it was at the starting phase.

Thing happened differently for the PRoPHET one. As simulation time increased, the overhead ratio of the PHOPHET increased significantly. In this case, Epidemic was moderate, PROPHET's result worsened, but Spray and Wait performed better than others. The graph of Overhead Ratio vs Simulation Time is shown in Figure 6.11.

Summary :

From Figure 6.12 we see that the value of Epidemic starts from 100.6786 at Simulation Time 5000s and it gradually decreases. At last, it becomes 96.6689 at Simulation Time 20000s. For PRoPHET, we see that the value of Overhead Ratio starts from 44.1667 at Simulation Time 5000s and it also increases gradually. Finally, it becomes 101.16 at Simulation Time 20000s. For Spray and Wait, we found that it starts from 24.6667 and ends in 18.4021 at our specified settings. For MaxProp, it starts from 60.0682 and finishes at 36.6988 for Simulation Time 20000s. In Addition, we can see that the value of Overhead Ratio is the lowest for Spray and Wait Routing Protocol for our specified settings.

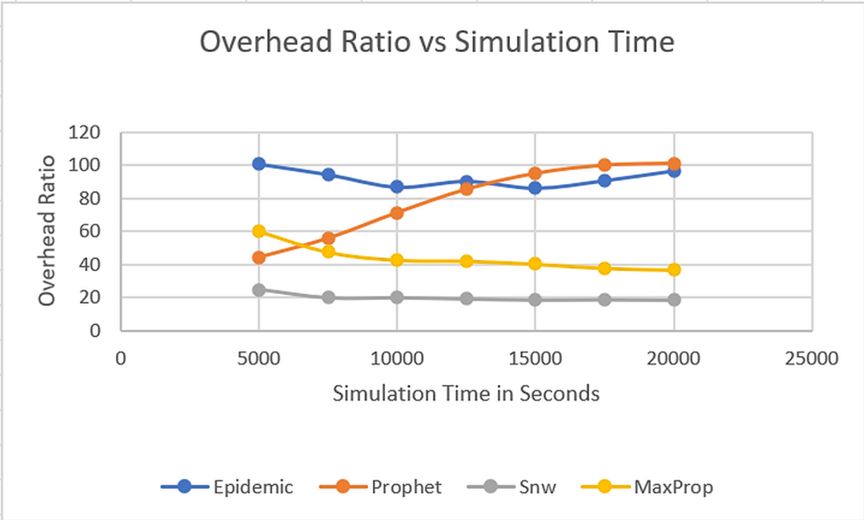


Figure 6.11: Overhead Ratio vs Simulation Time

Simulation Time	Overhead_ratio			
	Epidemic	Prophet	Snw	MaxProp
5000	100.6786	44.1667	24.6667	60.0682
7500	94.3673	55.7237	19.7816	47.4348
10000	86.7917	71.3085	19.8455	42.6594
12500	90.3371	85.4286	19.0982	42.0914
15000	86.1607	95.0305	18.4471	40.2553
17500	90.75	100.0327	18.6198	37.6907
20000	96.6689	101.16	18.4021	36.6988

Figure 6.12: Overhead Ratio vs Simulation Time Graph Value

6.3 Impact of Varying Buffer Size

In DTN, it is not known when an exchange opportunity will arrive until messages have to be stored in a buffer. So, the buffer needs to be strong enough to store the pending messages. Here, we observed the algorithms by increasing the buffer size from 5 MB to 50 MB. We have tried on a scenario containing 126 nodes, and messages are generated every 25 to 35 seconds.

6.3.1 Impact on Average Latency

As there is no effective buffer management scheme in Spray and Wait, Epidemic and PROPHET algorithms, as buffer size increases, their average Latency increases too. As buffer size increases, packet drop reduces. As a result, packets with larger hop count also reach the destination, increasing the average Latency to some extent.

But MaxProp has an active buffer management system. It determines which messages should be send first and which should be dropped first when the buffer gets full. So, average latency decreases for MaxProp with increased buffer size.The graph shows, with the increased buffer size, the average Latency of Spray and Wait increases the most, especially in between 10MB and 15MB. After that, the graph becomes a bit consistent. For Epidemic, the Latency was already higher as it forwards messages in flooding nature without maintaining any methodology. With increased buffer size, the average Latency increases too.

The PROPHET decreases its average Latency from 10MB to 15MB (where Spray and Wait increased the more), then it starts increasing till 20MB and later becomes consistent. For MaxProp, the average latency was almost the same while varying the buffer size, and it also decreased a little bit in some cases. So, MaxProp performed better than other 3 algorithms in this case. The graph of Average Latency vs Buffer Size is shown in Figure 6.13.

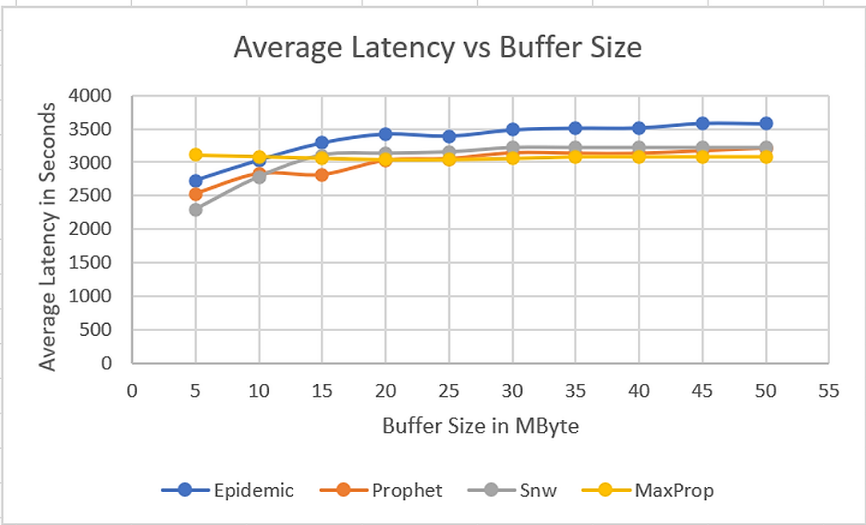


Figure 6.13: Average Latency vs Buffer Size

Summary :

