**Performance Analysis**

**of**

**DTN Routing Protocols**

**Submitted By-**

**Zakia Sultana Tanzil 16.02.04.004**

**Anika Rahman 16.02.04.012**

**Arunima Ayshee 16.02.04.066**

**Tahira Salwa Rabbi Nishat 16.02.04.070**

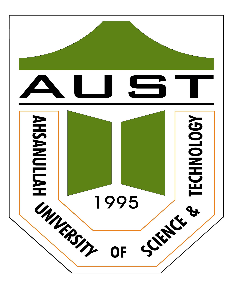
**Supervised By-**

**Md. Tanvir Ahmed**

**Assistant Professor**

**Dept. of CSE**

**Ahsanullah University of Science and Technology**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

DHAKA, BANGLADESH

DECEMBER 2020

**ABSTRACT**

Delay Tolerant Network (DTN) is distinguished because of its lack of connectivity and a lack of instant end-to-end paths. In these environments, popular ad hoc routing protocols such as AODV and DSR unable or fail to establish routes. It is because these protocols first try to establish a complete route, and when it is established, they forward the actual data. However, when instant end-to-end paths are unavailable or impossible to build, routing protocols must make a "store and forward" approach. In this approach, data is incrementally moved and stored in hopes throughout the network so that it can eventually reach its destination. A common technique that is used to maximize the probability of a message getting transferred successfully is to create many copies of the message in hope that at least one will succeed in reaching its destination. It is feasible only on networks with large amounts of local storage and internode bandwidth relative to the expected traffic. A more discriminate algorithm is needed in other network areas where usable storage and internode throughput opportunities are more tightly constrained. The use of DTN is unlimited in disaster recovery networks, military and tactical systems, wildlife tracking/monitoring sensor networks, vehicular communication, interplanetary networking, and communication in remote and rural areas and developing countries. Routing in DTN is a challenging problem because, at any given time case, the probability that there is an end-to-end connected path from a source to a destination is low. Since the routing algorithms for general networks assume that the links between nodes are stable and do not fail frequently, they do not usually work in DTN. Hence, the routing problem is still an active research area in DTN. In this paper, we will be studying and analyzing the performance of DTN.

### **TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **ABSTRACT** | | **i** | |
| **TABLE OF CONTENTS** | | **ii** | |
| **LIST OF FIGURES** | | **iii** | |
| **LIST OF TABLES** | | **iv** | |
| **Chapter 1: Introduction** | | | **1** | |
|  | 1.1 Motivation ………………………………………...……………..... | | 2 | |
|  | 1.2 Objective …………...………………………………………………. | | 4 | |

|  |  |  |
| --- | --- | --- |
| **Chapter 2: Literature Review** | | **5** |
|  | 2.1 Overview……………………………………………………………. | 5 |
|  | 2.2 Related Work ………....……………………………………………... | 5 |
|  |  |  |
| **Chapter 3: Background Study** | | **7** |
|  | 3.1 Description of Routing Protocols …..………………………………. | 7 |

|  |  |  |
| --- | --- | --- |
| **Chapter 4: Simulation Setup and Performance Metrics** | | **14** |
|  | 4.1 The ONE Simulator ………………………………………………. | 14 |
|  | 4.2 Simulation Environment Setup ………………...……………………. | 16 |
|  | 4.3 Performance Metrics ………………………………………………... | 18 |
|  | 4.4 Methodology ………………………..………………………………. | 18 |

|  |  |  |
| --- | --- | --- |
| **Chapter 5: Performance Analysis** | | **20** |
|  | 5.1 Impact of Varying Transmit Range …………………………………. | 20 |
|  | 5.2 Impact of Varying Simulation Time ……...…….…………………… | 23 |
|  | 5.3 Impact of Varying Buffer Size...………………..……………………. | 26 |

|  |  |  |
| --- | --- | --- |
| **Chapter 6: Future Work and Conclusion** | | **30** |
|  | 6.1 Limitation…………………………………………………………… | 30 |
|  | 6.2 Future Work…………….…………………………………………… | 31 |
|  | 6.3 Conclusion…….…………………………………………………….. | 31 |

|  |  |
| --- | --- |
| **REFERENCES** | **32** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| 1(a)  1(b)  1(c)  1(d)  1(e)  3(a)  3(b)  3(c)  3(d)  3(e)  3(f)  4(a)  4(b)  4(c)  4(d)  4(e)  5(a)  5(b)  5(c)  5(d)  5(e)  5(f)  5(g)  5(h)  5(i) | DTN Architecture …………….…………………………………………….  Store and Carry Forward Approach ....………………………………………  Terrorist Attack ……………………………………………………………..  Tsunami …………………….………………………………………………  Normal and During Disaster Communication System...……………………..  Single-Copy Case ……………………………………………………………  Epidemic ……………………………………………………………………  PRoPHET .…………………………………..……………………………...  Summarized View of Spray and Wait Protocol ….…………………………..  The MaxProp Routing Strategy ………....…………………………………...  Algorithm of MaxProp ……………………………………………………...  Overview of ONE Simulator ………………………………………………..  Screenshot of ONE Simulator ………………………………………………  Methodology …………………....…………………………………………...  Specification Settings ………………………………………………………..  Java Class ………………………………………………………...………….  Average Latency vs Transmit Range ..………………………………………..  Delivery Probability vs Transmit Range ……………………….…………......  Overhead Ratio vs Transmit Range ………………………...………………..  Average Latency vs Simulation Time ………………………………………...  Delivery Probability vs Simulation Time …………………………………….  Overhead Ratio vs Simulation Time ……………….…………...………….....  Average Latency vs Buffer Size ……………………………………………....  Delivery Probability vs Simulation Time ……………………………………..  Overhead Ratio vs Simulation Time …………………………………...…….. | 1  2  3  3  3  7  8  10  11  12  12  15  15  19  19  19  21  22  23  24  25  26  27  28  29 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Table 1  Table 2  Table 3  Table 4  Table 5 | Simulation Parameters and Values for Varying Transmission Range ..……….  Simulation Parameters and Values for Varying Simulation Time …. ..……….  Simulation Parameters and Values for Varying Buffer Size ………. ..……….  Parameters of Routing Algorithms ………………………………. ..……….  [Add Later] ………………………………….……………………………... | 16  16  17  17  x |

**Chapter 1**

**Introduction**

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. The application areas of DTN are challenging circumstances like military battlefields, deep underwater communication, natural disasters affected areas or remote area social networking 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.

