DEVELOPMENT OF AN IMPROVED INTEGRATED ROUTING PROTOCOL FOR OPPORTUNISTIC NETWORK WITH CONGESTION CONTROL USING PRE-EMPTIVE DATA EVICTION

BY

Basira YAHAYA

Department of Electrical and Computer Engineering
Faculty of Engineering
Ahmadu Bello University, Zaria

MAY, 2015

DEVELOPMENT OF AN IMPROVED INTEGRATED ROUTING PROTOCOL FOR OPPORTUNISTIC NETWORK WITH CONGESTION CONTROL USING PRE-EMPTIVE DATA EVICTION

By

Basira YAHAYA, B.Eng. (ABU) 2011

MSC/ENG/25808/2012-2013

A THESIS SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES,

AHMADU BELLO UNIVERSITY, ZARIA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE (MSC) DEGREE IN COMPUTER ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

MAY, 2015

DECLARATION

I,Basira YAHAYA declare that this thesisentitled "**Development of an Improved Integrated Routing Protocol for the Opportunistic Networks with Congestion Control using Pre-Emptive Data Eviction**" was carried out by me in the Department of Electrical and Computer Engineering, Ahmadu Bello University, Zariaas part of the requirements for the award of degree of Master of Science in Computer Engineering. To the best of my knowledge, this research work has never been submitted anywhere for the award of any certificate. All literatures used and cited are duly acknowledged in the reference pages.

Basira YAHAYA _____ _____

(Signature) **Date**

CERTIFICATION

This dissertationentitled "**DEVELOPMENT OF AN IMPROVED INTEGRATED ROUTING PROTOCOL FOR THE OPPORTUNISTIC NETWORKS WITH CONGESTION CONTROL USING PRE-EMPTIVE DATA EVICTION**" meets the requirements for the award of Master of Science (M.Sc.) degree and has been approved by the Department of Electrical and Computer Engineering, Ahmadu Bello University, Zaria for its contribution to knowledge and literary presentation.

Chairman, Supervisory Committee_____ _____
(Prof. M. B Mu'azu) Signature Date

Member, Supervisory Committee_____ _____
(Dr. S. Garba) Signature Date

Head of Department _____ _____
(Prof. M. B Mu'azu) Signature Date

Dean, School of Postgraduate Studies _____ _____
(Prof. A. Z Hassan) Signature Date

DEDICATION

This research work is dedicated to Almighty Allah (S.W.T) the ultimate sustainer.

ACKNOWLEDGEMENT

All praises are due to Allah (S.W.T), the most gracious, the most merciful for all the things He has done to my life.Peace and blessing be upon His Prophet, Muhammad (SAW) and his companions.

My endless appreciation goes to the chairman of my supervisory committee, Prof. M. B. Mu'azu, for his tireless effort, support, constant guidance and encouragement from the beginning to the end of this dissertation. My appreciation also goes to my other supervisor Dr. S. Garba for his patience, support and encouragement. May Allah reward both of them abundantly for the encouragement they gave to me right from the days of my undergraduate program. To all my lecturers, I thank you for the valuable contributions, time you created to go through my manuscript and for exposing me to manythings.

Special thanks to my parent Alhaji Yahaya Musa and Hajia Aishatu Bello for their support and prayers

A profound gratitude goes to my hubby Abdullahi Ahmed Aliyu for his kindness, care, patience and understanding during this research work and always. Your encouragement has been helpful to my studies. I must say that people like you are hard to come by and you mean a lot to me. I say thanks to my lovely daughter Hameedah Ahmed for her patience and understanding.

I appreciate my siblings; Rakiya, Aisha, Jamila, Adama, Sa'adatu, Farhana and Mu'ammar, my cousins; Maryam, Fauziya, Yahaya, Rabiu, and Zainab, my nieces; Rahma, Ummita and Aisha, my nephews; Almustapha, Aliyu and Ahmadidi, my inlaws, especially, Yahaya and Aminu for all the care, support and love you have shown me.

I say thank you to Ahmed Tijjani for the help you rendered to the completion of this research work. I am gratefulto my friends and my colleagues; Ene, Zainab M. A., Maimuna, Khadijah, Rukky, Y. Ibrahim, B. O. Sadiq, A. S. Abubakar, E. Obi, Josiah Haruna,Mal. Abdullahi Tukur, Soja, Valentine and the entire people of the Department of Electrical and Computer Engineering.

What I have to say is JAZAKUMMULLAHU-KHAIRAN.

ABSTRACT

The opportunistic network is one of the most interesting evolution of Mobile Ad-hoc Network (MANET).In an opportunistic network,information about the context in which users communicate is notavailable as there is no dedicated complete path from a source to a destination. This makes the design of an efficient routing protocol for an opportunistic network difficult. The research is aimed at the development of an improved integrated routing protocol in opportunistic networks. Simulation was carried out using the opportunistic network environment (ONE) simulator.This report presents the modeled opportunistic network using the epidemic, the prophet and the integrated routing protocol. Four pre-emptive congestion control strategies (use of acknowledgement, buffer size advertisement, data centric method and the avoidance of duplication) were developed and incorporated into the integrated routing protocol. The integrated routing protocol outperformed the epidemic and the prophet routing protocols by 5.4% and 4.3%, respectively with respect to the delivery probability. Results showed that the duplication avoidance improved the integrated routing protocol because it reduced the packet loss and improved the delivery probability when tested on a 10-node test model and on the benchmark Helsinki simulation area. Duplication avoidance reduced the packet loss by 58% and improved the delivery probability by 4% at the end of the simulation time of 43200 seconds when compared with the delivery probability and packet loss of the original integrated routing protocol on Helsinki which are 0.1048 and 670, respectively. The use of acknowledgement, buffer size advertisement, data centric method reduced the packet loss by 2.5%, 57% and 57%, respectively but did not improve on the delivery probability significantly. The final delivery probabilities ofthe use of acknowledgement, buffer size advertisement, data centric method are 0.105, 0.103 and 0.105, respectively. These results showed that duplication avoidance is the preferred congestion control strategy for the integrated routing considered in this research.

TABLE OF CONTENTS

DEVELOPMENT OF AN IMPROVED INTEGRATED ROUTING PROTOCOL FOR OPPORTUNISTIC NETWORK WITH CONGESTION CONTROL USING PRE-EMPTIVE DATA EVICTION

DECLARATION ..... **iv**

CERTIFICATION ..... **iv**

DEDICATION ..... **v**

ACKNOWLEDGEMENT ..... **vi**

ABSTRACT ..... **vii**

LIST OF FIGURES **xi**

LIST OF TABLES **xiii**

LIST OF ABBREVIATIONS ..... **xiv**

CHAPTER ONE: INTRODUCTION

1.1 Background 1
1.2 Problem Statement 3
1.3 Aim and Objectives 3
1.4 Methodology 4
1.5 Significant Contribution 5
1.6 Thesis Organization 5

CHAPTER TWO:LITERATURE REVIEW

2.1 Introduction 6
2.2 Review of Fundamental Concepts 6
2.2.1 Opportunistic networks 6
2.2.2 Opportunistic network node definition 9
2.2.3 Opportunistic networks and mobile Ad-hoc network 10
2.2.4 Routing in opportunistic networks 12
(i) Dissemination-based Routing *13*
(ii) Context-based Routing *13*
(iii) Routing Based on Fixed Infrastructure *14*
(iv) Routing Based on Mobile Infrastructure *14*
2.2.4.1 The Epidemic Routing protocol *14*
2.2.4.2 The Prophet Routing Protocol *15*
2.2.4.3 Integrated Routing Protocol *16*
2.2.4.4 Integrated routing protocol probabilities *17*
2.2.4.5 Routing strategies of the integrated routing *18*
2.2.5 Congestion in opportunistic networks 19
2.2.6 Buffer management 20
2.2.7 Pre-emptive data eviction 21

2.2.8 Evaluating the Performance of DTN Routing Protocols 23
2.2.9 The ONE Simulator 24
2.3 Review of Similar Works 26

CHAPTER THREE: MATERIALS AND METHODS

3.1 Introduction 35
3.2 Model Test Case 35
3.3 Installation and Configuration 36
3.4 Simulation 36
3.5 Visualization 47
3.6 Simulation of the Improved Integrated Routing Protocol with Congestion Control. 47
3.6.1 Buffer eviction using acknowledgement 47
3.6 2 Buffer size advertisement 48
3.6.3 Data-centric node congestion avoidance 49
3.6.4 Duplication avoidance 50
3.7 Validation of the congestion control strategies 51

CHAPTER FOUR: RESULTS AND DISCUSSION

4.1 Introduction 57
4.1 The Performance Evaluation of the Congestion Control Strategies 57
4.3 General Discussion 66

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

5.1 Introduction 68
5.2 Conclusions 68
5.3 Limitations 69
5.4 Recommendations for Further Work 69

REFERENCES 70

Appendix A1 .....

The Program Class for Epidemic Router 73

Appendix A2

Epidemic Routing Settings 74

Appendix B1

Program for Prophet Routing 76

Appendix B2

Prophet Settings 82

Appendix C1

Setting for Integrated Routing on 10-node test case 84

Appendix D1

Settings for integrated routing on Helsinki 86

Appendix E

program class for Helsinki 89

Appendix F1

Variation of Delivery Probability on 10 Nodes 93

Appendix F2

Variation of Packet Loss on 10 Nodes 94

Appendix F3

Time Variation of Delivery Probability on Helsinki 95

Appendix F4

Time Variation of Packet loss in Helsinki 96

LIST OF FIGURES

Figure 2.1: An Opportunistic Network 7

Figure 2.2: Typical Features of Opportunistic Network 10

Figure 2.3: Taxonomy of Routing/Forwarding Techniques for Opportunistic Networks 12

Figure 2.4: An Example of Transitive Communication 17

Figure 2.5: An Overview of the ONE Simulator GUI 25

Figure 3.1: Beginning the Epidemic Simulation 38

Figure 3.2: Nodes Building-up process in the Epidemic simulation 38

Figure 3.3: Congestion in the Epidemic Simulation 39

Figure 3.4: Communication between Nodes in Prophet Routing Protocol 41

Figure 3.5: Congestion in Prophet Routing Protocol 41

Figure 3.6: Simulating the Integrated Routing Protocol 43

Figure 3.7: Helsinki Simulation Area 45

Figure 3.8: An Overview of the Helsinki Integrated Routing Simulation 45

Figure 3.9: The Flow Chart of the Simulation Process 46

Figure 3.10: Use of Acknowledgement 48

Figure 3.11: Use of Advertisement 49

Figure 3.12: Data Centric Node Congestion Avoidance 50

Figure 3.13: Duplication Avoidance 51

Figure 3.14:Use of Acknowledgement on Helsinki Simulation Region 52

Figure 3.15:Use of Buffer Size Advertisement on Helsinki Simulation Region 53

Figure 3.16:Data Centric Node Congestion Avoidance on Helsinki Simulation Region 54

Figure 3.17: Duplication Avoidance on Helsinki Simulation Region 55

Figure 3.18: Flow Chart of the Routing Algorithm with Congestion Control 56

Figure 4.1:Variation of Packet Loss with Time 61

Figure 4.2:Variation of Delivery Probability with Time 62

Figure 4.3: Variation of Packet Loss with Time for Helsinki Region 65

Figure 4.4:Variation of Delivery Probability with Time for Helsinki Region 66

LIST OF TABLES

Table 4.1: Message Statistical Report 58

Table 4.2: Message statistical report for the Helsinki integrated routing 58

Table 4.3: Results for IRP Simulation for the 10-Node Test Case 59

Table 4.4: Results of Simulating the IRPCC for the 10-Node Test Case 60

Table 4.5: Result for the IRPCC for Helsinki Region 63

Table 4.6:Result for the IRPCC for Helsinki Region64

LIST OF ABBREVIATIONS

Acronyms	Definition
DTN	Delay Tolerant Network
FIFO	First-in-First-out
GUI	Graphic User Interface
IDE	Integrated Development Environment
IRP	Integrated Routing Protocol
IRPCC	Integrated Routing Protocol with Congestion Control
LAN	Local Area Network
NS2	Network Simulator-2
ONE	Opportunistic Network Environment
POR	Practical Opportunistic Routing
RAM	Random Access Memory
TCP	Transmission Control Protocol
TTL	Time To Life
Wi-Fi	Wireless Fidelity

CHAPTER ONE

INTRODUCTION

1.1 Background

Traditionally, communication between two or more devices in a network uses infrastructure-based wired networks. In some instances, however, such communication is limited as it becomes difficult to expand the network due to cost, environment, etc. This, coupled with the need for mobility and independence of network infrastructure, brought the need for wireless networks (Dinakar *et al.*, 2013).

A wireless network uses access points and base stations (for mobility) to ensure communication among users and between users and the network. Each base station and access point usually has a maximum number of defined users it can ideally handle. These traditional wireless networks, however, have the following constraints (Dinakar *et al.*, 2013) :

(i) Centralized administration is usually required.

(ii) Power requirements to serve all the base stations and access points.

(iii) Usually capital intensive to deploy and scale up.

(iv) Difficult to maintain.

The opportunistic network is an autonomous connection of users that communicate over a relatively bandwidth constrained wireless network. According to Kaur & Kaur (2009) and Verma & Srivastava (2012), it has the following basic features:

(i) It has wirelessly connected nodes which can be fixed or mobile (a node is an electronic device that is attached to a network and is capable of sending, receiving or forwarding information over communication channel).

(ii) A complete path between two nodes aspiring to communicate does not exist.

(iii) It does not have a fixed communication range.

(iv) Its routes are dynamically built as any node can opportunistically be used as the next hop provided it is likely to bring the message closer to the final destination.

(v) Network topology is also flexible as it can change at any time.

Since there is no direct connection between the source and the destination coupled with the fact that routes are built dynamically, designing an efficient routing protocol for opportunistic networks is a very critical issue. Information about the context in which users communicate is also a key piece of knowledge required to design an efficient routing protocol. In opportunistic networks, this information is not available, thus further making it difficult to design such routing protocols. Context information is the users' working address and institution, working environment and users behavior, probability of meeting another user or visiting a particular place. It is important to identify suitable forwarders based on context information about the destination (Verma & Srivastava, 2012).

The two main routing techniques for opportunistic networks as presented in literatures (Kaur & Kaur, 2009; Orozco, *et al.*; Verma & Srivastava, 2012) are:

i. The 'Prophet' (context-aware) routing.

ii. The 'Epidemic' (context-oblivious) routing.

'Prophet' is able to learn automatically from the past communications opportunities determined by user's mobility pattern and use them in the future. This self-learning feature is absent in 'Epidemic' as it floods the message all over the network, thereby causing congestion and creating high overhead. Epidemic is the only solution when context information about the users is not available (Lindgren, *et al.*, 2003). Verma & Srivastava,(2012) showed that 'epidemic' is better (in terms of message delay and delivery probability) in absence of context information while 'prophet' gives better result with context information available.

The integrated routing protocol for opportunistic network was developed by Verma & Srivastava (2012) in order to sum up the merits of the two protocols mentioned above. The integrated routing technique uses context data as soon as it is available and falls back to dissemination-based routing when context information is not available.

1.2Problem Statement

Information about the context in which users communicate is not available in an opportunistic network as there is no dedicated complete path from a source to a destination. This makes the design of an efficient routing protocol for an opportunistic network difficult(Verma & Srivastava, 2012). Utilizing the characteristics of opportunistic network to design a buffer management strategy is still an open issue (Pan et al., 2013).This research is aimed at improving the existing integrated routing protocol by incorporating buffer management into it as a means of controlling congestion using pre-emptive eviction of data.