Buffer Size	Average_Latency			
	Epidemic	Prophet	Snw	MaxProp
5	2728.826	2532.472	2300.186	3108.176
10	3033.302	2830.488	2783.961	3086.328
15	3291.735	2815.445	3106.527	3062.947
20	3425.477	3026.042	3132.684	3041.376
25	3392.837	3050.454	3152.613	3042.605
30	3487.337	3138.351	3217.288	3059.565
35	3513.283	3130.846	3217.288	3083.811
40	3516.164	3128.856	3217.288	3083.811
45	3584.901	3169.91	3217.288	3083.811
50	3577.013	3208.586	3217.288	3083.811

Figure 6.14: Average Latency vs Buffer Size Graph Value

From Figure 6.14 we see that the value of Epidemic starts from 2728.826 at Buffer Size 5MB and it gradually increases. At last, it becomes 3557.013 at Buffer Size 50MB. For PRoPHET, we see that the value of Average Latency starts from 2532.472 at Buffer Size 5000s and it also increases gradually. Finally, it becomes 3208.586 at Buffer Size 50MB. For Spray and Wait, we found that it starts from 2300.186 and ends in 3217.288 at our specified settings. For MaxProp, it starts from 3108.176 and finishes at 3083.811 for Simulation Time 50MB. In Addition, we can see that the value of Average Latency is the lowest for MaxProp Routing Protocol for our specified settings.

6.3.2 Impact on Delivery Probability

When the buffer space is low, it becomes full easily, so message dropping becomes more frequent. As buffer space gets bigger, the quantity of message droppings gets lesser, and the quantity of exchanged messages gets higher. As delivery probability depends on the number of exchanges or received messages, delivery probability gets higher with increased buffer size. That's why all the three algorithms showed increased delivery probability with increased buffer size.

For MaxProp, the delivery probability was already higher; it also increased quite a bit with buffer size increment. The graph of Spray and Wait also shows increment in delivery probability, just a little less MaxProp's. Epidemic also showed quite increments, but as its delivery probability was already lower than the other two, it didn't manage to copy up with PROPHET. The graph of Delivery Probability vs Buffer Size is shown in Figure 6.15.

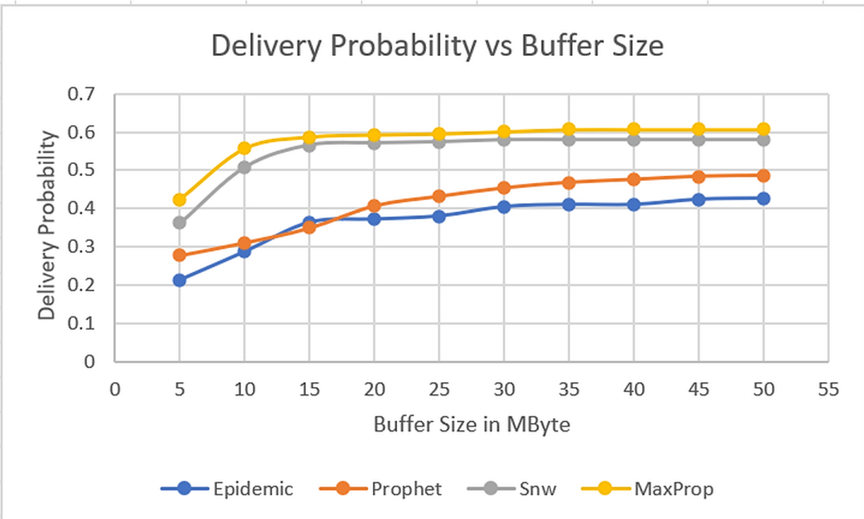


Figure 6.15: Delivery Probability vs Buffer Size

Summary :

From Figure 6.16 we see that the value of Epidemic starts from 0.2131 at Buffer Size 5MB

Buffer Size	Delivery-Probability			
	Epidemic	Prophet	Snw	MaxProp
5	0.2131	0.2787	0.3634	0.4235
10	0.2869	0.3115	0.5082	0.5574
15	0.3634	0.3497	0.5656	0.5874
20	0.3716	0.4071	0.571	0.5929
25	0.3798	0.4317	0.5738	0.5956
30	0.4044	0.4536	0.5792	0.6011
35	0.4098	0.4678	0.5792	0.6066
40	0.4098	0.4754	0.5792	0.6066
45	0.4235	0.4836	0.5792	0.6066
50	0.4262	0.4859	0.5792	0.6066

Figure 6.16: Delivery Probability vs Buffer Size Graph Value

and it gradually increases. At last, it becomes 0.4262 at Buffer Size 50MB. For PRoPHET, we see that the value of Delivery Probability starts from 0.2787 at Buffer Size 5MB and it also increases gradually. Finally, it becomes 0.4859 at Buffer Size 50MB. For Spray and Wait, we found that it starts from 0.3634 and ends in 0.5792 at our specified settings. For MaxProp, it starts from 0.4235 and finishes at 0.6066 for Buffer Size 50MB. In Addition, we can see that the value of Delivery Probability is the highest for MaxProp Routing Protocol for our specified settings.

6.3.3 Impact on Overhead Ratio

As increased buffer size increases the number of exchanged and delivered messages, it results in a decrement in overhead ratio. The overhead ratio of Spray and Wait was already low than the other two. With increased buffer size, its overhead ratio dropped a bit then becomes almost consistent. It showed the best performance in this case. MaxProp also showed fine decrement in overhead ratio with increased buffer size.

The overhead ratio in both PROPHET and Epidemic decreases too. But even though Epidemic had the highest Overhead at first, it decreased the most and became even lower than PRoPHET. PRoPHET's overhead ratio increased slightly from 5MB to 10MB, then it started to fall, but Epidemic outperformed PROPHET in this case. The graph of Overhead Ratio vs Buffer Size is shown in Figure 6.17.