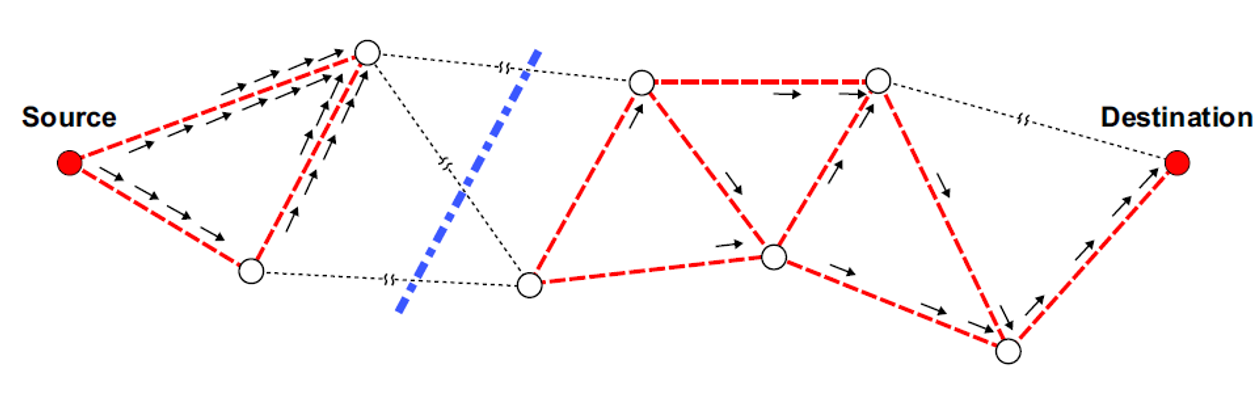


Figure 1(a): DTN Architecture

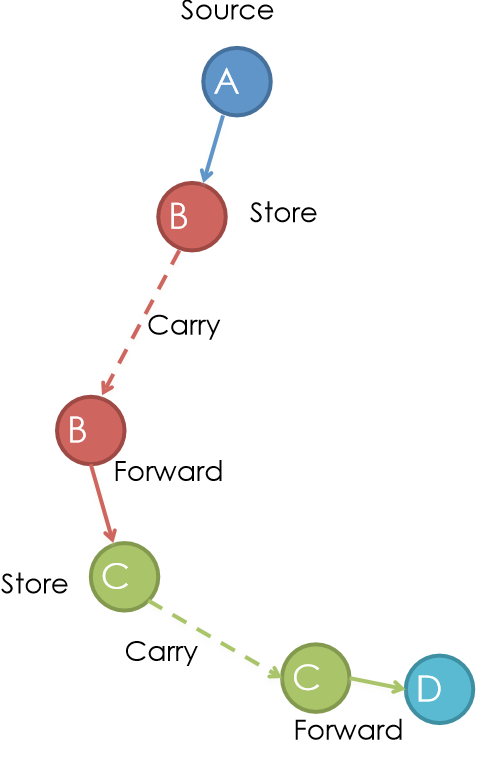


Figure 1(b): Store and Carry Forward Approach

DTN uses the store-and-forwarding approach. Messages may be sent to unavailable end hosts. Hop-to-Hop re-transmission happens in case of errors. In Store and Forward Approach, complete messages or a chunk of it is transferred and stored in nodes until it reaches the destination. Every node is associated with a persistent storage device (like hard disk) where it can store the messages. Here, a communication link may not be available for a long time. One node may send or receive data much faster or more certainly than the other nodes. A message, once transmitted, may need to be re-transmitted if error takes place.

* 1. **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. 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(c): Terrorist Attack

Figure 1(d): Tsunami

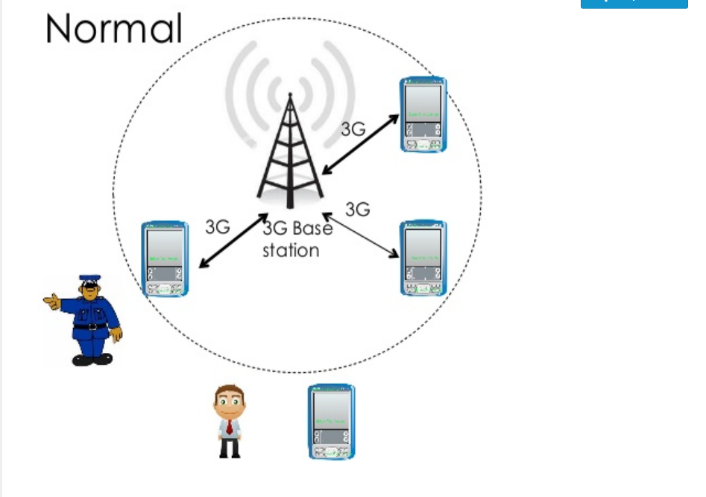
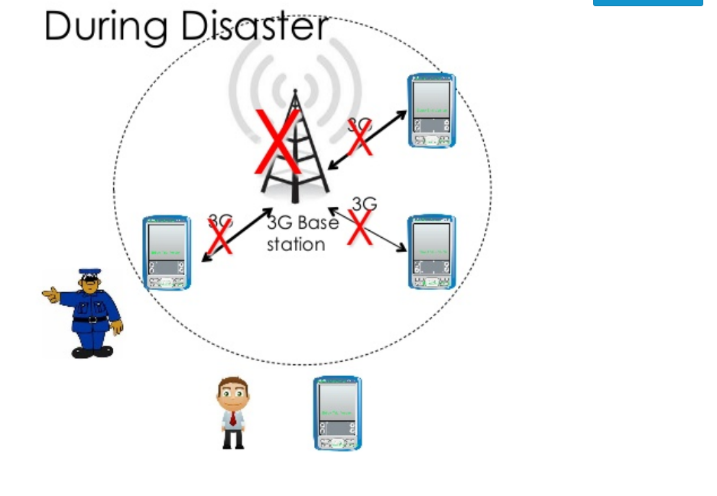
 

Figure 1(e): Normal and During Disaster Communication System

* 1. **Objective**

Delay-tolerant networking (DTN) is a point of view to computer network architecture that seeks to address the technical issues 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 paper, we present an evaluation of DTN routing protocols as related to intermittently connected mobile networks, focused on Epidemic [9], PROPHET [11], and Spray-and-Wait [12]. Epidemic routing is a clear example of the “store-carry-and-forward” approach, which forwards each copy of the messages (not common among the nodes in contact) 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 known destinations in the DTN. Then it replicates messages during opportunistic encounters only if the mobile node in contact that does not retain the message appears 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 [11] 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 [12] they recommended two efficient multi-copy schemes, called Spray and Wait and Spray and Focus that manage to overcome the shortcomings of flooding-based and other existing schemes.