1.3Aim and Objectives

The aim of this research is to develop an improved integrated routing protocol with congestion control using pre-emptive eviction of data.

The objectives are as follows:

(i) Development of the improved integrated routing protocol with congestion control via buffer management.

(ii) Simulation and comparison of performance of the improved integrated routing protocol with that of the conventional integrated routing protocol.

(iii) Validation of the developed model with a real-life scenario (Helsinki Region).

1.4 Methodology

The methodology adopted in carrying out this research is as follows:

i. Modeling of the Test Case Scenario.

ii. Development of the epidemic routing protocol and its simulation using the opportunistic network environment (ONE) simulator.

iii. Development of the prophet routing protocol and its simulation using ONE simulator.

iv. Development of the integrated routing protocol as a hybrid of the prophet and the epidemic routing protocol and its simulation using ONE simulator.

v. Simulation of the improved integrated routing protocol with congestion control using buffer management.

vi. Comparison of performance of the conventional integrated routing protocol with that of the improved integrated routing protocol.

vii. Validation of the improved integrated routing protocol using a real-life scenario (Helsinki region).

1.5 Significant Contribution

Many researchershave worked on designing a routing protocol for the opportunistic networks. Also, some researches have tried to address the problem of congestion in opportunistic networks.The significant contributions of this research work are as follows:

i. Development of an improved integrated routing protocol for the opportunistic network with Duplication-Avoidance based Congestion Control strategy
ii. Achieved a 58% reduction in the packet loss and 4% improvement in the delivery probability when tested on the benchmark Helsinki simulation area,

.

1.6 Thesis Organization

The general introduction has been presented in chapter one. The rest of the chapters are structured as follows: Detailed review of fundamental concept on the opportunistic networks, ONE simulator and areview of similar research works is presented in chapter two. In chapter Three, modeling of thetest case scenario, configuration, installation as well as simulation are presented. The results, its analysis and discussion are presented in chapter four. Conclusion and recommendations are discussed in chapter five. Quoted references and appendices are also provided at the end of this thesis.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The literature review provides the background knowledgerelevant to the scope of this research. This is provided in two perspectives: Review of fundamental concepts and the review of similar works.

2.2 Review of Fundamental Concepts

The review of relevant fundament concepts is carried out in this section. This review covers the concept of the opportunistic networks, routing in opportunistic network, congestion control strategies, as well as the tools for evaluating the performance of Delay Tolerant Networks (DTN).

2.2.1 Opportunistic networks

Opportunistic networking has received considerable interest from the research community in recent years. (Pelusi, *et al.*, 2006) . In opportunistic networks, mobile nodes are able to communicate with each other even if the route connecting them does not exist. However, it is difficult to design an efficient routing protocol for an opportunistic network. This is because the topology of such networks is continuously changing. Also in the opportunistic networks, information about the context in which users behave is missing. All these challenges are making it difficult to decide the best routing protocol for an opportunistic network.Opportunistic network is a delay tolerant network and it can be formed by small devices (fixed or mobile) communicating with each other over a wireless link. It is a type of network in which network contacts are intermittent or where link performance is highly variable. Hence, in order to make communication possible, the intermediate node may take custody of data when there is no

connection and forward it when connectivity resumes (Huang, *et al.*, 2008). Congestion occurs when a node's buffer become full. The lack of end-to end path between the source and the destination coupled with limited resources and high delays affect TCP/IP protocols in opportunistic environments. There is no propagation in the network making many Internet protocols (such as TCP) designed to assume quick return of acknowledgement and data susceptible to failure (Huang, *et al.*, 2008). A typical opportunistic network is illustrated in Figure 2.1.

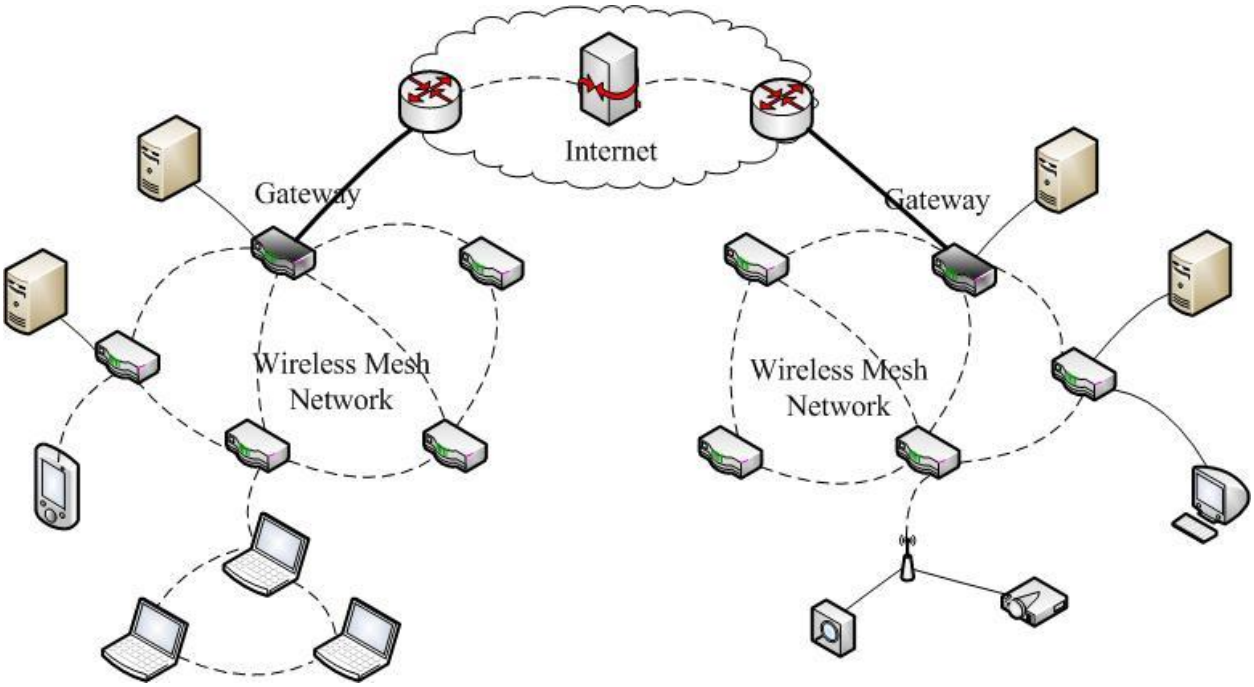


Figure 2.1: An Opportunistic Network (Dinakar, *et al.*, 2013).

Some of the characteristics of opportunistic networks include(Dinakar *et al.*, 2013):

(1) When the nodes move away or turn off their power, the link may shutdown resulting in unstable connectivity.

(2) Network partitioning may occur when no paths exist between the source and the destination.

(3) The node communicates in a store-and-forward mechanism, etc.

The store-and-forward mechanism is alternatively called packet switching. Its procedure is as follows (Lindgren *et al.*, 2003a; Zhao *et al.*, 2012):

(1) The message is divided into fixed-length packet.
(2) Every channel has input and output buffers for one entire packet.
(3) Every packet is individually routed from the source to the destination.
(4) One step of the store-and-forward switching is called hop. In a hop, the whole packet from one output buffer is copied to the next input buffer.
(5) Routing decisions are made by each intermediate router until the message reaches its final destination.

Some of the challenges in opportunistic networks include (Dinakar *et al.*, 2013; Huang, *et al.*, 2008):

(1) CONTACT; nodes contact with other nodes unpredictably because of node mobility.
(2) LINK DISCOVERY; this is used by networks to advertise their identity, capability and neighbors.
(3) STORAGE CONSTRAINT; the required storage space increases as a function of the number of messages in the network. Nodes store messages in their buffers when they are out of communication range as a result congestion which is unpreventable in this network, etc.

These networks are usually built around people, typically pedestrians. They are usually limited in speed, and have much slower propagation than wired or wireless links making delays in such

networks remain. The unstructured nature of opportunistic networks makes them extremely difficult to put any performance guarantees on data dissemination(Ristanovic, 2012).

The opportunistic network is challenging as its structure or connection is changing continuously. In order to resume communication after a break in connectivity due to unstable or incomplete path, unstable nodes (that keep the data during collapse periods) are used to forward the data when connectivity resumes (Dinakar, *et al.*, 2013). The opportunistic network is a self-configuring network that looks like an ad-hoc network and it uses all types of communication media like wireless-fidelity (Wi-Fi), Bluetooth, wireless, satellites, etc.

2.2.2 Opportunistic network node definition

In opportunistic network, a node is a device with short-range wireless communication capabilities. Communication range between two connected nodes is within walking distance of between 100-300 meters (Yogi & Chinthala, 2014). A node can be a mobile device carried by people or a fixed device. A mobile node consists of a user carrying a mobile device that acts as an opportunistic network node (for example: personal digital assistant, mobile phones etc.). A fixed node is a device kept at a dedicated location, that is not under the direct control of a user. An example of a fixed node is the information sprinkler that is connected to the backbone network. The backbone network may be a wired network that joins a set of information sprinklers and synchronizes their operation. The information sprinkler can operate in a sprinkler mode (that is information is only dispersed), or in a sink mode (that is information is only collected), or in both modes together. (Yogi &Chinthiala, 2014). Figure 2.2 shows three information sprinklers and an optional sprinkler backbone, the connection link shows two adjacent nodes close enough to exchange message with one and other. The communication range

indicated as dashed spheres is not an ideal sphere in practice due to communication signal interference with the surroundings.

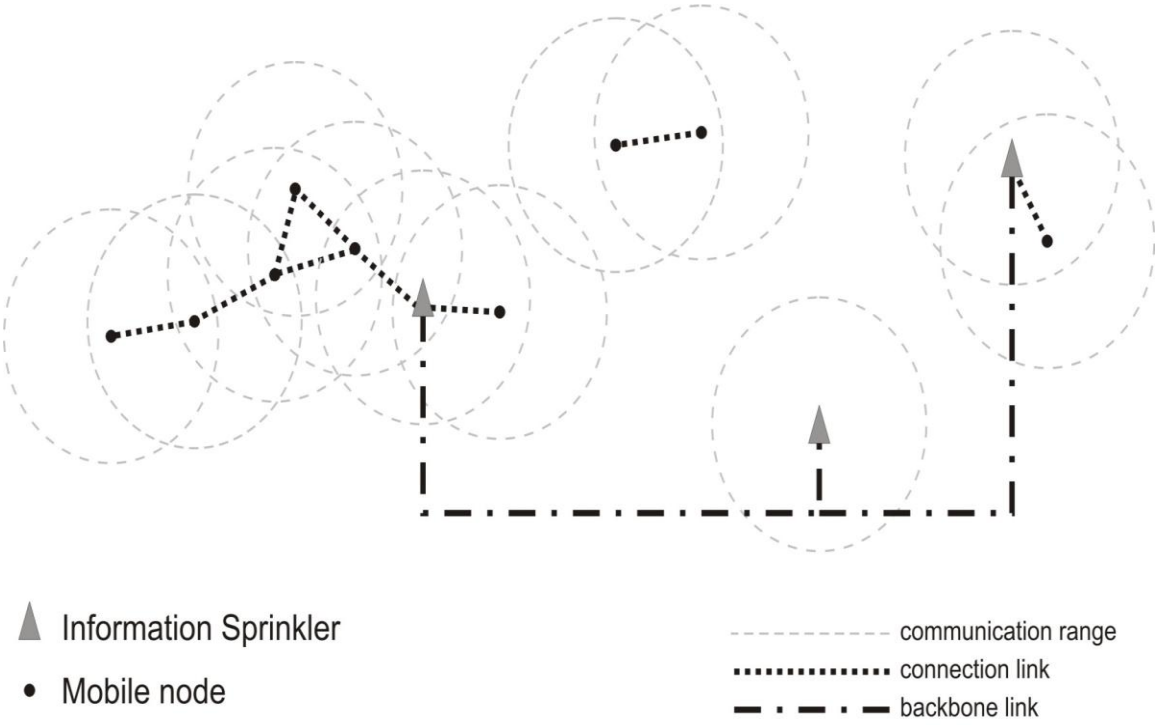


Figure 2.2: Typical Features of Opportunistic Networks (Yogi & Chinthala, 2014).

2.2.3 Opportunistic networks and mobile Ad-hoc network

Opportunistic network is one of the most interesting evolution of Mobile Ad-hoc Network (MANET). MANETS are autonomous, infrastructure-less, self-configuring networks which require knowledge of the network topology while opportunistic networks can be both infrastructure-less or infrastructure-based and do not necessarily require knowledge of the network topology. MANETS usually focus on end-to-end routing between mobile nodes, but opportunistic networks go beyond MANETS as end-to-end routes can rarely exist. As a result, an opportunistic network is typically considered in environments with intermittently connected wireless nodes, for example, opportunistic networks have been deployed as means of bridging

between isolated rural areas, wildlife monitoring, and networking using buses that follow predictable routes (Pelusi et al., 2006). An opportunistic network can resume link failure and forward data in a store-and-carry mechanism which is not possible in MANETs. As a result, opportunistic networks overcome the technical challenges of MANETS to a better extent which parallels the increase in performance and can satisfy the user in a better way (Dinakar, *et al.*, 2013; Ristanovic, 2012).

Opportunistic network solely exposes the one-hop communication scheme, where only directly connected nodes exchange message (for example, by learning about the information they are looking for), this implies that the opportunistic network resides on the application layer. MANETs resides on the network layer (Yogi and Chinthiala, 2014).

MANETS and opportunistic networks have the following combined applications (Dinakar, *et al.*, 2013):

(1) Tactical Networks (like in Military Operation).

(2) Emergency Service (for example rescue operations, disaster recovery and hospital).

(3) Education (for Campuses and Universities, virtual classrooms, during meetings, etc.).

(4) Coverage Extension: they can extend distance beyond the capabilities of typical cables and hence can be used to:

(i) provide backup incase of normal network failure,

(ii) provide link for portable or temporary workstations,

(iii) overcome situations where normal cabling is difficult or financially impractical, and

(iv) connects remotely mobile users or networks

2.2.4 Routing in opportunistic networks

Routing protocols decide which node to forward data to based-on specific network characteristic they observe. Routing performance improves when more knowledge about the expected topology is used. However, this knowledge is not readily available in opportunistic networks such that trade-off must be made between performance and knowledge requirement(Kaur & Kaur, 2009; Pelusi, *et al.*, 2006; Verma & Srivastava, 2012).