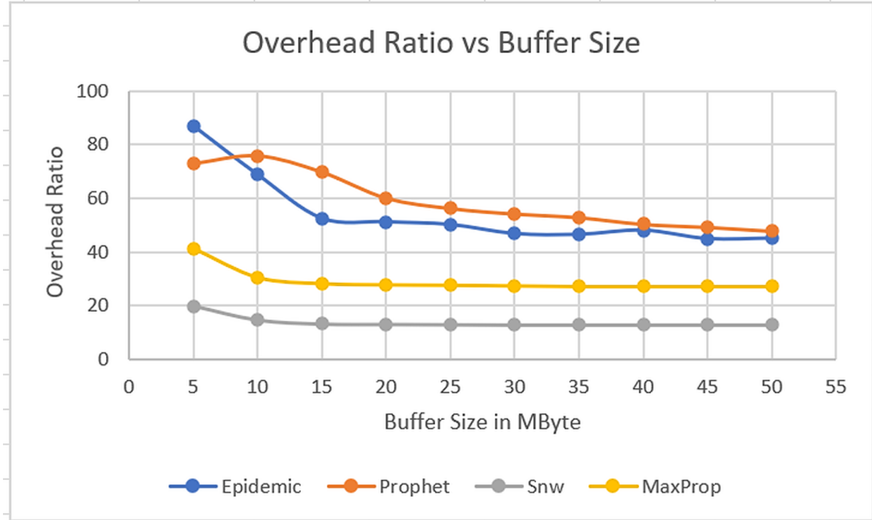


Figure 6.17: Overhead Ratio vs Buffer Size

Summary :

Buffer Size	Overhead_ratio			
	Epidemic	Prophet	Snw	MaxProp
5	86.9744	72.8137	19.8797	41.3548
10	68.819	75.7105	14.7366	30.6324
15	52.5038	69.7266	13.2609	28.3488
20	51.1838	60.094	13.134	27.9078
25	50.1799	56.2658	13.0429	27.7615
30	46.7905	54.2169	12.9198	27.5
35	46.4733	52.8972	12.9198	27.2613
40	48.0667	50.4023	12.9198	27.2613
45	44.9097	49.3051	12.9198	27.2613
50	45.0897	47.8631	12.9198	27.2613

Figure 6.18: Overhead Ratio vs Buffer Size Graph Value

From Figure 6.18 we see that the value of Epidemic starts from 86.9744 at Buffer Size 5MB and it gradually decreases. At last, it becomes 45.0897 at Buffer Size 50MB. For PRoPHET, we see that the value of Overhead Ratio starts from 72.8137 at Buffer Size 5MB and it also decreases gradually. Finally, it becomes 47.8631 at Buffer Size 50MB. For Spray and Wait, we found that it starts from 19.8797 and ends in 12.9198 at our specified settings. For MaxProp, it starts from 41.3548 and finishes at 27.2613 for Buffer Size 50MB. In Addition, we can see that the value of Overhead Ratio is the lowest for Spray and Wait Routing Protocol for our specified settings.

6.4 Impact of Varying Message Generation Rate

Here, we observed the algorithms by increasing the message generation rate from 1 to 10 per minute. We have tried on a scenario containing 126 nodes and the TTL was 300 minutes.

6.4.1 Impact on Average Latency

The average latency of Epidemic, PROPHET and Spray and Wait decreased with the increase of message generation rate per minute. When we simulated the algorithms and tuned the message per minute to 10 messages per minute, average latency dropped 32% for Epidemic, 44% for PROPHET and 28% for Spray and Wait. Here, Spray and Wait's initial average latency was already lower than others, so it achieved the lowest average latency among them all.

On the other hand, the average latency of MaxProp was already higher than other algorithms. When the number of messages per minute was increasing, MaxProp's average latency was increasing too. When the number of messages was 3 and 5 per minutes, the average latency was the most. Then, after we tuned at 6 messages per minute, latency started to decrease. The graph of Average Latency vs Message Generation Rate is shown in Figure 6.19.

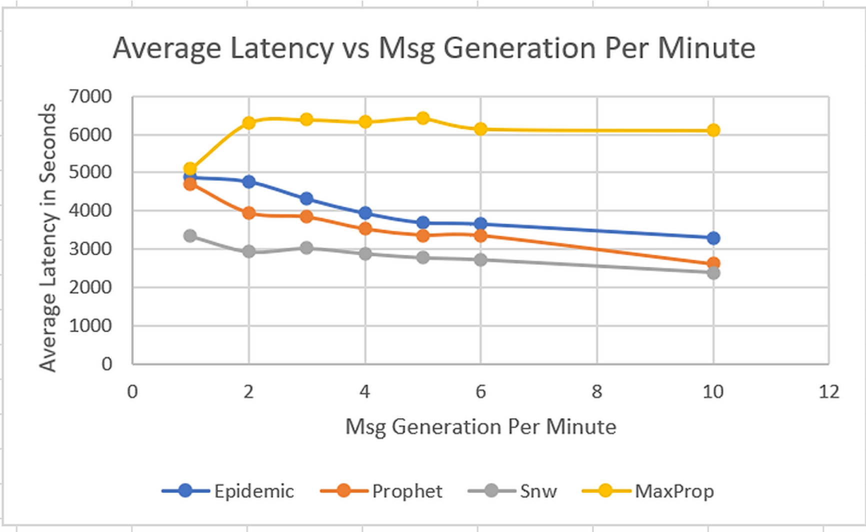


Figure 6.19: Average Latency vs Message Generation Rate

Summary :

From Figure 6.20 we see that the value of Epidemic starts from 4885.617 at Message Generation Rate 1 per minute and it gradually decreases. At last, it becomes 3304.268 at Message Generation Rate 10 per minute. For PRoPHET, we see that the value of Average Latency

Msg Gen	Average_Latency			
	Epidemic	Prophet	Snw	MaxProp
1	4885.617	4705.679	3351.023	5106.063
2	4775.294	3950.935	2937.356	6300.788
3	4321.659	3841.395	3025.923	6384.861
4	3942.778	3531.083	2885.856	6330.634
5	3694.102	3366.735	2776.192	6423.378
6	3657.886	3359.82	2725.977	6150.647
10	3304.268	2621.255	2387.529	6109.752

Figure 6.20: Average Latency vs Message Generation Rate Value

starts from 4705.679 at Message Generation Rate 1 per minute and it also decreases gradually. Finally, it becomes 2621.255 at Message Generation Rate 10 per minute. For Spray and Wait, we found that it starts from 3351.023 and ends in 2387.529 at our specified settings. For MaxProp, it starts from 5106.063 and finishes at 6109.752 for Message Generation Rate 10 per minute. In Addition, we can see that the value of Average Latency is the lowest for Spray and Wait Routing Protocol for our specified settings.

6.4.2 Impact on Delivery Probability

As we increase the message generation rate, the delivery probability of all the 4 algorithms started dropping. It dropped the most for MaxProp, then Spray and Wait. At the end of the simulation. At the end of the simulation, when the message generation rate per minute was 10, all of their delivery probability was nearer from one another. But Spray and Wait and MaxProp still had the highest message delivery probability than PROPHET and Epidemic algorithms. The graph of Delivery Probability vs Message Generation Rate is shown in Figure 6.21.