In [13] Delay and Disruption-Tolerant Networking (DTN) is a new communication model that can span across multiple networks and cope with harsh conditions not envisioned in the Internet model. Disregard the wide variation of operating conditions. The DTN implementations gain in providing sufficient performance for most situation.

In [14] this paper discusses the Delay Tolerant Network (DTN) service and protocol stack and prompts an implementation of it on the Android platform that is called "Bytewalla". It allows the use of phones for the physical transportation of data between network nodes in areas where there are no other links is available for sending the data.

In [15] this paper, they argue that it is not so much the choice or complication of social metrics and algorithms that bears the most weight on performance, but rather the mapping from the mobility process generating 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 in all situation considered.

In [16] opportunistic routing techniques have been proposed, whereas node may store-and-carry a message for some time, until anew forwarding opportunity arises. These algorithms most focus on relatively homogeneous settings of nodes. However, in many future applications, participating nodes may include handhelds, vehicles, sensors, etc.

In [17] this paper, they proposed a practical and efficient joint scheduling and drop policy that can optimize different performance metrics, such as average delay and delivery probability. At first, they use the theory of combat-based message broadcast to derive the optimal policy based on global knowledge about the network. Then, disclose a method that estimates all necessary parameters which using locally collected statistics information.

**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 Description of the Routing Protocols**

In this section, 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.

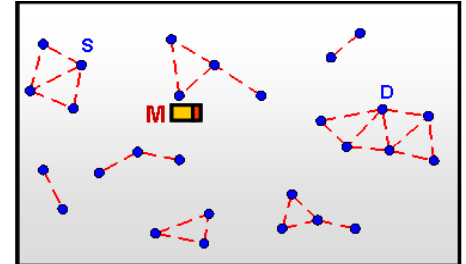


Figure 3(a): Single-Copy Case

**Multi-Copy Schemas:**

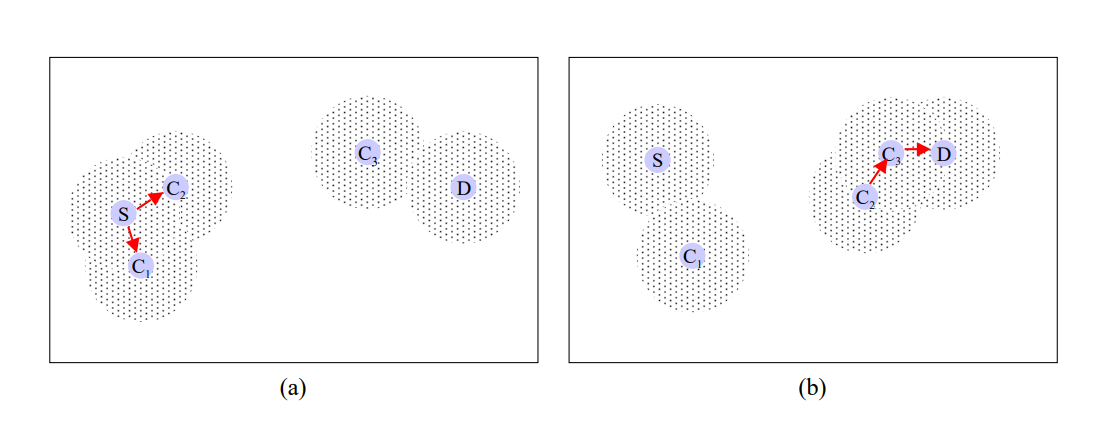
Multi-copy scheme onwards several copies of the same message to the network, i.e., replicates messages which are called replication-based.

**Epidemic:**

The epidemic routing protocol is used due to 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:

Figure 3(b): Epidemic



**Figure 1 (a):**  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 1 (b):**  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.

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.

**PRoPHET:**

The basic functioning of PRoPHET routing is similar to Epidemic and attempts to improve the delivery probability of messages. 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. Delivery predictability P(a,b) is stored in the internal delivery vector and gets updated whenever nodes encounter each other. The delivery predictability used by each node are recalculated at each opportunistic encounter according to the following three rules:

1. 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) × Pinit (1)**

Where **Pinit** is an initialization constant.

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

**P(a,b) = P(a,b)old × Ɣk (2)**

Where **Ɣk** is an aging constant.

1. The delivery predictability also follows the transitive property, that is, if a node A often encounters node B and node B frequently encounters node D, then node D probably 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) × β (3)**

Where **β** is a scaling constant which decides how large impact the transitivity should have on the delivery predictability.

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.7 In PROPHET if a node has visited a location various time then there is a possibility that this pattern will repeated in the future. In PROPHET each node uses probabilistic metric called delivery predictability to transfer messages to a reliable 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.

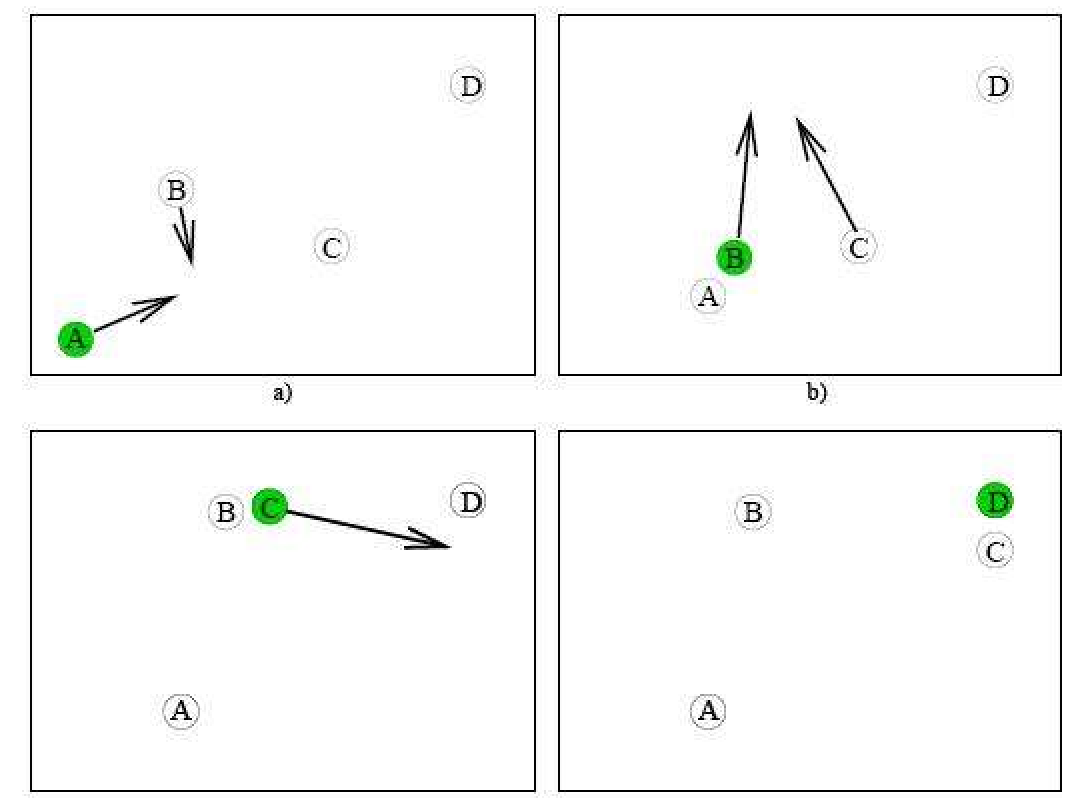


Figure 3(c): PRoPHET

**Spray and Wait:**

Spray and wait is a protocol that combines the speed of epidemic routing with direct transmission simplicity. It restricts the number of copies spread over 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.