Depending on the routing approach, opportunistic networks may be classified at a very top level into two categories; infrastructure-less and infrastructure-based networks. The infrastructure-less approach can be further divided as dissemination-based and context-based. In the infrastructure-based case, it is divided into the mobile infrastructure and fixed infrastructure (Pelusi, *et al.*, 2006). A typical taxonomy of opportunistic network is shown in Figure 2.3

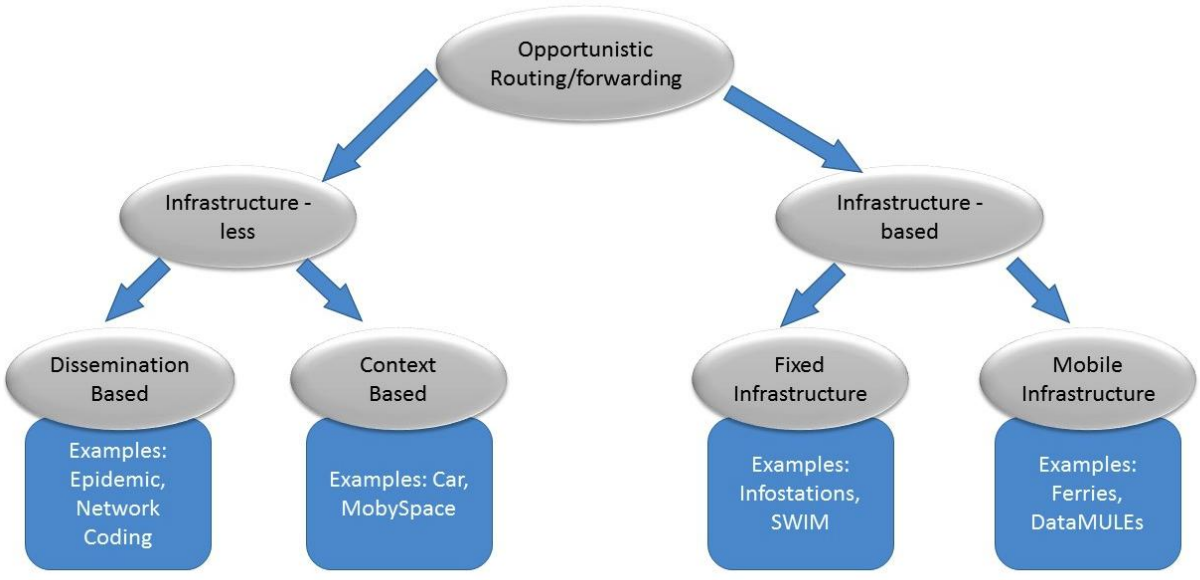


Figure 2.3: Routing/Forwarding Techniques for Opportunistic Networks (Pelusi, *et al.*, 2006)

(i) Dissemination-based Routing

It is also called the context oblivious routing. It provides a final delivery of messages to random destinations with minimal assumption of topology and connectivity of the network (Pelusi *et al.*, 2006). The heuristic behind this policy is that, the message should be broadcasted all over the network (flooding) expecting that it will eventually reach its destination. This technique works well in a highly mobile network where the contact opportunities needed for data diffusion are common. They limit message delay but consume resources (Pelusi *et al.*, 2006; Verma & Srivastava, 2012).

(ii) Context-based Routing

Mobility and contact pattern in the real world scenarios are likely to be predicted (for example, human mobility have a regular traffic pattern based on repeating behavioral pattern (like going to work, returning home, going to the market, and even social interaction)). It implies that if a node has visited a location or come in contact with a certain node several times before, it is likely to visit that location or meet that node again. Context-based routing utilizes this information to route data. In order to accomplish this, the protocol maintains a delivery predictability metric at every node. The delivery predictability metric refers to message delivery probability between two nodes(Pelusi *et al.*, 2006).

Hence, it is able to learn the past communication opportunities determined by users' mobility pattern and uses them efficiently in the future. Context-based routing provides congestion control, acceptable QoS with lower overhead. However, in the absence of context information, it provides high overheads and message delay (Pelusi *et al.*, 2006; Verma & Srivastava, 2012)

(iii) Routing Based on Fixed Infrastructure

With infrastructure-based routing, a node intending to deliver a message keeps it until it comes within reach of a access point belonging to it. Access points are gate ways which can provide Internet access or can be connected to a LAN(Kaur & Kaur, 2009).

Two protocol variations exist, one in which only node-to-base station communication is possible, while the second allows both node-to-base station and node-to-node communication. That is, a node intending to send a message to a destination node delivers the message to the base station directly if they are within communication range, else it delivers the message opportunistically to a near node that will forward it to the base station when encountered. Based on these two examples, fixed base stations play passive roles in opportunistic forwarding strategy because they act as information sinks (Kaur & Kaur, 2009).

(iv) Routing Based on Mobile Infrastructure

This technique is also called the Carrier-Based routing. Nodes of the infrastructure are mobile data collectors, as they move in either predetermined or arbitrary routes around the network area gathering messages from nodes they pass by.

When only node-to-carrier communications are allowed, these mobile nodes can be the only entities responsible for message delivery, thereby increasing connectivity in a sparse network by assuring that isolated nodes can also be reached (Pelusi, *et al.*, 2006).

2.2.4.1 The Epidemic Routing protocol

The epidemic protocol simply forwards all data to nodes in the network with minimal information of network connection or organization. Higher delivery probability is guaranteed

even though it saturates the network with many copies of the same data.In the epidemic routing protocol, each host maintains two buffers, one for storing the message it originated and the other for messages it is buffering for other host. Each mobile device stores a summary vector which contains a summary of the messages stored in its buffer. When two nodes come into communication range, they exchange their summary vector, then each node checks which message is missing and then request for the missing message. By so doing, the messages get flooded in the network, and eventually, the message reaches its destination.

The epidemic routing protocols are the only solution when context information about the user is absent, but they generate high overhead, suffer from network contention and may potentially lead to network congestion(Jones & Ward, 2006; Pelusi *et al.*, 2006; Verma & Srivastava, 2012).

2.2.4.2 The Prophet Routing Protocol

Prophet is a variant of the epidemic routing protocol that uses past history of encounters and transitivity. It minimizes resource usage while still attempting to achieve the best case routing capabilities of the epidemic routing in intermittently connected networks. That is, when properly configured, prophet gives better performance in cases where resources (buffer space and bandwidth) are limited. The prophet protocol uses the same concept as the context-based routing protocol by learning the past communication opportunities determined by users mobility pattern and using them efficiently in the future to route information. Like the context-based protocol, it is then able to maintain a delivery predictability metric at every node and also provide congestion control, as well as acceptable QoS with lower overhead.

However, in the absence of context information, it provides high overheads and message delay (Pelusi *et al.*, 2006; Verma & Srivastava, 2012)

2.2.4.3 Integrated Routing Protocol

In some cases, it is not guaranteed that a node with a higher delivery probability will be discovered in reasonable time. Also, some nodes may be new in the network, implying that their context information is not spread in the network. As a result of these, the integrated routing protocol was made to combine the features of the prophet and the epidemic routing protocol. It uses delivery predictability information when it is available and then uses dissemination-based routing when context information is not available.

It allows messages to be buffered at intermediate nodes thereby exploiting the mobility pattern of nodes to bring messages closer to their final destinations by transferring them to other nodes as they meet. Figure 2.4 shows the mobility of nodes in a scenario that can be used to deliver a message to its final destination. The physical position of four nodes (A, B, C, and D) at four time instants is represented in the four sub-figures (a) to (d). At the beginning, node A has a message intended to be delivered to node D, but there is no direct path between nodes them due to limited range of wireless connections. The mobility of nodes allows the message to first be transferred to node B, and then to node C when they come in communication range, and eventually to node D. This is known as transitive networking.

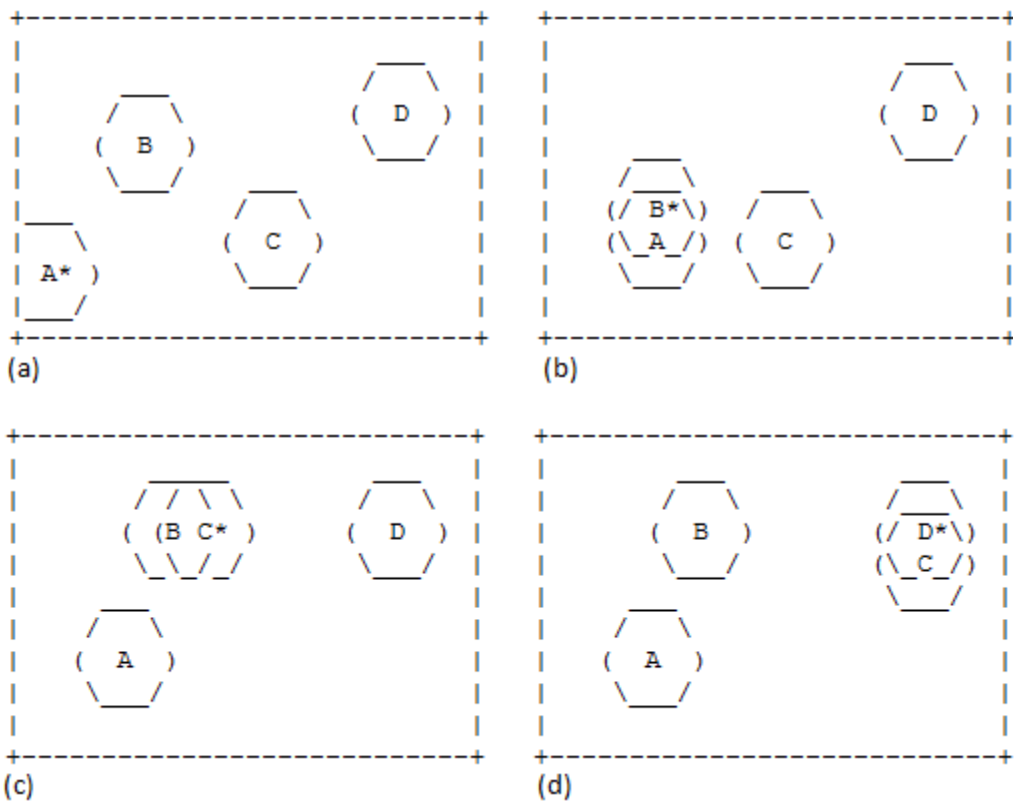


Figure 2.4: An Example of Transitive Communication (Lindgren et al., 2003).

Nodes need to know the contact probabilities to all other nodes in the network, every node maintains a probability matrix.A node then updates its matric with that of other nodes if the other node has a more recent update time attribute so that the two nodes will have identical contact probability matrices after communication. The integrated routing protocol was observed to overcome the challenges of other routing schemes with respect to opportunistic networks (Verma & Srivastava, 2012).

2.2.4.4 Integrated routing protocol probabilities

The metric is updated whenever a node meets with other nodes, so that nodes that are often met have a high message delivery probability. When node x meets node y, the delivery probability of node x for y is updated by (Verma & Srivastava, 2012):

$P'_{xy} = P_{xy} + (1+P_{xy})\ P_0$(2.1)

Where $P_0$is an initial probability (P_0 ranges between 0 and 1), P'_{xy} is the current delivery probability of node x for y and P_{xy} is former delivery probability of node x for y.

When node x does not meet with node y for some predefined time, the delivery probability decreases by (Verma & Srivastava, 2012):

$P'_{xy} = \gamma^k P_{xy}$(2.2)

Where γ is the aging factor (γ< 1), and k is the number of time units since the last update.

When node x receives node y's delivery probabilities, node x may compute the transitive delivery probability through y to z by (Verma & Srivastava, 2012):

$P'_{XZ} = P_{XZ} + (1 + P_{XZ})P_{xy}P_{YZ}\beta$ (2.3)

Where β is a design parameter for the impact of transitivity. According to Verma & Srivastava (2012) $\beta \in [0,1]$.

2.2.4.5 Routing strategies of the integrated routing

When a message arrives at a node, there might not be an available path to the destination. As such, the node might have to buffer the message and upon each encounter with another node, a decision must be made on whether or not to transfer a particular message. Furthermore, it may also be sensible to forward a message to multiple nodes to increase the probability that a message is really delivered to its destination (Verma & Srivastava, 2012). Whenever a node meets with other nodes, they all exchange their messages (or their probability matrices). If the destination of a message is the receiver itself, the message is delivered. Otherwise, if the probability of delivering the message to its destination through this receiver node is greater than or equal to a certain threshold, the message is stored in the receiver's storage to forward to the destination. If

the probability is less than the threshold, the receiver discards the message. If all neighbors of sender node have no knowledge about destination of message and sender has waited more than a configured time, the sender will broadcast it to all its current neighbors. This process will be repeated at each node until it reaches its final destination (Verma & Srivastava, 2012).

A routing protocol was developed where a message is transferred to other nodes when two nodes meet if the delivery probability to the destination of the message is higher than that of the other node. Deciding when to forward a message is not an easy task(Verma & Srivastava, 2012). In some cases it might be sensible to select a fixed value and only give a message to nodes that have delivery probability greater than that fixed value for the destination of the message. On the other hand, when encountering a node with low delivery predictability, it is not certain that a node with a higher metric will be encountered within reasonable time. It may be possible that a destination is new and context information about that destination is not spread in network. To address these problems a new concept was introduced whereby, the integrated routing approach distributes copies of messages to all its neighbors (same as flooding-based techniques), after a configurable time, when nodes have not received any context information about the destination of message (Verma & Srivastava, 2012).

2.2.5 Congestion in opportunistic networks

Nodes in opportunistic networks have limited resources, but they are still willing to altruistically forward messages for other nodes in the network. When forwarding is done through replication, there is a fear for congestion at the nodes. In opportunistic networks, no end-to-end connection is established and therefore congestion cannot be detected and controlled by a feedback loop (Bjurefors, 2014)

The challenge is how to avoid congestion without the feedback loop, using just local information at nodes. Avoiding congestion can be done by pre-emptive eviction of data items from the buffers of the nodes. Congestion algorithms have the potential to improve delivery ratio and decrease average delay. With an uncontrolled eviction policy there is a potential risk that all replicated copies of the data may be evicted before all destinations have been reached, hence decreasing the delivery ratio (Bjurefors, 2014).

2.2.6 Buffer management

Opportunistic network characterized by sparse connecting, fault tolerance and forwarding through mobility. Hence, many routing protocols (Spyropoulous et al., 2005; Santos et al., 2011; Journi & Jorg 2008; Nguyen et al., 2007; Vahdat & Becker 2000)adopt the flooding base scheme to deal with the problem of network partitioning and unpredictability in connections to increase the message delivery. Node receives packet, stores them is their buffer, carry them when moving and pass them to other nodes when they meet each other. The excessive multi-copies spraying in the network cause serious network congestion by exhausting the nodes, there by influencing the performance of transmission (Pan, et al., 2013). Therefore, buffer management plays an important role in improving the routing performance of the network.

Buffer management does the following (Pan, et al., 2013):

a. Deletes redundant information in the system.
b. Formulate reasonable queue strategy.
c. Controls congestion.
d. Build up cache replacement policy.

Since congestion in an opportunistic network occurs when a node's buffer is overwhelmed with messages that may not have been forwarded to another node, a node has to evict messages from its buffer in order to reduce the number of items in its buffer. Also, nodes in an opportunistic network cannot rely on acknowledgements since a contemporaneous end-to-end connection may never exist. The nodes that created messages do not know that a node in the network is congested. The congested node itself has to reduce or solve the problem, basing its decision on what to drop from the buffer using local information available at the nodes. Information can be collected from other nodes (Bjurefors, 2014).

Several methods have been developed on how to deal with congestion in opportunistic networks, which range from buffer advertisement beacon to algorithm to off load data.

2.2.7 Pre-emptive data eviction

In order to reduce congestion, there must be a way to select and eliminate messages (pre-emptive eviction of messages). These strategies are described as follows (Bjurefors, 2014):

i. Buffer eviction using acknowledgement

The use of acknowledgement in opportunistic network differs from that of legacy network. In the Internet, TCP acknowledgements are used to show that the message has reached its destination hence, the message does not need to be retransmitted and nodes discard the message from their buffer. But this is not possible in opportunistic networks as there is no end-to-end connectivity. To use acknowledgement in an opportunistic network, time-to-live that is longer than the message must be attached, else, a copy of the message can still live on the network.