Summary :

From Figure 6.22 we see that the value of Epidemic starts from 0.3214 at Message Generation Rate 1 per minute and it gradually decreases. At last, it becomes 0.1119 at Message Generation Rate 10 per minute. For PRoPHET, we see that the value of Delivery Probability starts from 0.3393 at Message Generation Rate 1 per minute and it also decreases gradually. Finally, it becomes 0.1346 at Message Generation Rate 10 per minute. For Spray and Wait, we found that it starts from 0.6566 and ends in 0.1275 at our specified settings. For MaxProp, it starts from 0.7986 and finishes at 0.1342 for Message Generation Rate 10 per minute. In Addition, we can see that the value of Delivery Probability is the highest for PRoPHET Routing Protocol for our specified settings.

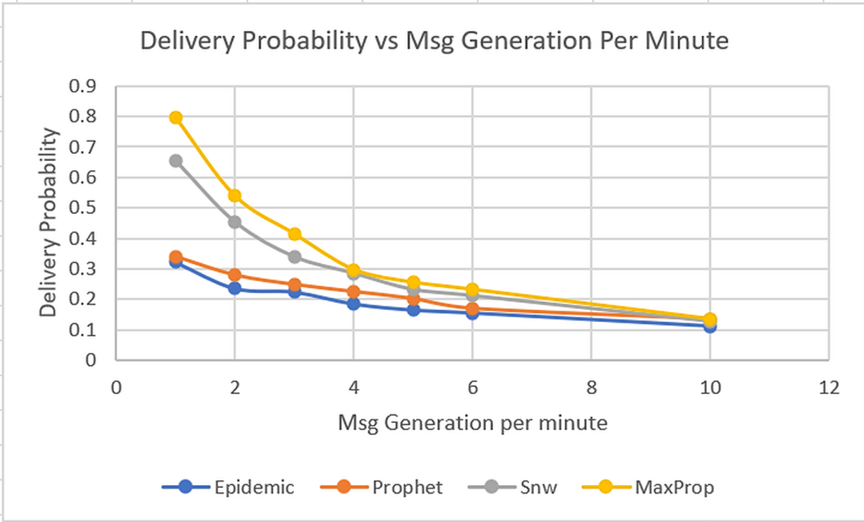


Figure 6.21: Delivery Probability vs Message Generation Rate

Msg Gen	Delivery-Probability			
	Epidemic	Prophet	Snw	MaxProp
1	0.3214	0.3393	0.6566	0.7986
2	0.2348	0.2786	0.4545	0.54
3	0.223	0.2474	0.3397	0.4143
4	0.1839	0.2242	0.2855	0.297
5	0.164	0.2013	0.2316	0.2547
6	0.1546	0.1691	0.2117	0.2312
10	0.1119	0.1346	0.1275	0.1342

Figure 6.22: Delivery Probability vs Message Generation Rate Value

6.4.3 Impact on Overhead Ratio

The initial overhead ratio was the lowest for spray and wait and the most for PROPHET and Epidemic as their message forwarding mechanism can overwhelm the resources. As the message generation rate per minute was increased overhead ratio, Spray and Wait's overhead ratio increased too. At the end of the simulation, its overhead increased more than 70%. But as in its spray phase, it restricts the number of copies of the messages, its overall overheard ratio was still lower than others.

When message generation per minute was 3, Epidemic's overhead ratio dropped nearly 44% and PROPHET's overhead ratio dropped almost 50%. It kept dropping for both of them. When message generation ratio was 10, both Epidemic's and PROPHET's overhead ratio dropped almost 66%. For MaxProp, as the message generation rate per minute was increasing, first, its overhead ratio was decreasing. When message generation rate per minute was 3, its overhead ratio dropped nearly 32%. Then it suddenly it increased almost 57% then its previous overhead ratio when message generation rate was 4. Then overhead ratio decreased a bit for message generation rate 5 and 6. After that, at message generation rate 10 per minute, overhead ratio suddenly increased to 141% of its initial stage which is a huge spike. The graph of Overhead Ratio vs Message Generation Rate is shown in Figure 6.23.

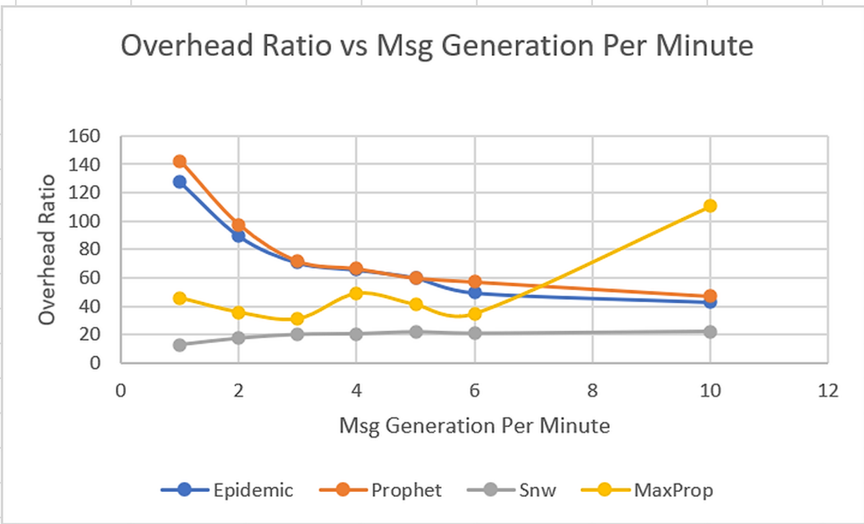


Figure 6.23: Overhead Ratio vs Message Generation Rate

Summary :

Msg Gen	Overhead_ratio			
	Epidemic	Prophet	Snw	MaxProp
1	127.6652	142.439	12.9748	45.962
2	89.3965	97.5061	17.4729	35.7136
3	70.6227	71.702	20.0706	31.1867
4	65.6385	66.4286	20.4976	49.1284
5	60.0458	59.8547	21.8355	41.156
6	49.4577	57.1732	20.8491	34.7735
10	42.8832	47.1067	22.1196	110.7956

Figure 6.24: Overhead Ratio vs Message Generation Rate Value

From Figure 6.24 we see that the value of Epidemic starts from 127.6652 at Message Generation Rate 1 per minute and it gradually decreases. At last, it becomes 42.8832 at Message Generation Rate 10 per minute. For PRoPHET, we see that the value of Overhead Ratio starts from 142.439 at Message Generation Rate 1 per minute and it also decreases gradually. Finally, it becomes 47.1067 at Message Generation Rate 10 per minute. For Spray and Wait, we found that it starts from 12.9748 and ends in 22.1196 at our specified settings. For MaxProp, it starts from 45.962 and finishes at 110.7956 for Message Generation Rate 10 per minute. In Addition, we can see that the value of Overhead Ratio is the lowest for Spray and Wait Routing Protocol for our specified settings.