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.

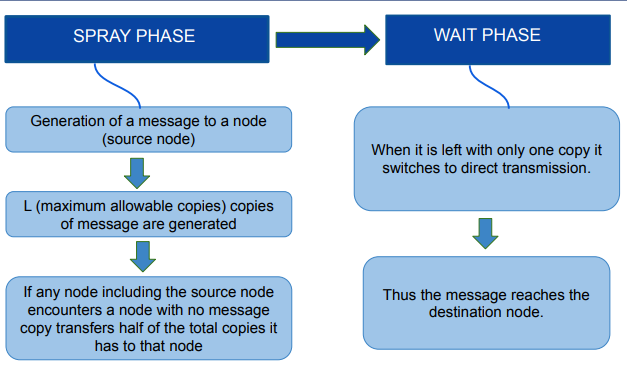


Figure 3(d): Summarized view of Spray and Wait protocol

**MaxProp:**

MaxProp is a flooding-based routing protocol which means, each node spreads a number of copies of each message though they may have no information about the network. 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.

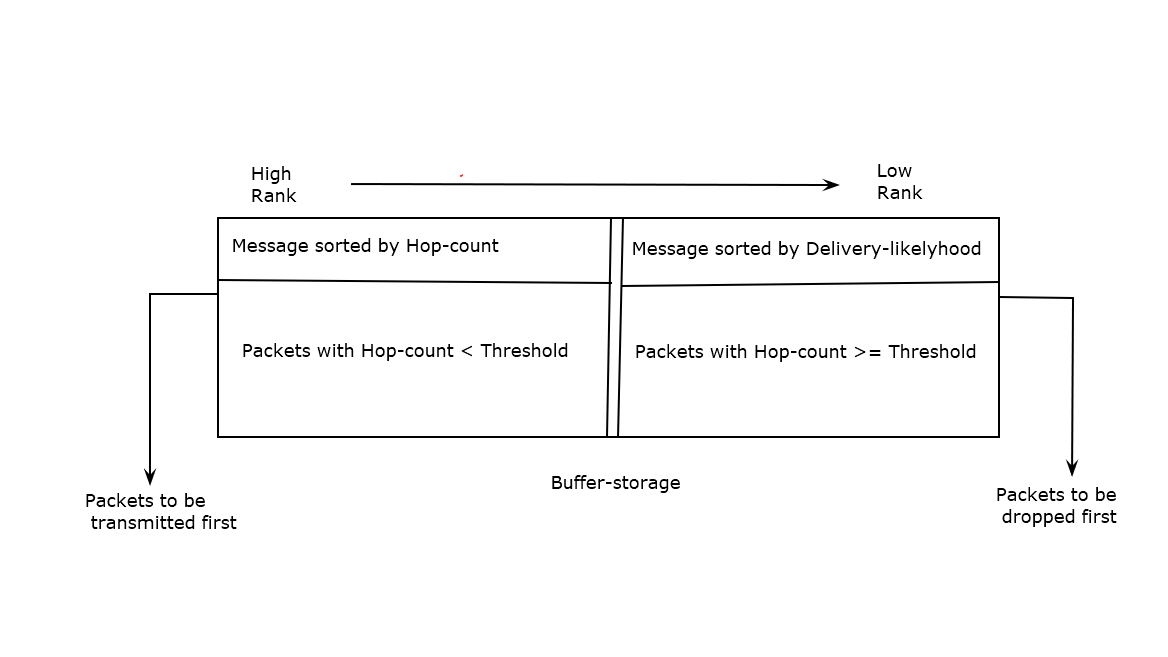


Figure 3(e): The MaxProp Routing Strategy

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.

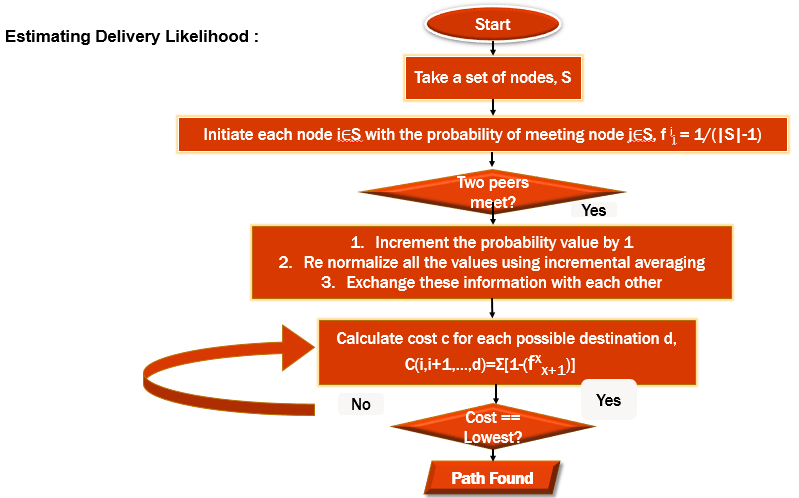


Figure 3(f): Algorithm of MaxProp

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.

**Chapter 4**

**Simulation Setup and Performance Metrics**

In this paper, we focus on the performance analysis of Epidemic, PRoPHET, and Spray-and-Wait routing protocols in an intermittently connected mobile network. All routing protocols are simulated using Opportunistic Network Environment (ONE) [14] with version of 1.5.1. This section describes the environment modeling parameters, performance metrics, and the analysis of results of the simulations.

**4.1 The ONE Simulator**

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 [16] and the ONE simulator project page [32] where the source code is also available.

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.