The advantage with acknowledgement is that nodes are sure that the message has been delivered to the destination before the message is evicted from the buffer. The disadvantage is that it takes time for an acknowledgement to disseminate in the network. Acknowledgements also need time-to-live (TTL) to avoid the packets from lingering in the network (Bjurefors, 2014).

ii. Buffer size advertisement

Congestion collapse (congestion caused by aggregated undelivered packet that waste bandwidth) by replication can be avoided by nodes sharing their buffer utilization statistics with neighboring nodes. With those statistics, a node can estimate the congestion level at neighboring nodes. By advertising their free buffers, the neighboring nodes can take decision on what to forward and how much to transfer, making it possible to avoid overloading the node and to prioritize message in order to use the buffer space as efficiently as possible (Bjurefors, 2014).

iii. Data-centric node congestion avoidance:

In the data-centric strategy, messages are forwarded based on the interest in the data. Using this principle in opportunistic networks, it is assumed that a node is more likely to provide buffer space for data items that are of interest to the node itself. It is also assumed that forwarding nodes keep data that they are interested in, which makes the interested forwarding node to become the new source (Bjurefors, 2014). Nodes can choose to evict data that is of little interest to the nodes in the network, as few nodes will request for that data. Nodes can also decide to evict data that is of high interest by assuming that other nodes probably have the data, since it will be frequently requested for and shared. The disadvantage of this strategy is that there is an increase in storage of data items that will never be forwarded or have already reached all nodes interested in the data. Either they are data items that no node is interested in, or data items that

have been replicated many times. These data item become stale and consume buffer space, which could have been used to forward other data item (Bjurefors, 2014).

iv. Duplication Avoidance:

In this strategy, a node keeps only messages it does not have in its buffer. When a node comes into communication range with other nodes, it receives a message and checks if it has the same message in its buffer. If it has the message, it then deletes the message in order to avoid duplication. This method avoids unnecessary wastage of buffer space by ensuring that no same copies of messages are kept in the buffer.

The four aforementioned congestion control strategies were used to improve the performance of the integrated routing protocol in order to see the one that performed better.

2.2.8 Evaluating the Performance of DTN Routing Protocols

Many folds settings have been used to evaluate the performance of Delay Tolerant Network(DTN) routing protocols. They mostly include some kind of mobility models built from synthetic mobility model taken from traces of real-life models, some of which are (Keranen & Ott, 2009):

i. **Diesel Net**: This may be used to obtain real-world traces which can be made available to other researchers. Its flexibility is limited and large-scale operation is expensive. Diesel net is typically limited to those running test-bed.

ii. **CRAWDAD project**: This provides a repository where numerous real-world traces are available. These traces give insight into the real-world interaction between mobile users

from different angles and make a valuable data source for mobility and connectivity characteristics obtained from synthetic model. However, real-life traces have their own limitations since the population analyzed in the traces is limited, hence, the result is biased. Also the time granularity is small in order not to drain the mobile device battery quickly and results cannot be arbitrarily scaled-up thus, limiting what can be evaluated.

iii. **TRANSIMS** allows modeling a population and how they interact at certain locations or in public vehicles. It does not include details on the path between such locations thus, limiting its suitability for generating mobility data for pedestrians.

In summary, a number of test-beds, real-world traces, tools or simulation models address only a particular subset of users. Traces and test-beds are limited in their coverage and granularity. Vehicular mobility models do not consider pedestrians, and constriants on maps are often not taken into account. Also the models do not allow the incremental addition of further features to stepwise increase the degree of reality of the simulation. The Opportunistic Network Environment(ONE) simulator contributes an environment for DTN protocol evaluation, which has internal and external mobility models, different DTN routing schemes, interactive inspection as well as post-processing. ONE allows the incremental addition of further features to stepwise increase the degree of reality of the simulation.

2.2.9 The ONE Simulator

ONE is a customizable java-based simulator targeted for research in DTN. It can import mobility data from real world traces or other mobility generators. It can also produce a variety of reports from node movement to messages passing. The simulator is written in java (1.6), it allows a user

to simulate different scenario quickly and in a flexible manner. It also provides one easy way to generate statistics from the simulation performed. It can be run on Linux, Windows or any other platform supporting java. ONE is capable of (Keranen & Ott, 2009; Benamar et al., 2014):

i. Generating node movement using different models.

ii. Routing messages between nodes with various DTN routing algorithm.

iii. Visualizing both mobility and messages passing in real time in its Graphical User Interface (GUI). Figure 2.5 shows an overview of ONE simulator GUI.

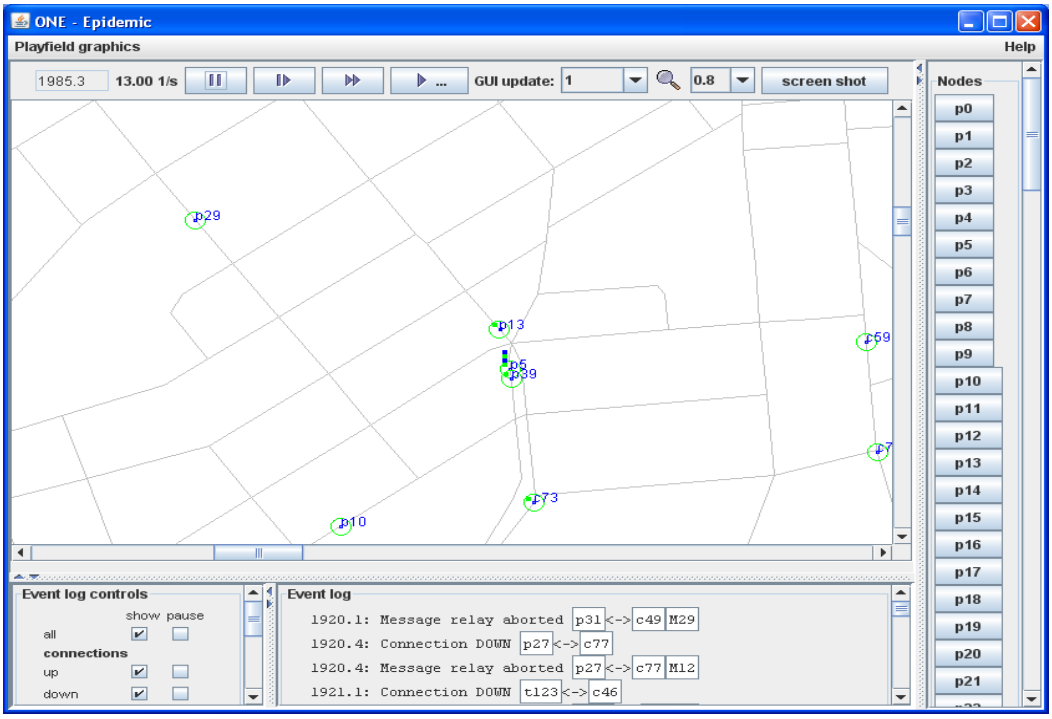


Figure 2.5: An Overview of the ONE Simulator GUI.

Several built-in routing modules are available in ONE (that is, direct forwarding, first contact, spray and wait, prophet, maxprop, epidemic), also external simulators such as Network Simulator-2 (NS2) can be used. For external programs, a separate parser is needed to convert

message events to a form the ONE understands. ONE is able to visualize results of the simulation in two ways (Keranen & Ott, 2009):

i. Showing the simulation in real time within the GUI with node locations, current paths, connections among nodes, number of messages carried by a node.
ii. Using a map-based movement model, showing all map paths in the GUI.

The whole simulation can be configured to pause when an event occurs and a single node can be selected for further information about the message a node is carrying and about routing modules state (Keranen & Ott, 2009).

2.3 Review of Similar Works

Considerable work has been done by many researchers to design an efficient routing protocol for opportunistic networks. Others evaluated the performance of the existing routing protocol. The key ones are critically reviewed as presented hereunder:

Lilien *et al.*, (2006)were the first to introduce a new paradigm and technology called "Opportunistic Network". They were the first to define and investigate opportunistic networks where they pointed out its possible challenges (identity spoofing, denial of service (DOS) attacks by malicious devices etc.). They also suggested Intrusion Detection as security measure since malicious devices or networks are likely to join opportunistic networks because of lack of initial authentication mechanism. However, they did not show the architecture of the opportunistic network nor did they propose a routing protocol for the network that would have aided the understanding of the network.

Pelusi *et al.*, (2006) surveyed the most interesting case studies related to opportunistic networking (wildlife monitoring, opportunistic networks for developing area and pocket

switched networks in the Haggle project) and discussed and organized a taxonomy for the main routing and forwarding approaches in these network. They also gave further possible scenarios to make opportunistic networks part of the Next-Generation Internet. They admitted that routing and forwarding are the most compelling issues in opportunistic network but they did not design any protocol for such networks.

Chen *et al.*, (2006)presented a hybrid scheme called "H-EC" which combined the strength of erasure coding and the advantages of aggressive forwarding. Their simulation evaluated the performance of H-EC, EC and R-EC schemes in terms of delivery ratio and latency. They showed that H-EC offered robustness in worst-case delays scenarios. However, they did not compare their scheme with well-known routing algorithms like Prophet, HIBOP, Spray and wait, etc.

Amin *et al.*, (2007) presented a practical network organization scheme for mobile WiMAX mesh networks. The solutions exploited the connected nature of wireless mesh networks to minimize communication interruption during hand-over. The solutions were assessed using the NS-2 simulator. However, the model made simplified assumptions of the mesh link as being error-free and full-duplex. It also assumed that all mobile stations had uniform bandwidth demands and mobility pattern. However, these assumptions can affect accuracy of their model.

Lin *et al.*, (2008) introduced CodeOR an opportunistic routing technique in a wireless mesh network with segmented network coding. CodeOR allowed transmission of multiple segments to fully utilize network resources. It is especially appropriate for real time multimedia applications through the use of small segment sizes to decrease decoding delays. They showed (using plots) that their technique significantly outperformed existing approaches in network throughput while

maintaining a similar amount of data transmission. However, their work was based on the assumptions that all nodes used a constant transmission rate which is not the case in reality. These assumptions must have affected the efficacy of their model.

Keranen et al., (2009) presented the opportunistic networking environment (ONE) simulator designed for DTN routing and application protocol. The ONE simulator is a customizable Java program that allows users create scenarios based upon different synthetic movement models and real-world traces. They created an extensive set of ready-to-use modules (six synthetic mobility models that can be parameterized and combined to approximate real-world mobility pattern). Using the framework they implemented six configurable well-known opportunistic networks routing scheme (direct delivery, first contact, spray –and-wait, prophet, max-prop and epidemic) and showed simple simulation to demonstrate the simulator's flexible support for DTN protocol evaluation. However, the message generation activity in ONE did not take into account group relationship and context information which would have provided better modeling for buildings and traffic.

Islam & Waldvogel,(2011) proposed a simulation technique (called "EPO1") which exposed the shortcomings of flooding as a benchmark for opportunistic networks. EPO1 used the Dijkstra algorithm to find the shortest path. They showed that their method did not suffer the bottlenecks that limited the performance of 'epidemic' routing even when bandwidth is limited and that it delivered accurate information about the network characteristics (such as contact frequency, contact duration, number of paths, etc.). They did not however, consider aspects like commutative throughput on the diameter of the network and faults in clusters which would have played an important role in improving their EPO1.

Moreira *et al.*, (2011)developed a model to help designers carry out fair comparisons among protocols. Their simulation model compared the performance of standard approaches like epidemic, flooding and bubble rap, showing how they performed under the same conditions, metrics and scenario. The research showed the usage of different parameter setups and performance metrics could result in assessment that is biased to some proposal. Hence, a designer can classify their own proposal as well as assessing them. The research however, did not consider more routing protocols including the spray and wait, the integrated routing protocol etc. to make their work more robust.

Trivino-Cabrera & Canadas-Hurtada, (2011)surveyed the opportunistic routing in multihop wirelessnetworks. They classified the opportunistic routing protocol into five groups based on the metric used and gave a summary of the main features of opportunistic routing protocol. However, they did not consider the presence of obstacles in the geographical information based routing protocol which is likely to create a restriction on the mobility of nodes and consequently, affecting the routing performance.

Basalamah *et al.*, (2012)developed a new link-correlation-aware opportunistic scheme which improved the performance of opportunistic routing by selecting algorithm that helped to diversify opportunistic links. They analyzed the opportunistic routing gain under the presence of link correlation considering the loss of DATA and ACK packets. However, they did not extend their analysis to multiple forwarders and they assumed that the ACK packet size is always received which must have affected the efficacy of their model.

Zhao *et al.*, (2012)performed a systematic performance analysis of two opportunistic routing protocols (EXOR AND MORE) which were implemented in the OMNeT++ simulator. Their

analysis justified the scenarios where opportunistic routing performed better than traditional MANET routing. However, they used MORE which is a protocol based on random network coding (Random network coding algorithms have high computational complexities in packet processing and cannot support high data rates).

Verma & Srivastava,(2012)designed an integrated routing protocol which used context data when available and fell back to dissemination-based routing when context information was not available. They compared the epidemic and prophet routing techniques with the integrated routing and showed that the integrated routing provided better results in terms of message delivery probability, and message delay whether context information about users was available or not. They did not consider other parameters such as message delay, congestion, resources consumption that could improve the performance of the integrated routing protocol.

Tsapeli & Tsaoussidis, (2012) presented a novel routing algorithm for the opportunistic networks which combines the probabilistic routing with erasure coding. They allocated code blocks to nodes that relied on a probabilistic metric for evaluating nodes potential in order to enhance the robustness of the erasure coding. They showed that their algorithm outperformed the simple erasure coding scheme both for small and worst delay performance cases. However, they did not vary the threshold in order to get it appropriate value, and this must have affected the efficacy of their model.

Dinakar *et al.*, (2013) compared Mobile Ad-hoc Networks (MANETS) and Opportunistic Networks where link failure occurs frequently and the challenges in implementing them. They showed that opportunistic networks can be utilized in MANETS to overcome the frequent link

failures. However, their studies were analytical only, because they did not simulate any routing technique for opportunistic networks.

Pan et al., (2013) proposed a comprehensive integrated buffer management strategy (CIM) which was evaluated using ONE simulator. Unlike the existing studies, their research did not focus only on a subdivision of buffer management. They integrated different types of buffer management methods (queuing strategy, cache replacement, redundant removal) and took into consideration all information relevant to the message delivery and network resources. They compared their model with the first-in-first-out (FIFO), prophet and the spray-and-wait and showed that their CIM improved the performance in terms of delivery ratio, overhead ratio, and average delivery delay. However the spray-and-wait had higher delivery ratio and lower overhead than their model showing that there is still room for improving the buffer management strategy.

Asgari *et al.*, (2013) presented a novel approach for intelligent routing and transmitting messages from source to destination based on distributed learning algorithm. They used NS2 simulation software and compared the results with HIBOP and the epidermal routing algorithm, where they showed that their approach is more efficient (since it generated lower traffic) than previously presented approaches (HIBOP and Epidemic). However, epidemic routing is limited in its usefulness as a benchmark for opportunistic networks since it contains a lot of shortcomings (overhead generation, resources wastage, congestion).