6.5 Impact of Varying TTL

TTL(Time-To-Live) is the amount of time or lifespan on an active message. In every network, there are nodes and nodes have buffers where messages are stored. The messages can be of that particular node or can be transmitted by another node. There, each message has a time set up, after which that message is no longer usable by the application and will be dropped eventually. This time limit is referred as TTL.

Here, we observed the algorithms by varying the TTL from 60 to 420 minutes. We have tried on a scenario containing 126 nodes and messages are generated in every 25 to 35 seconds.

6.5.1 Impact on Average Latency

As the TTL was increased all 4 algorithms showed increment in average latency. At first all of their initial average latency were nearly same. As the TTL was increased, the average latency of MaxProp increased at a high rate. As the TTL became 420 minutes, MaxProp's average latency increased nearly 3.4 times.

On the other hand, Spray and Wait showed the least increment in average latency. It increased less than 1.7 times where Epidemic increased 2.26 times and PROPHET increased almost 2 times.

Spray and Wait performed better and MaxProp performed worse than other three algorithms. The graph of Average Latency vs TTL is shown in Figure 6.25.

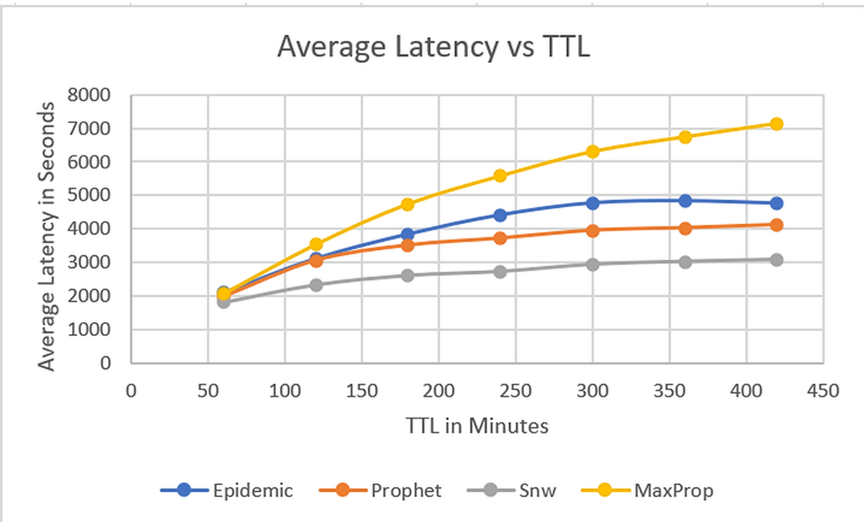


Figure 6.25: Average Latency vs TTL

Summary :

From Figure 6.26 we see that the value of Epidemic starts from 2107.285 at TTL 60 minutes

TTL	Average_Latency			
	Epidemic	Prophet	Snw	MaxProp
60	2107.285	1985.832	1824.137	2066.366
120	3123.804	3060.039	2335.626	3536.363
180	3848.819	3511.07	2613.031	4740.865
240	4417.019	3718.277	2728.998	5574.339
300	4775.294	3950.935	2937.356	6300.788
360	4835.85	4026.427	3023.131	6737.185
420	4763.624	4118.734	3079.71	7136.775

Figure 6.26: Average Latency vs TTL Value

and it gradually increases. At last, it becomes 4763.624 at TTL 420 minutes. For PRoPHET, we see that the value of Average Latency starts from 1985.832 at TTL 60 minutes and it also increases gradually. Finally, it becomes 4118.734 at TTL 420 minutes . For Spray and Wait, we found that it starts from 1824.137 and ends in 3079.71 at our specified settings. For MaxProp, it starts from 2066.366 and finishes at 7136.775 for TTL 420 minutes. In Addition, we can see that the value of Average Latency is the lowest for Spray and Wait Routing Protocol for our specified settings.

6.5.2 Impact on Delivery Probability

As we were increasing the TTL, at first the delivery probability of all 4 algorithms were increasing. But after the TTL hit 120 minutes, delivery probability of Epidemic and PROPHET started decreasing. By the end of the simulation at 420 minutes, Epidemic's delivery probability dropped 12.8% and PROPHET's delivery probability dropped to 15.3%. On the other hand, the delivery probability of MaxProp and Spray and wait were on the increase throughout the simulation. By the end of the simulation, MaxProp's delivery probability increased nearly 59%. For Spray and Wait, 25.8% increased delivery probability was achieved. MaxProp showed promising result in this simulation. The graph of vs Delivery Probability TTL is shown in Figure 6.27.

Summary :

From Figure 6.28 we see that the value of Epidemic starts from 0.2621 at TTL 60 minutes and it gradually decreases. At last, it becomes 0.2286 at TTL 420 minutes. For PRoPHET, we see that the value of Delivery Probability starts from 0.3183 at TTL 60 minutes and it also decreases gradually. Finally, it becomes 0.2695 at TTL 420 minutes . For Spray and Wait, we found that it starts from 0.3682 and ends in 0.4634 at our specified settings. For MaxProp, it starts from 0.3477 and finishes at 0.553 for TTL 420 minutes. In Addition, we can see that the value of Delivery Probability is the highest for MaxProp Routing Protocol