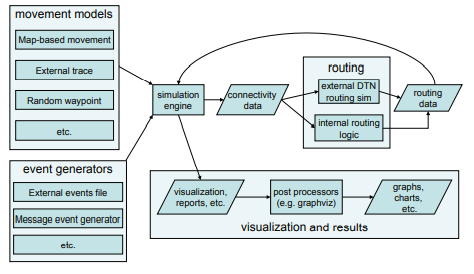


Figure 4(a): Overview of ONE Simulator

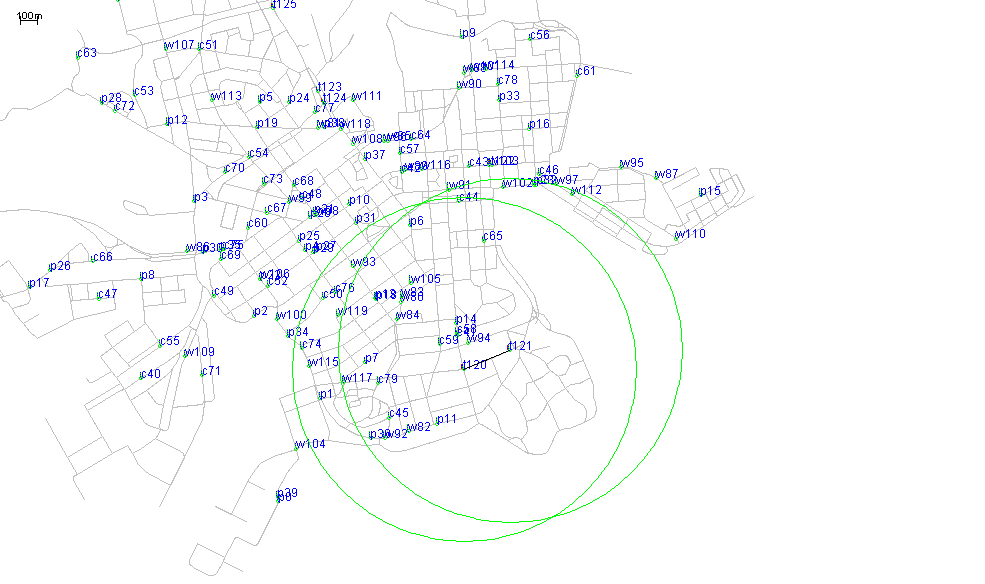


Figure 4(b): Screenshot of ONE Simulator

**4.2 Simulation Environment Setup**

Parameters of simulation setup and routing algorithms are defined in Table 1, Table 2, Table 3 and Table 4 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. Table 4 summarizes the simulation configuration for routing algorithms.

**Simulation Setup for varying Transmit Range :**

Table 1: Simulation Parameters and Values for Varying Transmit Range

|  |  |
| --- | --- |
| **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 and Spray-and-wait |
| 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 |

**Simulation Setup for varying Simulation Time:**

|  |  |
| --- | --- |
| **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 and Spray-and-wait |
| 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 2: Simulation Parameters and Values for Varying Simulation Time

**Simulation Setup for varying Buffer Size:**

|  |  |
| --- | --- |
| **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 and Spray-and-wait |
| Buffer Size | 5MB, 10MB, 15MB, 20MB, 25MB, 30MB,  35MB, 40MB, 45MB, 50MB |
| 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 3: Simulation Parameters and Values for varying Buffer Size

**Simulation Setup for varying Msg Generation Rate:**

|  |  |
| --- | --- |
| **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 and Spray-and-wait |
| Buffer Size | 5MB |
| Message Generation Rate (message/min.) | 1,2,3,4,5,6,10 (In Minutes) |
| 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 |

**Simulation Setup for varying TTL:**

|  |  |
| --- | --- |
| **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 and Spray-and-wait |
| 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) |
| Mobility | Random Way Point |
| Movement model | Shortest Path Map Based |
| Message Size | 500 KB – 1 MB |
| Simulation Area Size | 4500m \* 3400m |

|  |  |  |
| --- | --- | --- |
| **Routing Algorithm** | **Parameters** | **Values** |
| Epidemic | - | - |
| PRoPHET | Seconds in Time Unit | 30s |
| Spray-and-Wait | No. of Copies (L) | 6 |

Table 4: Parameters of Routing Algorithms

**4.3 Performance Metrics**

The three metrics to measure the performance of the different protocols:

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

Delivery ratio = … … … (1)

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

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

Overhead Ratio= … … … (2)

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

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

Average Latency= … … … (3)

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

**4.4 Methodology**

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

ONE Simulator

(Protocol Environment)