Orozco *et al., (2013)*presented an analytical model of opportunistic networks which was used to understand network behavior in different scenarios. Their model used Markov Chain theory to

describe how messages propagate. Two routing strategies (Epidemic and spray & wait) were formalized in terms of the number of copies and maximum hops allowed for a message. Three variables (delivery time, mean time to arrival of a message and the expected number of the message copies at delivery time) were used to evaluate the performance of this routing algorithm. Results obtained showed that the epidermal routing technique was a better option (it had shorter delay in transmission) but it consumed memory while the spray and wait is preferable in applications that needed to save energy to prolong the life of a battery and for nodes that did not have large memory. However, their work was limited to only epidemic and spray and wait algorithms, and their model was based on the assumption that nodes have infinite energy and memory which is not the case in reality.

Hu *et al.*, (2013) developed a practical opportunistic routing (POR) technique to address the high speed and multi-rate wireless network requirements for opportunistic network. They also performed a novel path cost calculation which enabled good path selection by considering the ability of nodes to select appropriate data rates to match the channel condition. They performed their experiment on a 16-node test bed and showed that their POR achieved significantly better performance than the compared protocols (MORE, ExOR, MIXIT and Crelay). They however, did not extend their POR to exploit features in IEEE 802.11n and IEEE 802.11ac networks which would have made their work more widespread.

Martin-Campillo *et al.*, (2013)compared the efficiency of some routing protocol in the opportunistic networks (epidemic, prophet, maxprop, time-to-return (ttr)) using simulation in realistic disaster scenarios with different characteristics (that is, using different number of nodes,

different number of messages and different message sizes) in order to see their performance and suitability to different scenarios. They used ONE simulator and showed that maxprop performed better in all scenarios. They also showed that ttr had a low overhead and cost though it had a low performance. However, disaster situations are not likely to have persistent Internet connectivity as assumed in the study and this must have affected the efficacy of their model.

Oliveira *et al.*, (2014) presented a more realistic evaluation of opportunistic network contacts by evaluating the performance of several queuing and forwarding strategies over different routing protocols and showed that the behavior of the network changed significantly from what has been reported in previous studies, due to effect of collision, channel contention and transmission error. They used the NS2 simulator and concluded that buffer management and forwarding prioritization can influence the performance of the network by up to 30%. However, they did not consider fading, newer communication standards such as IEEE 802.11p and mobility models which can affect the efficacy of their model.

Yogi & Chinthala, (2014)discussed the differences between the opportunistic network and the mobile peer-to-peer network, and the differences between the opportunistic network and the mobile ad-hoc network. They classified the opportunistic networks' application as active collaboration (AC) and passive collaboration (PC). They also discussed privacy issues in the opportunistic network but they did not state how to improve privacy in the opportunistic networks which would have made their studies more useful.

Vidya & Hemanth, (2014)surveyed the routing and congestion control strategies in opportunistic networks and classified the routing protocols into forwarding-based and flooding-based approach. They presented the taxonomy of the communication networks and discussed

some of the challenges in opportunistic networks. They also discussed the congestion control strategies in opportunistic networks. However, their studies is analytical and their conclusion has already been presented in the work of Verma and Srivastava.

Silva *et al.*, (2015) surveyed the processes of DTN congestion control and organised a taxonomy that helped in mapping and comparing the existing DTN congestion control mechanism. They organised a flow and congestion control mechanism which showed how congestion is detected in each mechanism, its evaluation platform, routing type, contact type, and application of each mechanism. They deduced that "there is no universal congestion control mechanism that will be applicable to all DTN scenarios and applications". However, with this quantitative and qualitative studies, they were not able to propose a novel congestion control mechanism that can be applied to a number of applications.

Based on the aforementioned studies and shortcomings, there is still need to design or improve on the available routing protocols in opportunistic networks as congestion is inherent in opportunistic network and it reduces routing performance. How to fully utilize the characteristics of opportunistic network to design a buffer management strategy is still an open issue.Also opportunistic networks are prone to malicious attack. This work focuses on improving the integrated routing protocol by controlling congestion. This is done by pre-emptive eviction of data items from the buffers of nodes

CHAPTER THREE

MATERIALSAND METHODS

3.1 Introduction

This chapter describes the detailed procedure carried out in modeling the opportunistic network using the epidemic, the prophet and the integrated routing protocol in the ONE simulator. The processes of modeling and simulation of the improved integrated routing with congestion control is also presented in this chapter.

3.2 Model Test Case

The model test case is an opportunistic network environment consisting of ten (10) nodes. Communication was assumed to be between modern mobile phones and similar devices having 10MB of free RAM for buffering messages. The area used covered 450 x 340m^2. Nodes were taken as pedestrians moving at random speeds of 0.5 to 1.5m/s. Bluetooth transmission speed of 2 Mbit/s and a range of 10m was used. A high-speed transmission speed of 10Mbit/s and 1000m range was considered. The message size ranges between 500kB to 1MB. The event generation range is between 25-35seconds and a time-to-life of 5hours was assigned. An initial probability of 0.75 was used in the prophet and integrated routing protocol delivery probability. Also, the model settled at values of beta β and gamma γ as 0.25 and 0.98, respectively in calculating the delivery probability using equations (2.1) to (2.3) of section 2.2.4.4.

3.3 Installation and Configuration

A compressed source code of THE ONE (version "ONE_1.5.1.tar.gz") was downloaded from http://www.netlab.tkk.fi/tutkimus/dtn/theone/down/one_1.5.1.tar.gz, the source code was then extracted and compiled into Java eclipse Integrated Development Environment (IDE).

The configuration was carried out using 'Eclipse IDE for Java Developers which was downloaded from http://www.eclipse.org/downloads/ to debug and run the ONE project. Another option is the 'Netbeans' which can be downloaded from https://netbeans.org/community/releases/73/index.html. The reason for using eclipse is that it is more user friendly. Also,Netbeans consumes too much computer resources and the running speed of it is slower than the Eclipse (Liu, 2013).

The downloadedeclipse IDE was uncompressed and packaged to a folder to run the eclipse application. Two important jars were imported for the ONE to compile, that is, 'DTNConsoleConnected.jar' and ECLA.jar. 'DTNConsoleConnector' jar offers a way for DTN simulator to use command to communicate with DTN simulated scenarios, that is, information could be managed and obtained from console instead of only gaining information from output files. As for ECLA.jar file, it is used to simulate network interface hardware and it tries to finish the work in the link layer.

3.4 Simulation

The opportunistic network was simulated in the ONE simulator as a dynamic set of nodes. Nodes are allowed to move randomly.The simulation was carried out using 10 nodes in order to check the workability of the model. The epidemic, prophet, and integrated routing protocol were

simulated using the same scenario as modeled in section 3.2. The simulation of each protocol is described as:

(a) Epidemic Routing Protocol.

The sub-program for the epidemic class is shown below:

```
package routing;
import core.Settings;
//Epidemic message router
public class EpidemicRouter extends ActiveRouter {
	public EpidemicRouter(Settings s) {
		super(s);
		//TODO: read&use epidemic router specific settings (if any)
	}
	protected EpidemicRouter(EpidemicRouter r) {
		super(r);
	//TODO: copy epidemic settings here (if any)
	}
```

The epidemic router was called from the setting in Appendix A2 as:

Group.router = EpidemicRouter

At the beginning, nodes have no messages in their buffer as shown in Figure 3.1. After some time, node's buffer builds up as seen in Figure 3.2. The node's buffers continue building up as messages are flooded in the network, up to an extent of becoming overwhelmed with messages. At this moment, the nodes are exhausted thereby causing serious congestion. The presence of red colour on the nodes implies that there is congestion in those nodes. This is shown in Figure 3.3. The program class and routing settings for the epidemic routing are shown in Appendices A1 and

A2 respectively. The result obtained from simulating the epidemic routing protocol is shown in Table 4.1

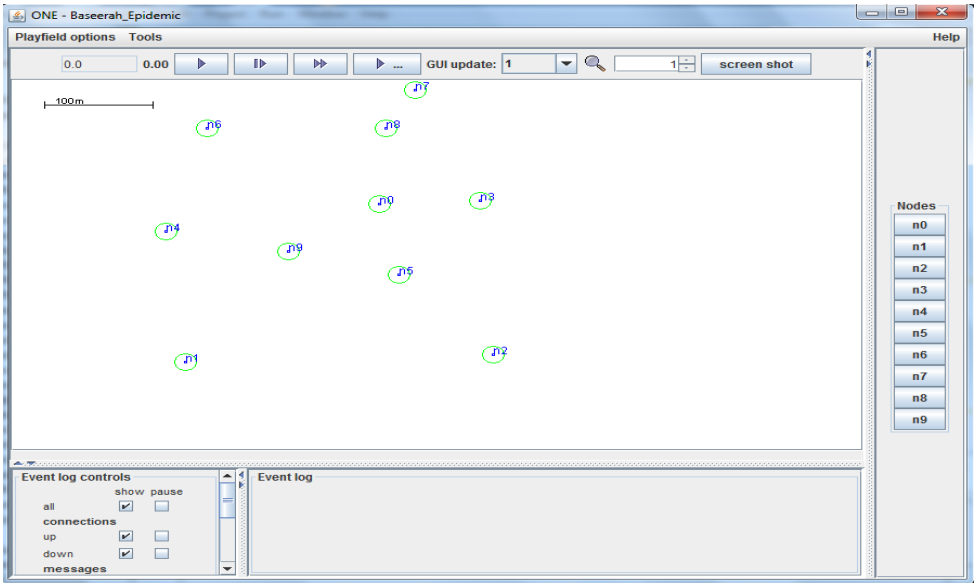


Figure 3.1: Beginning the Epidemic Simulation.

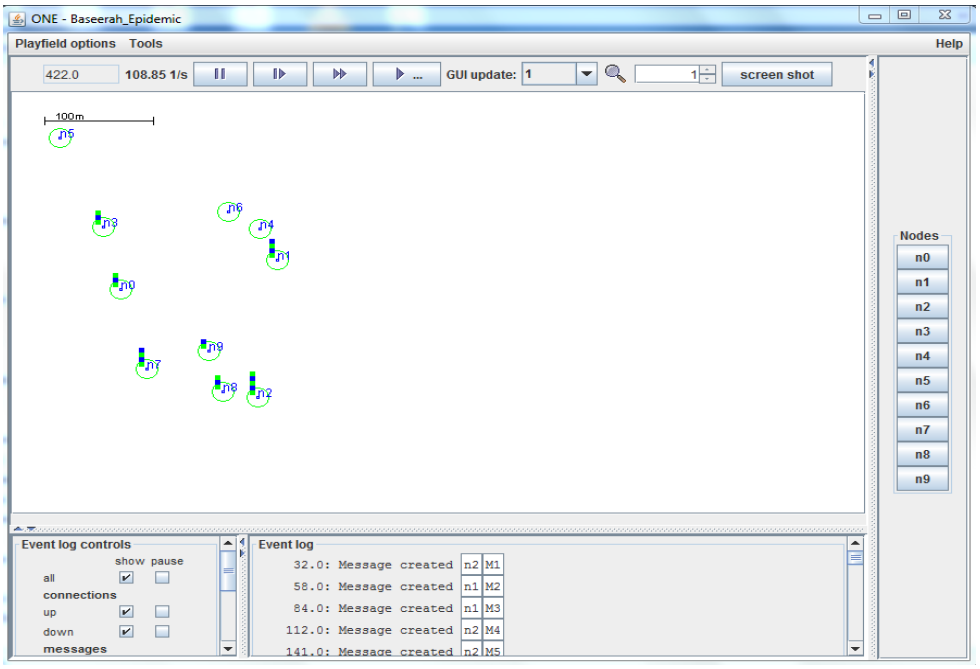


Figure 3.2: Nodes Build-up Process in the Epidemic Simulation

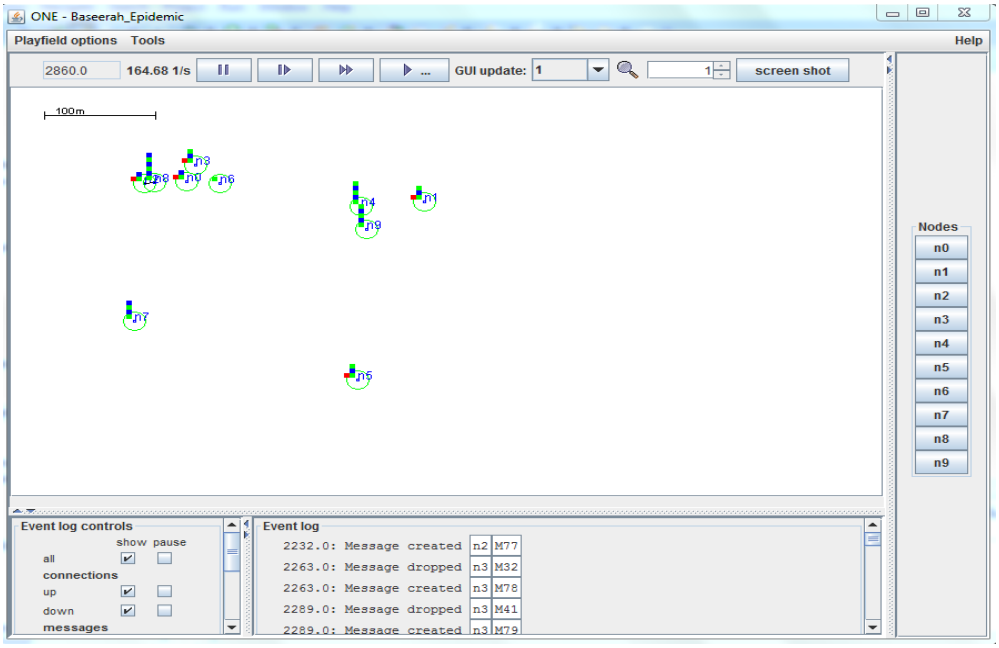


Figure 3.3: Congestion in the Epidemic Simulation.

(b) The Prophet Routing Protocol

The Sub-Program for the prophet routing is shown as:

```
public class ProphetRouter extends ActiveRouter {
                public static final double P_INIT = 0.75;
                public static final double DEFAULT_BETA = 0.25;
                public static final double GAMMA = 0.98;
                public static final String PROPHET_NS = "ProphetRouter";
        public static final String SECONDS_IN_UNIT_S ="secondsInTimeUnit";
                public static final String BETA_S = "beta";
        private int secondsInTimeUnit;
        private double beta;
        private Map<DTNHost, Double> preds;
        private double lastAgeUpdate;
public ProphetRouter(Settings s) {
```

```
        super(s);
        Settings prophetSettings = new Settings(PROPHET_NS);
        secondsInTimeUnit = prophetSettings.getInt(SECONDS_IN_UNIT_S);
        if (prophetSettings.contains(BETA_S)) {
                beta = prophetSettings.getDouble(BETA_S);
        }
        else {
                beta = DEFAULT_BETA;
        }

        initPreds();
}
```

The prophet router was called from the settings in Appendix B2 as:

Group.router = ProphetRouter

At the beginning of the simulation, node's buffers are free as the case is with the epidemic routing. After some time the buffers begin to build, when node 3 met node 2 as shown in Figure 3.4, they exchanged their delivery predictability information and then each node updates its own metric. Node 3 forwards a message to node 2 since node 2 has a higher delivery probability. When nodes with higher delivery probability are not encountered in reasonable time, buffers keep their messages and as new messages are formed, the buffer space gets exhausted. This congests the network as shown in Figure 3.5.