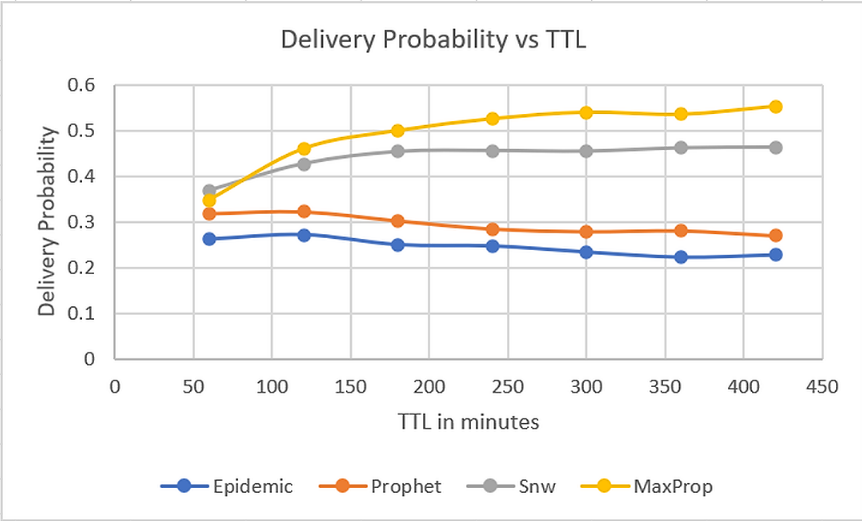


Figure 6.27: Delivery Probability vs TTL

TTL	Delivery-Probability			
	Epidemic	Prophet	Snw	MaxProp
60	0.2621	0.3183	0.3682	0.3477
120	0.2717	0.3224	0.4271	0.46
180	0.2505	0.3025	0.4538	0.4997
240	0.2478	0.2846	0.4559	0.5257
300	0.2348	0.2786	0.4545	0.54
360	0.2238	0.2806	0.462	0.5352
420	0.2286	0.2695	0.4634	0.553

Figure 6.28: Delivery Probability vs TTL Value

for our specified settings.

6.5.3 Impact on Overhead Ratio

As the TTL were increased, overhead ratio of Epidemic and PROPHET algorithms increased. The TTL of messages were increasing and there is not any active buffer managing system in these algorithms. So, their overhead ratio was increasing. It increases almost 1.5 times for Epidemic and 2 times for PROPHET algorithm.

But Spray and Wait limits the replication of messages and MaxProp drops messages by their priority, which indicates it has a buffer management system. So, their overhead ratios were decreased. MaxProp's overhead ratio decreased 1.5 times where for Spray and Wait. It decreased almost 1.2 times. But as the overhead ratio was already low for Spray and Wait, it seems to perform better in this simulation. The graph of Overhead Ratio vs TTL is shown in Figure 6.29.

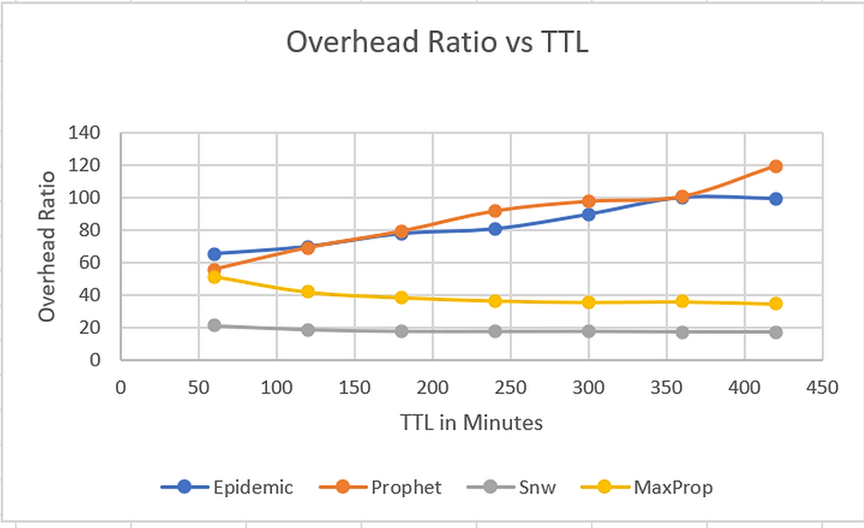


Figure 6.29: Overhead Ratio vs TTL

Summary :

TTL	Overhead_ratio			
	Epidemic	Prophet	Snw	MaxProp
60	65.1828	55.5247	21.0948	51.2126
120	69.5718	68.9321	18.5112	41.8244
180	77.5765	79.0294	17.4525	38.5247
240	80.4282	91.5584	17.4009	36.5859
300	89.3965	97.5061	17.4729	35.7136
360	99.8563	100.4585	17.1733	36.0499
420	99.1347	119.4555	17.1226	34.7339

Figure 6.30: Overhead Ratio vs TTL Value

From Figure 6.30 we see that the value of Epidemic starts from 65.1828 at TTL 60 minutes and it gradually increases. At last, it becomes 99.1347 at TTL 420 minutes. For PRoPHET, we see that the value of Overhead Ratio starts from 55.5247 at TTL 60 minutes and it also increases gradually. Finally, it becomes 119.4555 at TTL 420 minutes . For Spray and Wait, we found that it starts from 21.0948 and ends in 17.1226 at our specified settings. For MaxProp, it starts from 51.2126 and finishes at 34.7339 for TTL 420 minutes. In Addition, we can see that the value of Overhead Ratio is the lowest for Spray and Wait Routing Protocol for our specified settings.

6.6 Impact of Varying Number of Nodes

In DTN, nodes are the devices that move during network operations. They play some important roles in DTN. In flooding-based algorithms Epidemic, Spray and wait, MaxProp etc.

increased nodes can be means of more transport option for messages. Here, we have increased the number of nodes from 120 to 360, which can be a possible scenario of a sparse to dense network and varied the simulation time from 5000 seconds to 20000 seconds for all the algorithms to observe their performance in different situations.

6.6.1 Impact on Average Latency

As the number of nodes were increased slowly, all 4 routings algorithms showed reduction in average latency. MaxProp algorithm showed the most reduction in Average latency then others. Its latency reduced to nearly half of the initial average latency. Epidemic also showed declined average latency. Though the other two algorithm showed reduced average latency, it was not that much significant.

The Spray and Wait algorithm showed the least reduction in average latency, it was almost consistent throughout the process. But its initial average latency was already low. So, even though it didn't show the most reduction, it performed better than other algorithms.

On the other hand, MaxProp seemed to be a good option to choose for a large sized network, as its average latency got significantly lower with the increment in number of nodes. The graph of Average Latency vs Number Of Nodes is shown in Figure 6.31.