Report of Performance

DTN Protocol in the form of JAVA class

Figure 4(c): Methodology

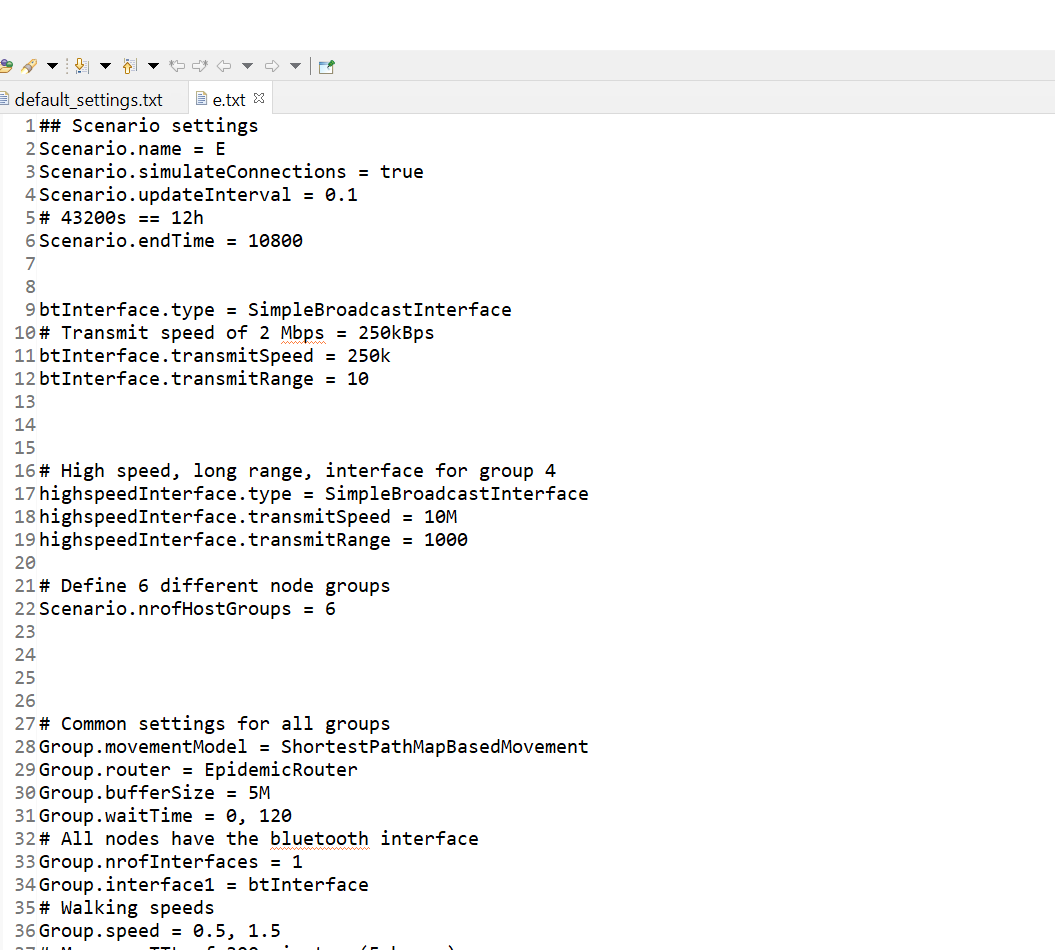
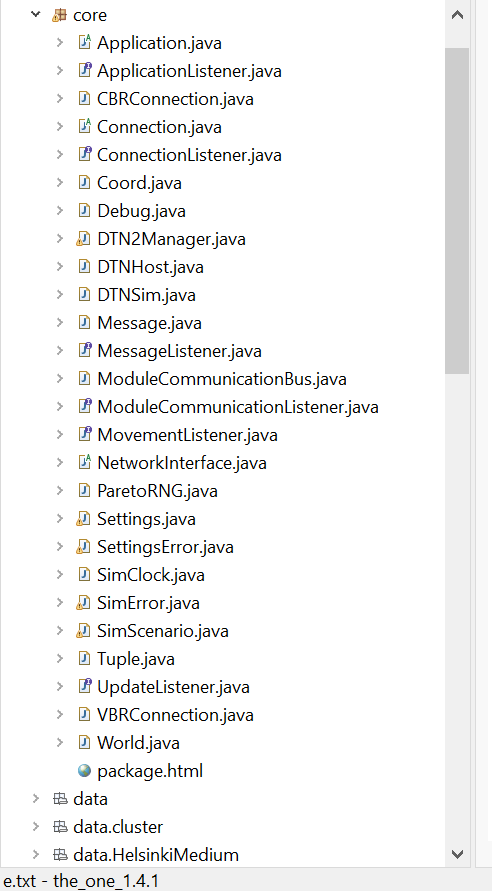
 

Figure 4(d): Specification Settings

Figure 4(e): Java Class

**Chapter 5**

**Performance Analysis**

**5.1 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.

**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.

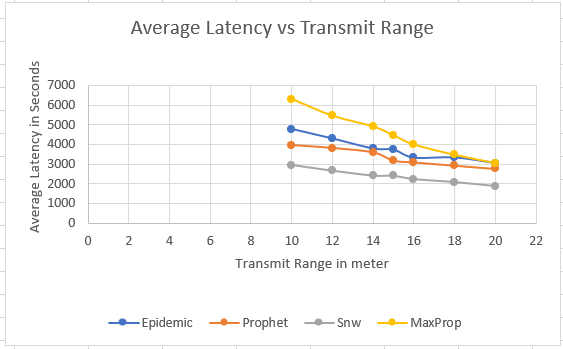


Figure 5(a): Average Latency vs Transmit Range

**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.

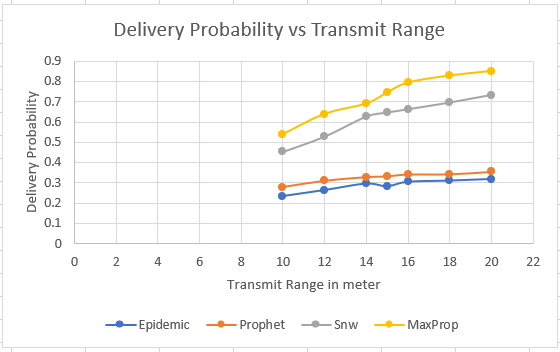


Figure 5(b): Delivery Probability vs Transmit Range

**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.

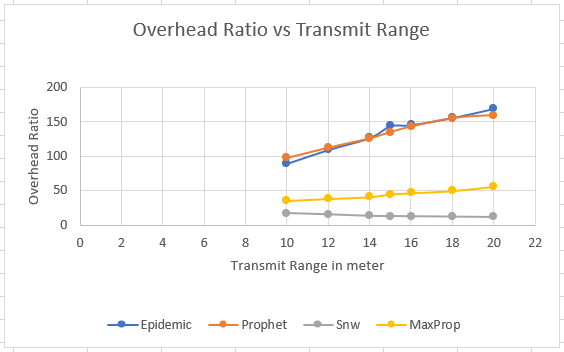


Figure 5(c): Overhead Ratio vs Transmit Range

**5.2 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.

**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.

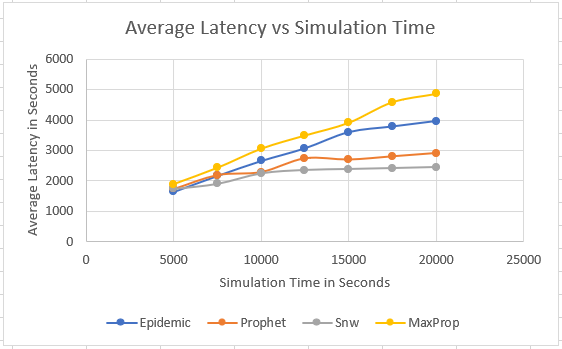


Figure 5(d): Average Latency vs Simulation Time

**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.

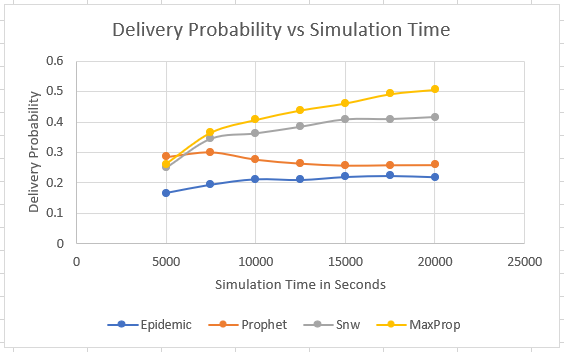


Figure 5(e): Delivery Probability vs Simulation Time

**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 almost 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 due to 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.