Appendices B1 and B2 show the program class and routing settings for the prophet routing protocol. The result obtained from simulating the prophet routing protocol is shown in Table 4.1

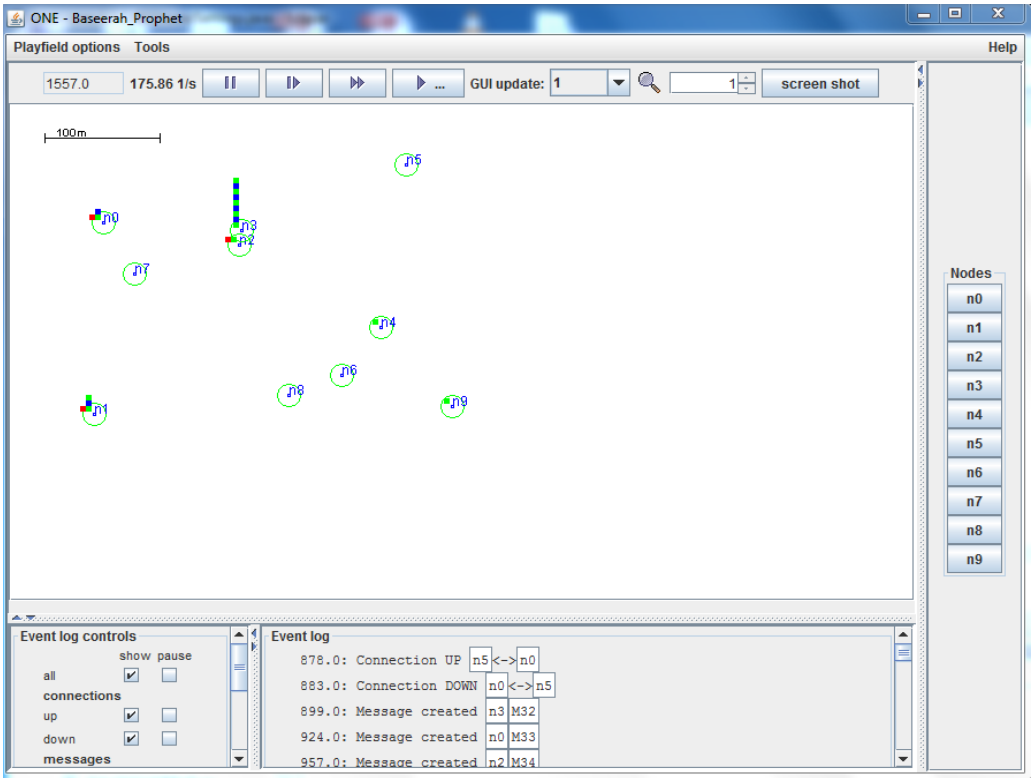


Figure 3.4: Communication between Nodes in Prophet Routing Protocol.

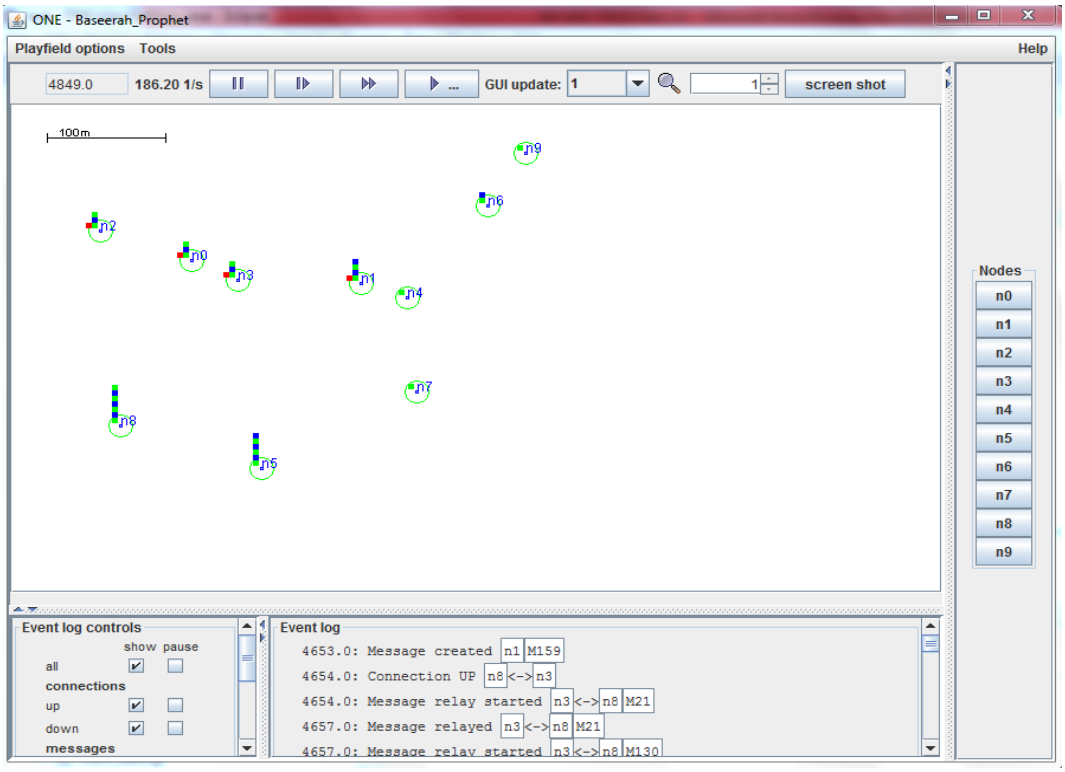


Figure 3.5: Congestion in Prophet Routing Protocol.

(c) The Integrated Routing

The scenario settings for the integrated routing is shown as:

```
Scenario.name = Baseerah_Integrated
Scenario.simulateConnections = true
Scenario.updateInterval = 1.0
Scenario.endTime = 43200
Scenario.nrofHostGroups = 1
btInterface.type = SimpleBroadcastInterface
btInterface.transmitSpeed = 250k
btInterface.transmitRange = 10
highspeedInterface.type = SimpleBroadcastInterface
highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 1000
Group.movementModel = RandomWaypoint
Group.router = [ProphetRouter;EpidemicRouter;] #[Integrated]
ProphetRouter.secondsInTimeUnit = 1
Group.bufferSize = 10M
Group.nrofInterfaces = 1
Group.interface1 = btInterface
```

The routing setting for the integrated routing protocol is shown in Appendix C1. The program class was developed by combining the class of the epidemic routing and the prophet routing (Appendices A1 and B1).

When nodes 0, 2 and 6 come into communication range as shown in Figure 3.6, they exchange their contact probability and time attribute. Nodes update their matrices if the other node has a more recent update time attribute. The result obtained from simulating the integrated routing protocol is shown in Table 4.1

Despite improving the performance of routing by hybridizing the prophet and epidemic routing protocols, there is still congestion in the network as depicted by nodes with red colours in Figure 3.6. Hence, congestion control needs to be incorporated into this integrated routing protocol.

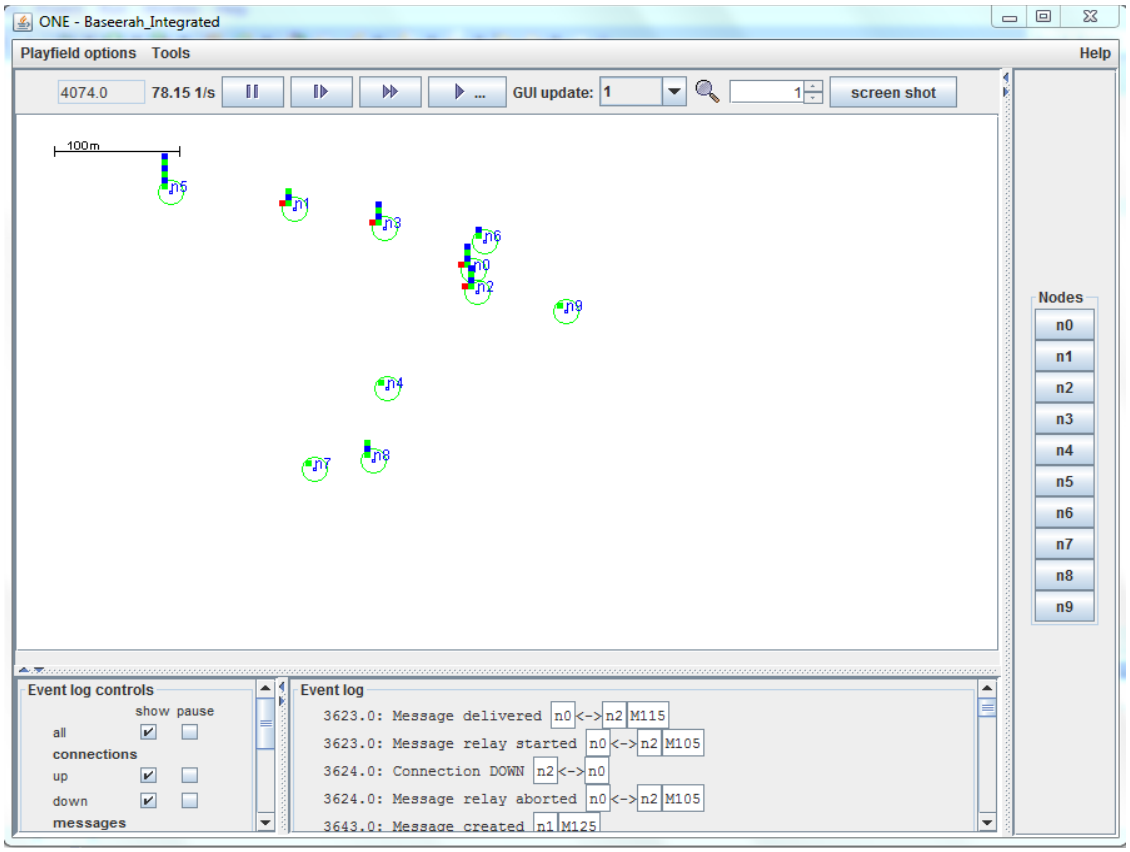


Figure 3.6: Simulating the Integrated Routing Protocol.

A research carried outby Moreira, et al., (2011) showed that comparing the performance of routing protocol using different conditions, parameters, metrics, and scenarios could result to an assessment that is biased (Moreira, et al., 2011). This is because the performance metrics of a given scenario may favour a given model.

The same simulation scenario as that of the conventional integrated routing protocol was then used to simulate the integrated routing protocol. (researches in opportunistic networks use the Helsinki region to validate results due to the fact that the opportunistic network environment simulator originated from the region, an example of this is a research carried out by Verma and Srivastava, (2012), Keranen and Ott, (2007)). A part of the Helsinki downtown area as depicted in Figure 3.7 was used (4500 X 3400m). Communication between modern mobile phones and similar devices (for example, personal digital aid) was assumed. Devices have up to 20MB of free RAM for buffering messages. Users travel on foot, in cars or trams. Some paths to the parks, shopping malls and trams route have been added. Simulation was run using 100 nodes. Mobile nodes have different speeds and pause times. Pedestrians move at random speeds of 0.5 to 1.5m/s, taking a pause time between 0 to 120s. Cars are optional, when present, they make 20% of node count, moving at a speed of 10 to 50km/h, with a pause time of 0 to 120s. Trams run at a speed of 7 to 10m/s, pausing for 10 to 30s at each configured stop. An assumption of 10m range, 2Mbits bluetooth and a low power use of 802.11bWLAN (30m range, 4.5Mbits). Mobile users generate messages on an average of once per hour per node. Messages lifetime of 3, 6 and 12hours were used. The message size used was uniformly distributed between 100kb (test message) and 2MB (digital photo). Figure 3.8 shows the integrated routing carried out on Helsinki map where nodes follow a predefined path and communicates in a transitive manner. Appendix D1 shows the simulation settings. The result obtained from simulating the integrated routing protocol on Helsinki is shown in Table 4.2

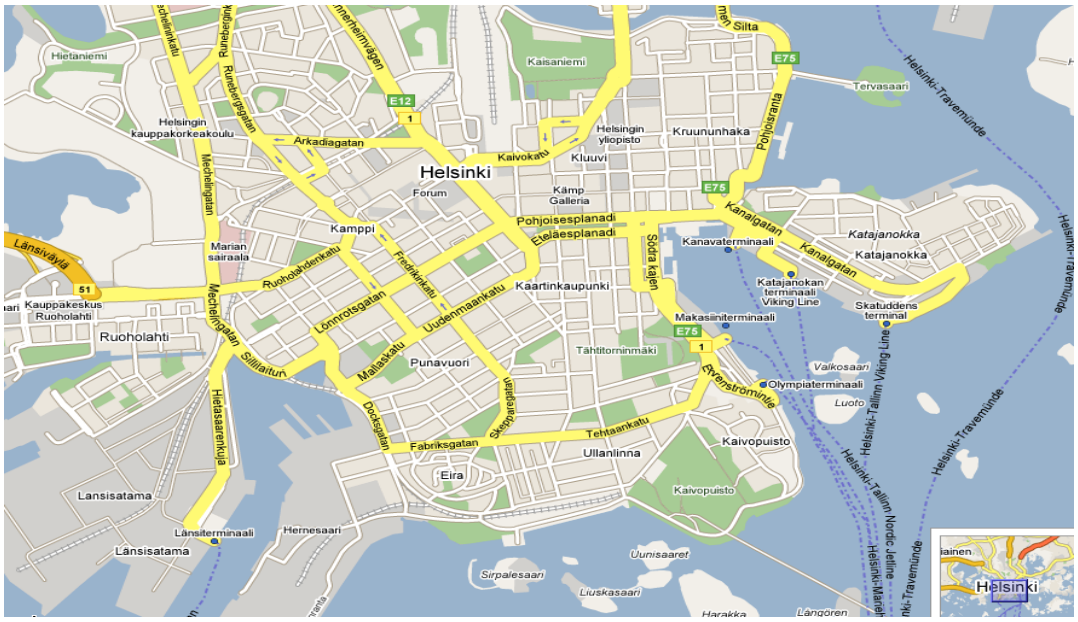


Figure 3.7: Helsinki Simulation Area.