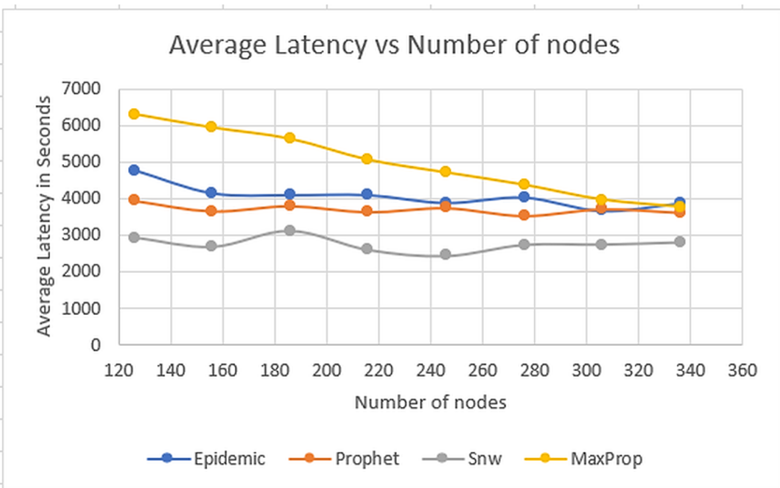


Figure 6.31: Average Latency vs Number of Nodes

Summary :

From Figure 6.32 we see that the value of Epidemic starts from 4775.294 at Number of Nodes 126 and it gradually decreases. At last, it becomes 3880.702 at Number of Nodes 336. For PRoPHET, we see that the value of Average Latency starts from 3950.935 at Number of Nodes 126 and it also decreases gradually. Finally, it becomes 3625.142 at Number of Nodes 336 . For Spray and Wait, we found that it starts from 2937.356 and ends in 2804.889

Number of nodes	Average_Latency			
	Epidemic	Prophet	Snw	MaxProp
126	4775.294	3950.935	2937.356	6300.788
156	4153.857	3664.354	2687.387	5933.967
186	4103.131	3805.348	3106.527	5627.213
216	4103.977	3640.42	2608.721	5056.142
246	3892.344	3751.143	2445.815	4709.161
276	4038.107	3538.458	2738.962	4375.236
306	3676.872	3720.311	2747.119	3971.278
336	3880.702	3625.142	2804.889	3776.556

Figure 6.32: Average Latency vs Number of Nodes Value

at our specified settings. For MaxProp, it starts from 6300.788 and finishes at 3776.556 for Number of Nodes 336. In Addition, we can see that the value of Average Latency is the lowest for Spray and Wait Routing Protocol for our specified settings.

6.6.2 Impact on Delivery Probability

As the number of nodes were increased, MaxProp and Spray and Wait algorithm showed improved delivery probability where Epidemic and PROPHET showed deteriorated performance.

At initial stages, MaxProp and Spray and Wait had higher delivery Probability then Epidemic and PROPHET because of their algorithmic properties.

For PROPHET and Epidemic, their delivery probability got decreased to 10-12% with the increased number of nodes. On the other hand, MaxProp showed 55.77% increment and Spray and Wait showed almost 59% increment in deliver probability when the number of nodes were increased to 360. This is mainly because more nodes ensure increased number of opportunistic contacts between nodes.The graph of Delivery Probability vs Number Of Nodes is shown in Figure 6.33.

Summary :

From Figure 6.34 we see that the value of Epidemic starts from 0.2348 at Number of Nodes 126 and it gradually decreases. At last, it becomes 0.2074 at Number of Nodes 336. For PRoPHET, we see that the value of Delivery Probability starts from 0.2786 at Number of Nodes 126 and it also decreases gradually. Finally, it becomes 0.2491 at Number of Nodes 336 . For Spray and Wait, we found that it starts from 0.4545 and ends in 0.7235 at our specified settings. For MaxProp, it starts from 0.54 and finishes at 0.8412 for Number of Nodes 336. In Addition, we can see that the value of Delivery Probability is the highest for MaxProp Routing Protocol for our specified settings.

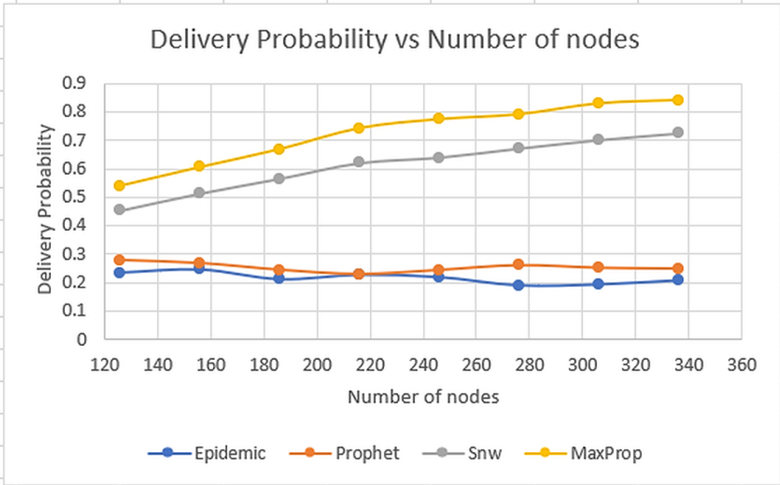


Figure 6.33: Delivery Probability vs Number of Nodes

Number of nodes	Delivery-Probability			
	Epidemic	Prophet	Snw	MaxProp
126	0.2348	0.2786	0.4545	0.54
156	0.2457	0.2683	0.514	0.6057
186	0.2115	0.245	0.5656	0.6687
216	0.2272	0.2302	0.6201	0.7426
246	0.219	0.2444	0.6393	0.7741
276	0.1896	0.2608	0.6708	0.7912
306	0.193	0.2519	0.7002	0.8296
336	0.2074	0.2491	0.7235	0.8412

Figure 6.34: Delivery Probability vs Number of Nodes Value

6.6.3 Impact on Overhead Ratio

The overhead ratio here shows how the cost of the network will be as the network became denser. The Epidemic and PROPHET routing algorithm showed huge increment when the network become denser. Especially after 240 nodes, the overheard ratio got larger. Finally, when simulation stopped after adding 360 nodes to the network, the overhead ratio of Epidemic became nearly 8.3 times larger than its initial stage. For PROPHET, it became 6.7 times larger than its initial stage.

Spray and Wait limits the number of message replication and MaxProp drops messages when the buffer is full. So, their initial overhead ratios were already low than the other two algorithms. When the nodes were increased, Spray and Wait showed great positive performance. Its overhead ratio went down 1.4 times, where for other algorithms, overhead ratio was quite high. For MaxProp, overhead ratio increased almost 4 times, still it was much lesser than Epidemic and PROPHET results. The graph of Overhead Ratio vs Number Of Nodes is shown in Figure 6.35.