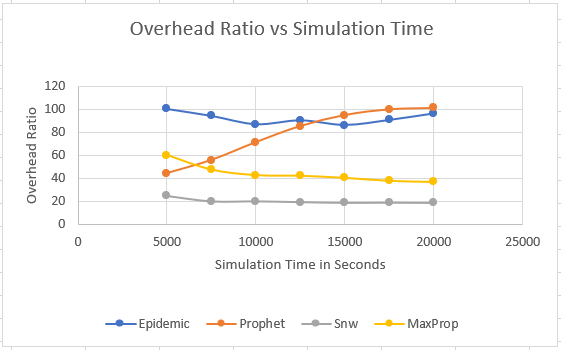


Figure 5(f): Overhead Ratio vs Simulation Time

**5.3 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.

**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.

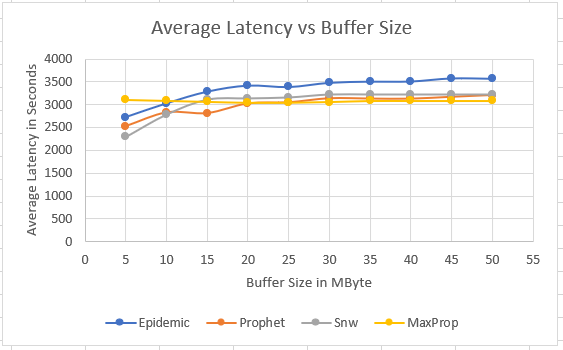


Figure 5(g): Average Latency vs Buffer Size

**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.

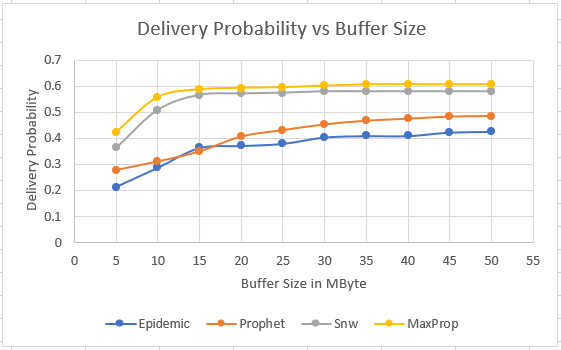


Figure 5(h): Delivery Probability vs Buffer Size

**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.

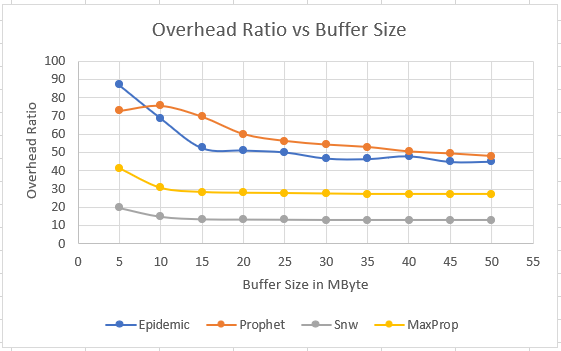


Figure 5(i): Overhead Ratio vs Buffer Size

**5.4 Impact of Varying Msg 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.

**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.

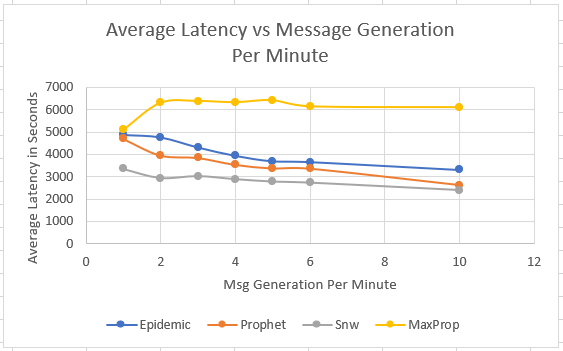


Figure 5(j): Average Latency vs Message Generation Per Minute

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.

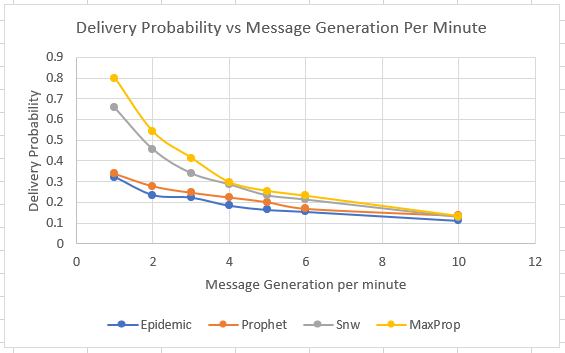


Figure 5(k): Delivery Probability vs Message Generation Per Minute

**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.

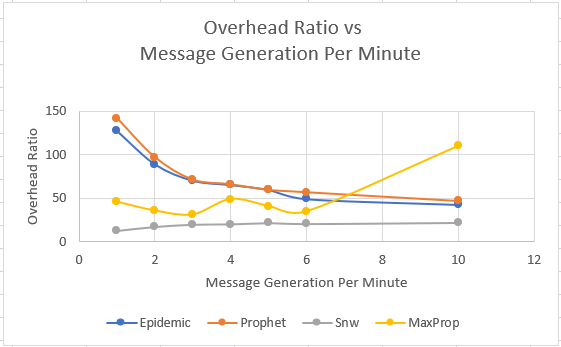


Figure 5(l): Overhead Ratio vs Message Generation per minute

**5.5 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.

**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.

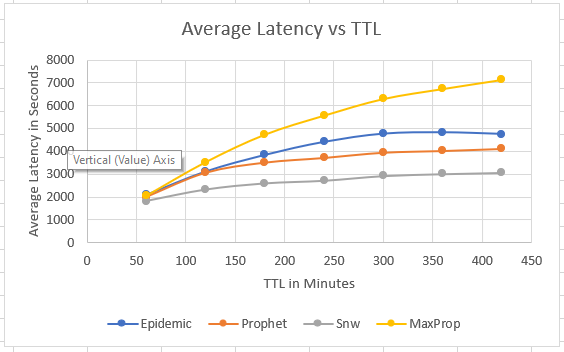


Figure 5(m): Average Latency vs TTL

**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.

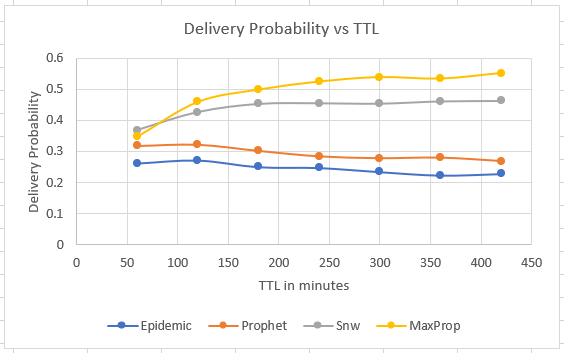


Figure 5(n): Delivery Probability vs TTL

**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.

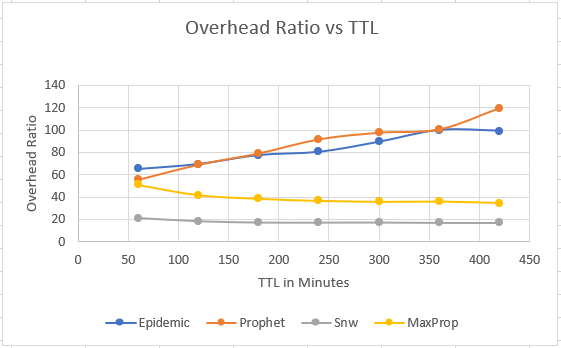


Figure 5(n): Overhead Ratio vs TTL

5.6 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.