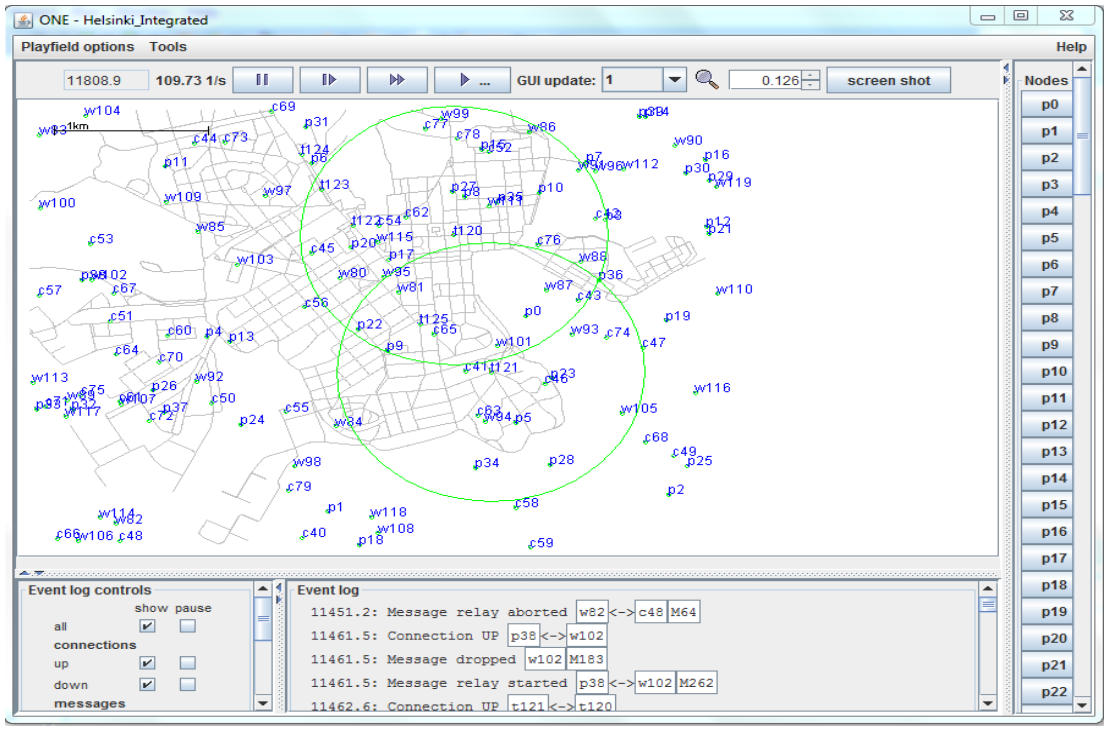


Figure 3.8: An Overview of the Helsinki Integrated Routing Simulation.

The general flow chart for the simulation is shown in Figure 3.9.

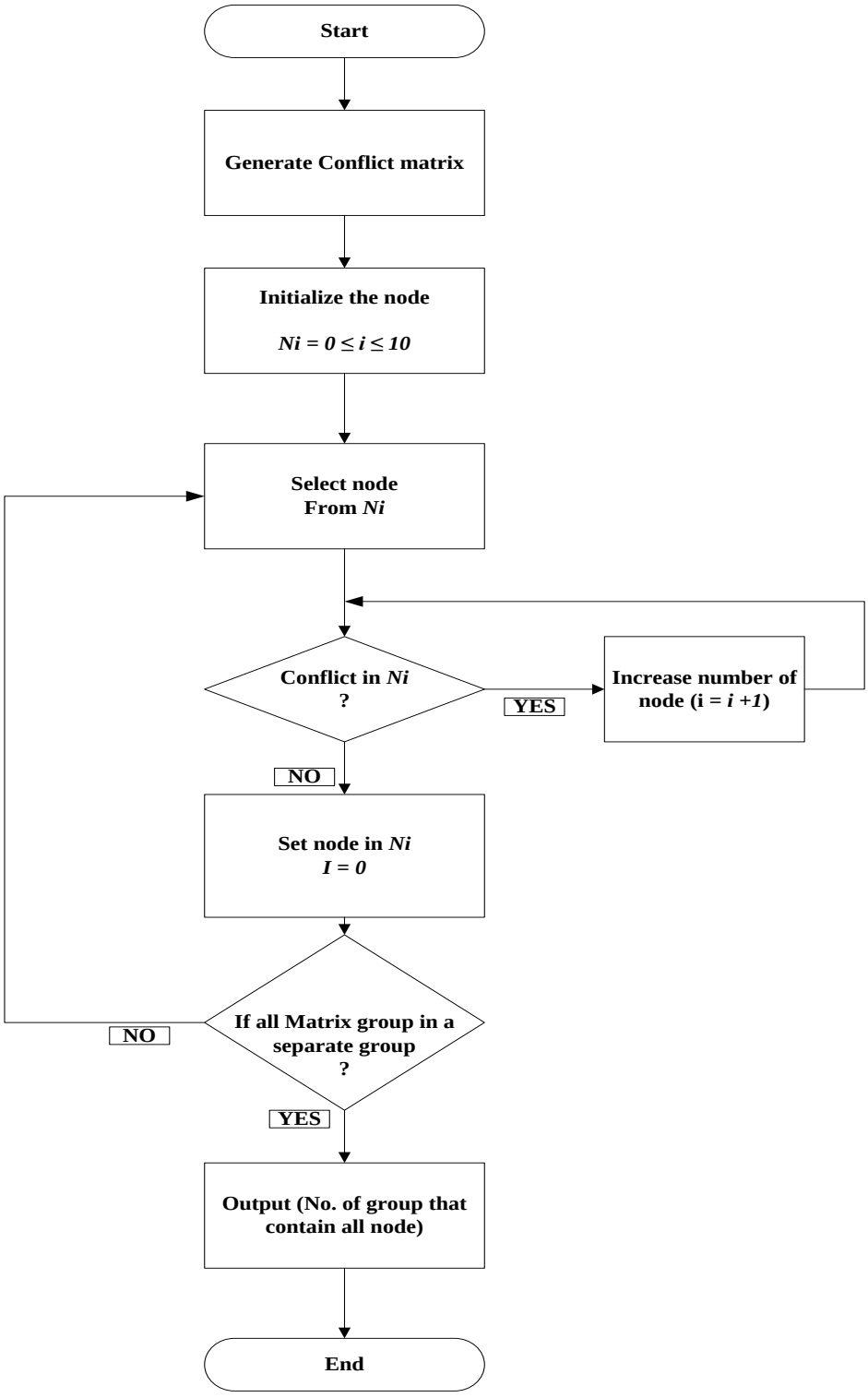


Figure 3.9: The Flow Chart of the Simulation Process

3.5 Visualization

The simulation is visualized in real-time within the GUI shown in Figures 3.1to 3.6. Node's location, number of messages carried by nodes, connection among nodes, and current path are visualized in the main window. The GUI also produces a log of events in the simulation which can be filtered based on interest as shown in Figures 3.1 to 3.6. A node can be selected for further inspection by selecting it from the list, thereby making it possible to retrieve more information about the message the node is carrying and about the routing module's state. The whole simulation can be configured to pause when an event of interest occurs. The view can be zoomed and the speed of simulation canbe adjusted interactively. Red colour on a node represents a congested node.

3.6 Simulation of the Improved Integrated Routing Protocol with Congestion Control.

Four congestion control strategies (use of acknowledgement, buffer size advertisement, data centric method and the avoidance of duplication) were developed and programed into the integrated routing protocol.Each strategy was used to model the opportunistic network in the ONE simulator.Appendix E shows the program class for the congestion control process.

3.6.1 Buffer eviction using acknowledgement

The use of acknowledgementas the means of congestion control was called from the scenario setting of the ONE using the following Code:

Scenario.name = Baseerah_Acknowledgement

This code prompts the graphical user interface of the ONE simulator in Figure 3.10. Figure 3.10 shows the simulation process of using Acknowledgement as the congestion control strategy in

the ONE simulator. hence, the images on the screen are nodes moving randomly. It can be seen that some nodes have no message in their buffer (nodes 5, 7 and 8), while the buffer of other nodes have built up messages. The results obtained from this simulation are shown in Table 4.4.

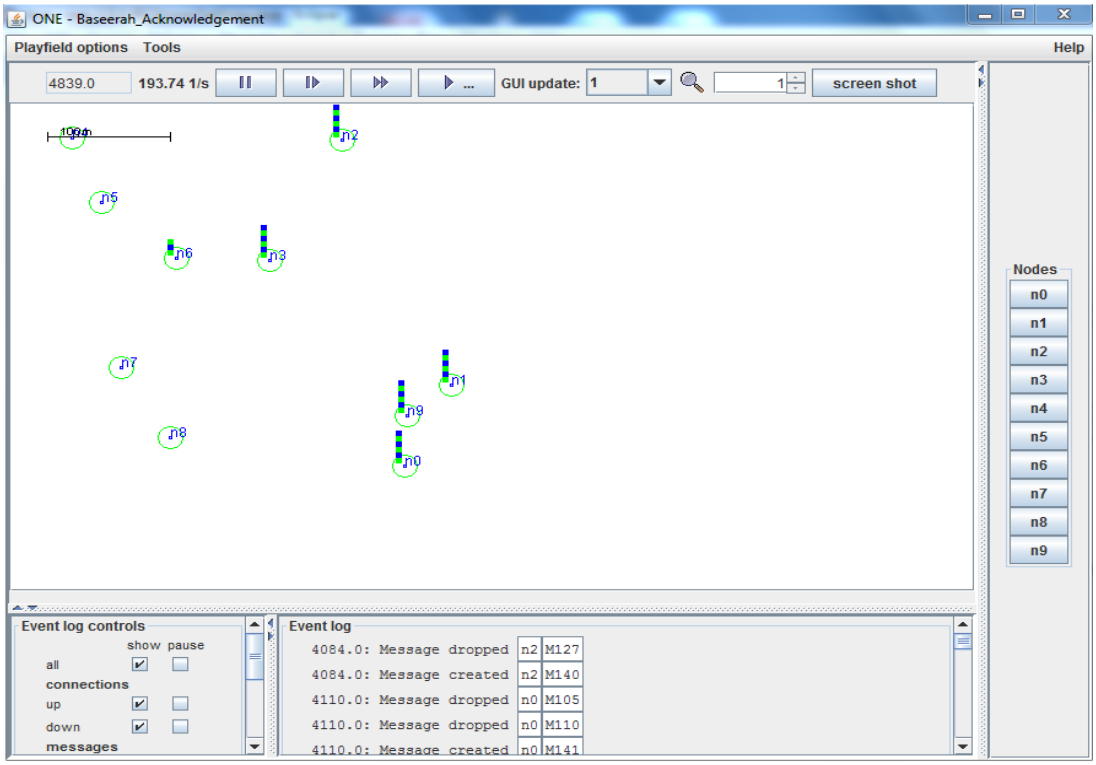


Figure 3.10: Use of Acknowledgement for the 10-Node Model.

3.6 2 Buffer size advertisement

The use of buffer size advertisement was called from the scenario setting of the ONE using the following Code:

Scenario.name = Baseerah_Advertisement

This code prompts the graphical user interface of the ONE simulator as shown in Figure 3.11.The simulation process of using advertisement as the congestion control strategy in the ONE simulator is shown in Figure 3.11. The event log on Figure 3.11 shows what is happening

at the nodes during the simulation process at any point in time.The results obtained from this simulation are shown in Table 4.4.

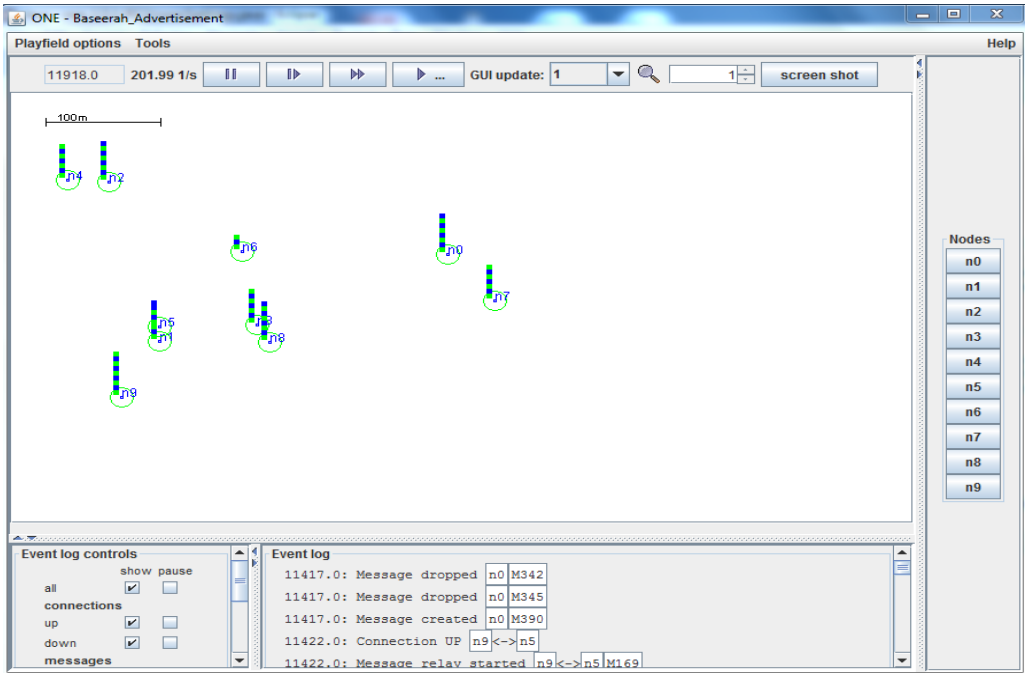


Figure 3.11: Use of Advertisement for the 10-Node Model

3.6.3 Data-centric node congestion avoidance

The use of data-centric node congestion avoidance was called from the scenario setting of the ONEusing the following Code:

Scenario.name = Baseerah_DataCentric

This code prompts the graphical user interface of the ONE simulator as shown in Figure 3.12 which is the simulation process of using data-centric node congestion avoidanceas the congestion control strategy. The event log gives an update what is happening at nodes at any time instant and the images seen on the screen are nodes moving randomly.The results obtained from this simulation are shown in Table 4.4.