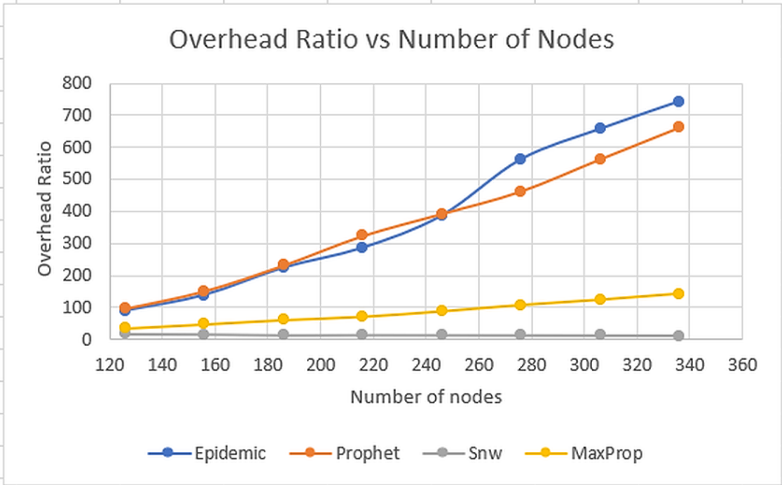


Figure 6.35: Overhead Ratio vs Number of Nodes

Summary :

Number of nodes	Overhead_ratio			
	Epidemic	Prophet	Snw	MaxProp
126	89.3965	97.5061	17.4729	35.7136
156	138.3175	151.8342	16.4328	48.3571
186	223.7638	231.824	13.2609	61.7103
216	285.6867	323.6858	14.0817	72.1088
246	386.175	392.507	13.7002	88.8435
276	561.9603	462.4934	13.1347	108.5311
306	657.4291	562.2147	12.6354	124.7764
336	743.5974	660.8489	12.1958	143.4557

Figure 6.36: Overhead Ratio vs Number of Nodes Value

From Figure 6.36 we see that the value of Epidemic starts from 89.3965 at Number of Nodes 126 and it gradually increases. At last, it becomes 743.5974 at Number of Nodes 336. For PRoPHET, we see that the value of Overhead Ratio starts from 97.5061 at Number of Nodes 126 and it also increases gradually. Finally, it becomes 660.8489 at Number of Nodes 336 . For Spray and Wait, we found that it starts from 17.4729 and ends in 12.1958 at our specified settings. For MaxProp, it starts from 35.7136 and finishes at 143.4557 for Number of Nodes 336. In Addition, we can see that the value of Overhead Ratio is the lowest for Spray and Wait Routing Protocol for our specified settings.

Chapter 7

Conclusion and Future Work

7.1 Limitation

In this chapter, we have evaluated the performance of DTN routing protocols, i.e., Epidemic, PRoPHET, Spray-and-Wait and MaxProp in intermittently connected mobile networks (ICMNs). We have a tendency to attempt our best to fulfil our task properly. But we all know that nothing can be done without blemish and difficulties. We've analyzed the performance based on a few parameters. So, it is determined that performance will be measured based on more parameters so that we can find out the best routing algorithm accurately. We will try our best to overcome the faults and make our system more reliable.

7.2 Future Work

We've proposed to analyze the performance of four routing protocols- Epidemic, PRoPHET, Spray-and-Wait and MaxProp. Among them, all four protocols have been analyzed based on six parameters. In future, we'll evaluate the processing time and memory space of all routing protocols based on those six parameters. We'll also try to find out the complexity of these routing algorithms.

7.3 Conclusion

The Delay Tolerant Networks are under the enormous research for its various routing algorithms and the areas of its applicability. Each algorithm has its own assets and liabilities. The algorithms which are used for routing protocols in DTN are not bounded only to the algorithms described in this work. For the selection of algorithms to be implemented, it is useful to examine them comparatively and select the best as per the application scenario. In this work, we have considered the performance of DTN routing protocols, i.e., Epidemic, PRoPHET, and Spray-and-Wait in intermittently connected mobile networks (ICMNs). Simulation results show the performance comparison of the DTN routing protocols in terms of message delivery probability, average latency and overhead ratio with the variation of transmit range, simulation time and buffer size respectively. From these results we may conclude that the best successor for routing messages in ICMNs is Spray-and-Wait. DTN is wireless networks where end to end connection is not possible and frequent link disconnection due to mobility of nodes, the low density of nodes or when the network extends over long distances.

References

[1] A. Socievole and S. Marano, “Evaluating the impact of energy consumption on routing performance in delay tolerant networks,” in *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 481–486, IEEE, 2012.

[2] A. Vahdat, D. Becker, *et al.*, “Epidemic routing for partially connected ad hoc networks,” 2000.

[3] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks,” *ACM SIGMOBILE mobile computing and communications review*, vol. 7, no. 3, pp. 19–20, 2003.

[4] 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*, pp. 252–259, 2005.

[5] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: The single-copy case,” *IEEE/ACM transactions on networking*, vol. 16, no. 1, pp. 77–90, 2008.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Efficient routing in intermittently connected mobile networks: The multiple-copy case,” *IEEE/ACM transactions on networking*, vol. 16, no. 1, pp. 77–90, 2008.

[7] A. G. Voyiatzis, “A survey of delay-and disruption-tolerant networking applications,” 2012.

[8] H. Ntareme, M. Zennaro, and B. Pehrson, “Delay tolerant network on smartphones: Applications for communication challenged areas,” in *Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition*, pp. 1–6, 2011.

[9] T. Hossmann, T. Spyropoulos, and F. Legendre, “Know thy neighbor: Towards optimal mapping of contacts to social graphs for dtn routing,” in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, IEEE, 2010.

[10] T. Spyropoulos, T. Turletti, and K. Obraczka, "Routing in delay-tolerant networks comprising heterogeneous node populations," *IEEE transactions on mobile computing*, vol. 8, no. 8, pp. 1132–1147, 2008.

[11] A. Krifa, C. Barakat, and T. Spyropoulos, "Message drop and scheduling in dtns: Theory and practice," *IEEE Transactions on Mobile Computing*, vol. 11, no. 9, pp. 1470–1483, 2011.

[12] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*, pp. 1–10, 2009.

[13] J. Burgess, B. Gallagher, D. D. Jensen, B. N. Levine, *et al.*, "Maxprop: Routing for vehicle-based disruption-tolerant networks.," in *Infocom*, vol. 6, Barcelona, Spain, 2006.

Generated using Undegraduate Thesis LaTeX Template, Version 1.4. Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh.

This thesis was generated on Thursday 1st July, 2021 at 4:03pm.