**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.

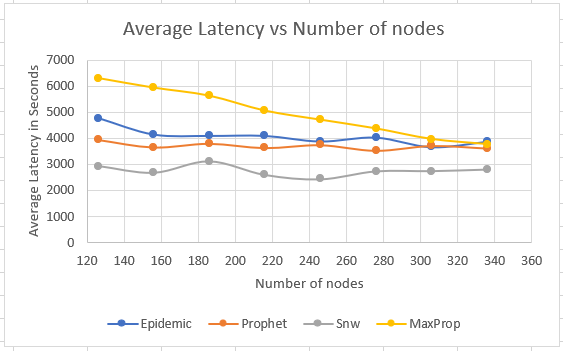


Figure 5(o): Average Latency vs Number of Nodes

**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.

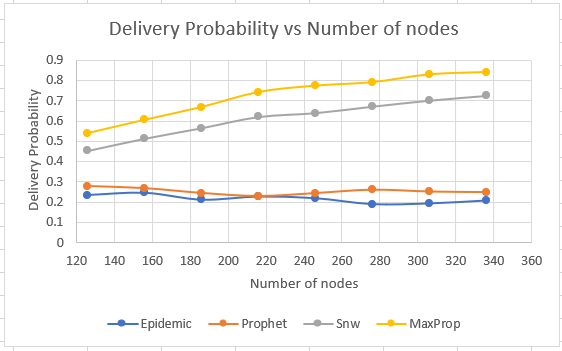


Figure 5(p): Delivery Probability vs Number of Nodes

**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.

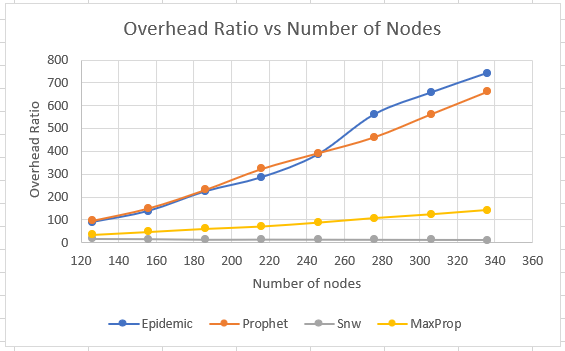


Figure 5(q): Overhead Ratio vs Number of Nodes

**Chapter 6**

**Future Work and Conclusion**

**6.1 Limitation**

In this paper, we have evaluated the performance of DTN routing protocols, i.e., Epidemic, PRoPHET, and Spray-and-Wait in intermittently connected mobile networks (ICMNs). We try 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. As we’ve described four routing protocols but analyzed only three protocols for time limitation. We will try our best to overcome the faults and make our system more reliable.

**6.2 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.

**6.3 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 investigated 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] K. Fall, “A delay-tolerant network architecture for challenged internets,” in Proc. of ACM SIGCOMM, Karlsruhe, Germany, Aug. 2003, pp. 27-34.

[2] Charles E. Perkins and Elizabeth M. Royer, "Ad hoc On Demand Distance Vector Routing," in Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, Feb. 1999, pp. 90-100.

[3] A. Vahdat and D. Becker, “Epidemic routing for partially connected ad hoc networks,” Department of Computer Science, Duke University, Tech. Rep. CS-2000-06, Apr. 2000.

[4] S. Jain, K. Fall, and R. Patra, “Routing in a delay-tolerant network,” in Proc. of ACM SIGCOMM, Portland, OR, 2004, pp. 145–157.

[5] A. Lindgren, A. Doria, and O. Scheln, “Probabilistic routing in intermittently connected networks,” ACM Mobile Comput. and Commun. Review, vol. 7, no. 3, pp. 19–20, Jul. 2003.

[6] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, “Spray and Wait: an efficient routing scheme for intermittently connected mobile networks,” in Proc. of ACM WDTN, Philadelphia, PA, Aug. 2005, pp. 252–259.

[7] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation", in Proc. of Second Int. Conf. on Simul. Tools and Tech., 2009, Rome, Italy.

[8] Tkk/Comnet. Project page of the ONE simulator. [Online]. Available: http://www.netlab.tkk.fi/tutkimus/dtn/theone, 2013.

[9] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, “MaxProp: routing for vehicle-based disruption-tolerant networks,” in Proc. of IEEE INFOCOM, Barcelona, Spain, April 2006.

[10] T. Small and Z. J. Haas, “Resource and performance tradeoffs in delay-tolerant wireless networks,” in Proc. of ACM WDTN, Philadelphia, PA, USA, August 2005, pp. 260–267.

[11] ThrasyvoulosSpyropoulos,Konstantin’s Psounis, CauligiRaghavendra,"Efficient Routing in Intermittently Connected Mobile Networks: The Single-copy Case",IEEE

[12] ThrasyvoulosSpyropoulos,Konstantin’s Psounis, CauligiRaghavendra,"Efficient Routing in Intermittently Connected Mobile Networks: The Multi-copy Case ",IEEE

[13] Artemios G. Voyiatzis."A Survey of Delay- and Disruption-

Tolerant ",IEEE

[14] Hervé Ntareme,"Delay Tolerant Network on smart phones: Applications for communication challenged areas ",Royal Institute of Technology, KTH Forum 120, 164 40 Kista

[15] TheusHossmann, ThrasyvoulosSpyropoulos, and Franck Legendre,"Know Thy Neighbor: Towards Optimal Mapping ofContacts to Social Graphs for DTN Routing",Computer Engineering and Networks Laboratory ETH Zurich, Switzerland , IEEE member

[16] ThrasyvoulosSpyropoulos,Thierry Turletti,Katia Obraczka,"Routing in Delay Tolerant Networks Comprising Heterogeneous Node Populations ",Member, IEEE

[17] Amir Krifa, ChadiBarakat,"Message Drop and Scheduling in DTNs: Theory and Practice",Senior Member, IEEE