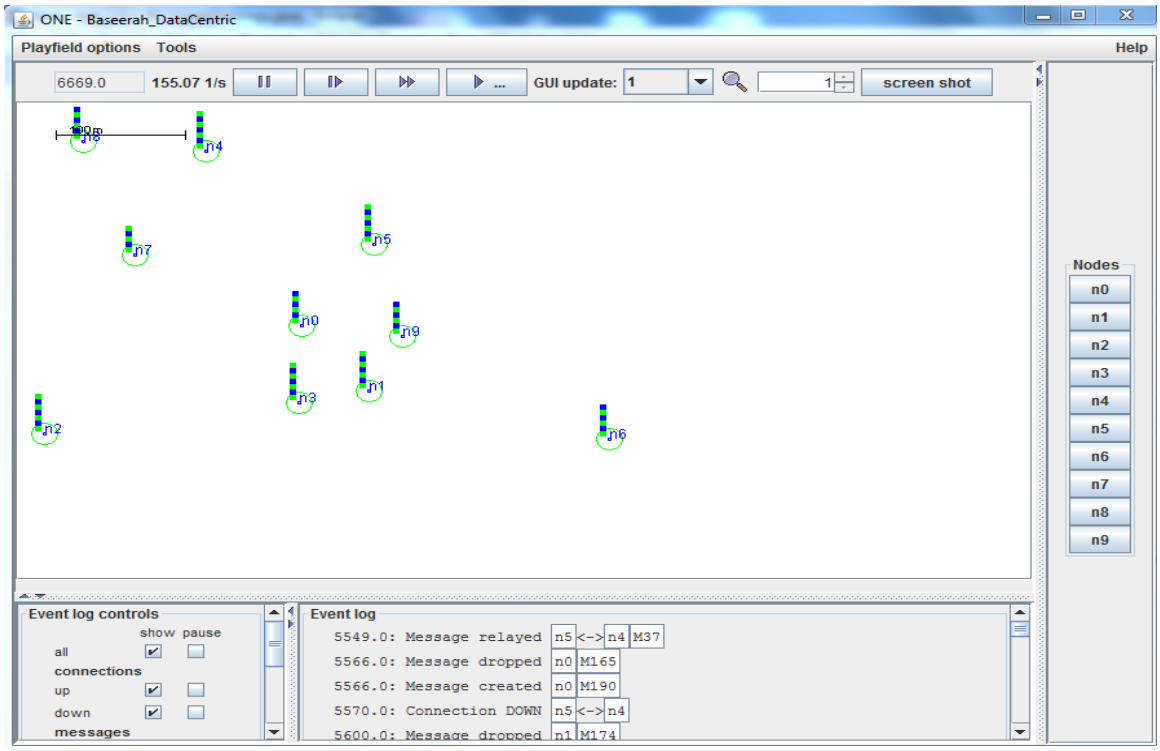


Figure 3.12: Data Centric Node Congestion Avoidance for the 10-Node Model

3.6.4 Duplication avoidance

The use of duplication avoidance was called from the scenario setting of the ONE using the following Code:

Scenario.name = Baseerah_DuplicationAvoidance

This code prompts the graphical user interface of the ONE simulator as shown in Figure 3.13 which is the simulation process of using duplicationavoidanceas the congestion control strategy in the ONE simulator. Ten nodes (n0 to n9) were used as depicted in Figure 3.13 and the event log gives an updateof what happens at nodes at any given time. The results obtained from this simulation are shown in Table 4.4

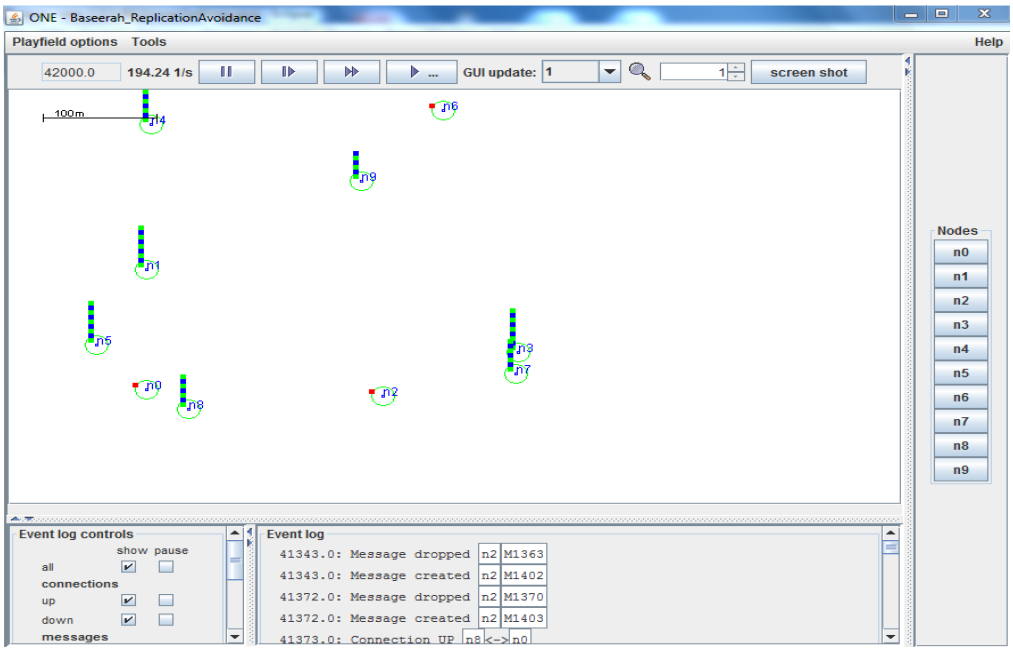


Figure 3.13: Duplication Avoidance for the 10-Node Model

From Figures 3.10 to 3.13, it can be seen thatthe red colour is reduced, and it appeared towards the end of the simulation time. This implies thatthe congestion control strategies reduced congestion and increased the span of the simulation time before congestion occurs. Nodes' buffers build to an appreciable value before congestion occurs.

3.7 Validation of the congestion control strategies

The same simulation scenario (Helsinki Region) as that of the conventional integrated routing protocol was used to validate the improved integrated routing protocol.The four congestion control strategies were used in the validation and Figures 3.14, 3.15, 3.16 and 3.17 show an overview of the simulation process in the ONE simulator.

Figure 3.14 shows how acknowledgement is used in the integrated routing protocol to reduce congestion, such that nodes delete copies of messages from their buffer once they receive an

acknowledgement that the message has reached its destination.The results obtained from this simulation are shown in Table 4.6.

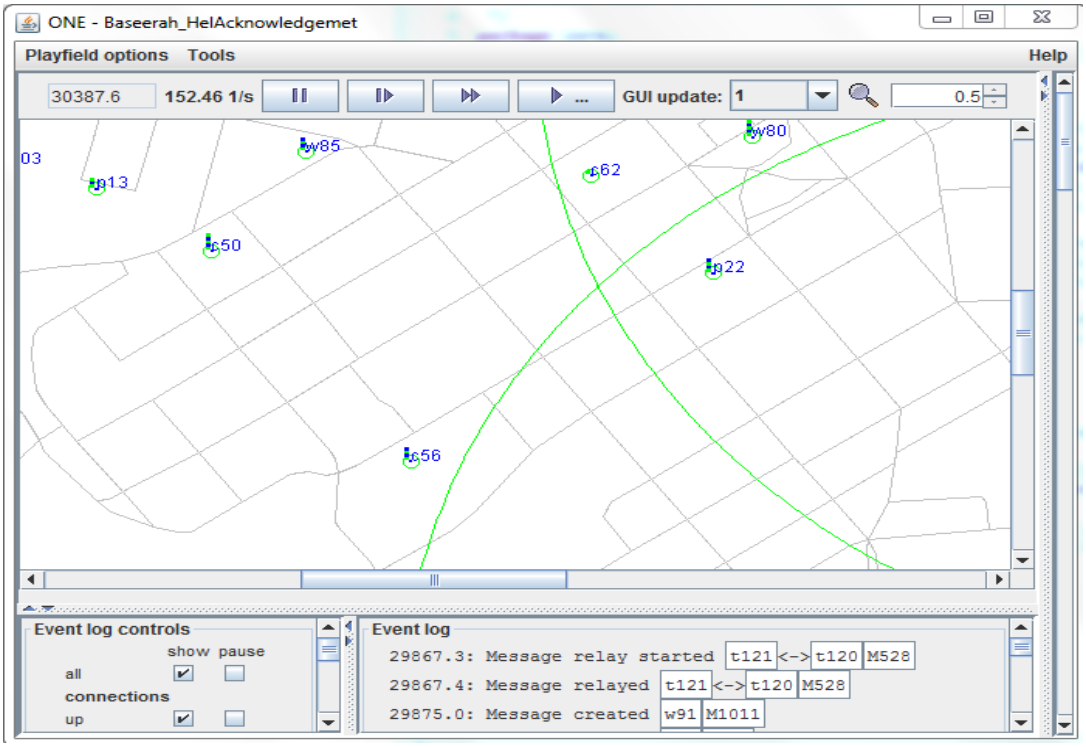


Figure 3.14: Use of Acknowledgement on Helsinki Simulation Region

In Figure 3.15, nodes advertise their buffer size so that a sending node does not send what the receiving node cannot carry in order not to overload the receiving node. The snap shot was taken at 10033.6s of the simulation time as represented in Figure 3.15.The results obtained from this simulation are shown in Table 4.6.

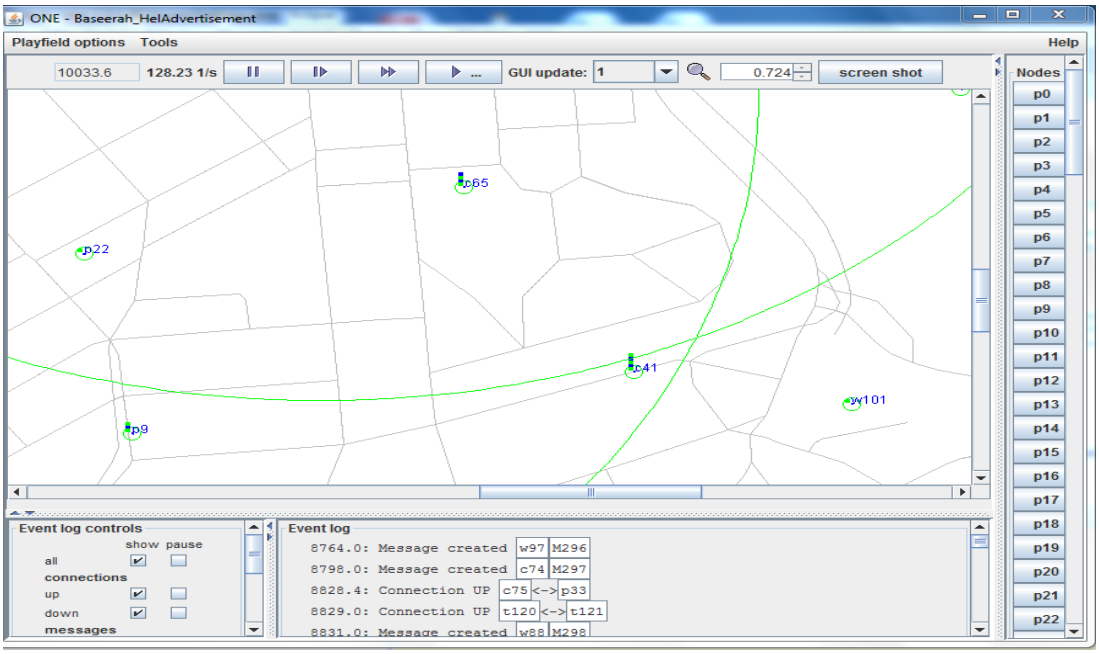


Figure 3.15: Use of Buffer Size Advertisement on Helsinki Simulation Region.

Figure 3.16 shows the data centric node congestion avoidance. Here, nodes receive only messages they are interested in, or messages they think it will have high demand from other nodes.

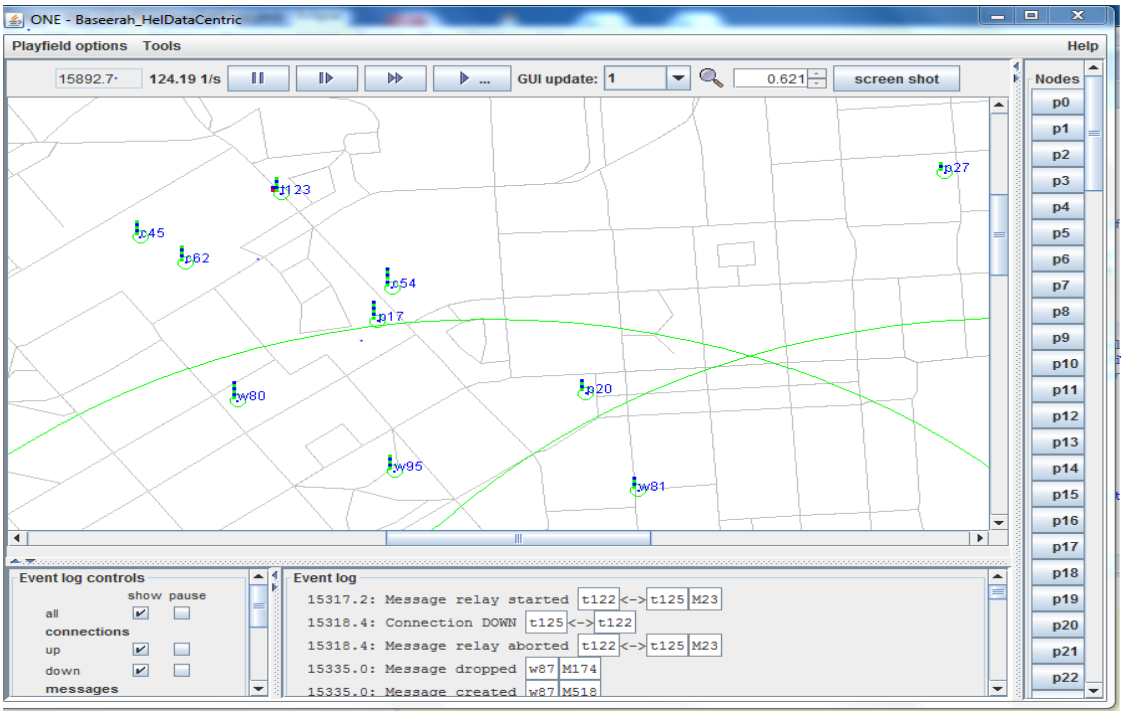


Figure 3.16: Data Centric Node Congestion Avoidance on Helsinki Simulation Region.

The snap shot was taken at 1589.7s of the simulation time as depicted in Figure 3.16. The event log gives an update of what is happening at the nodes in any given time and the results obtained from this simulation are shown in Table 4.6

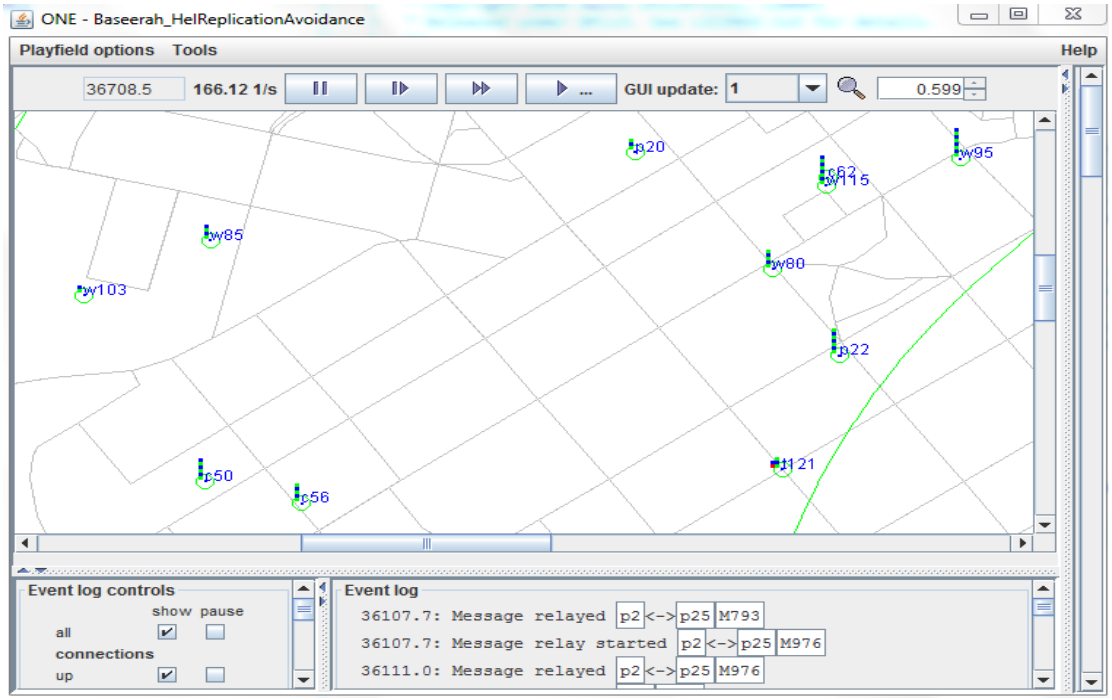


Figure 3.17: Duplication Avoidance on Helsinki Simulation Region

Figure 3.17 shows how duplication avoidance is used in the integrated routing to reduce congestion in the routing process. Before a node receives a message from a sending node it must check its buffer in order not to save the same copies of message. The results obtained from this simulation are shown in Table 4.6

The flow chart of the simulation process is shown in Figure 3.18

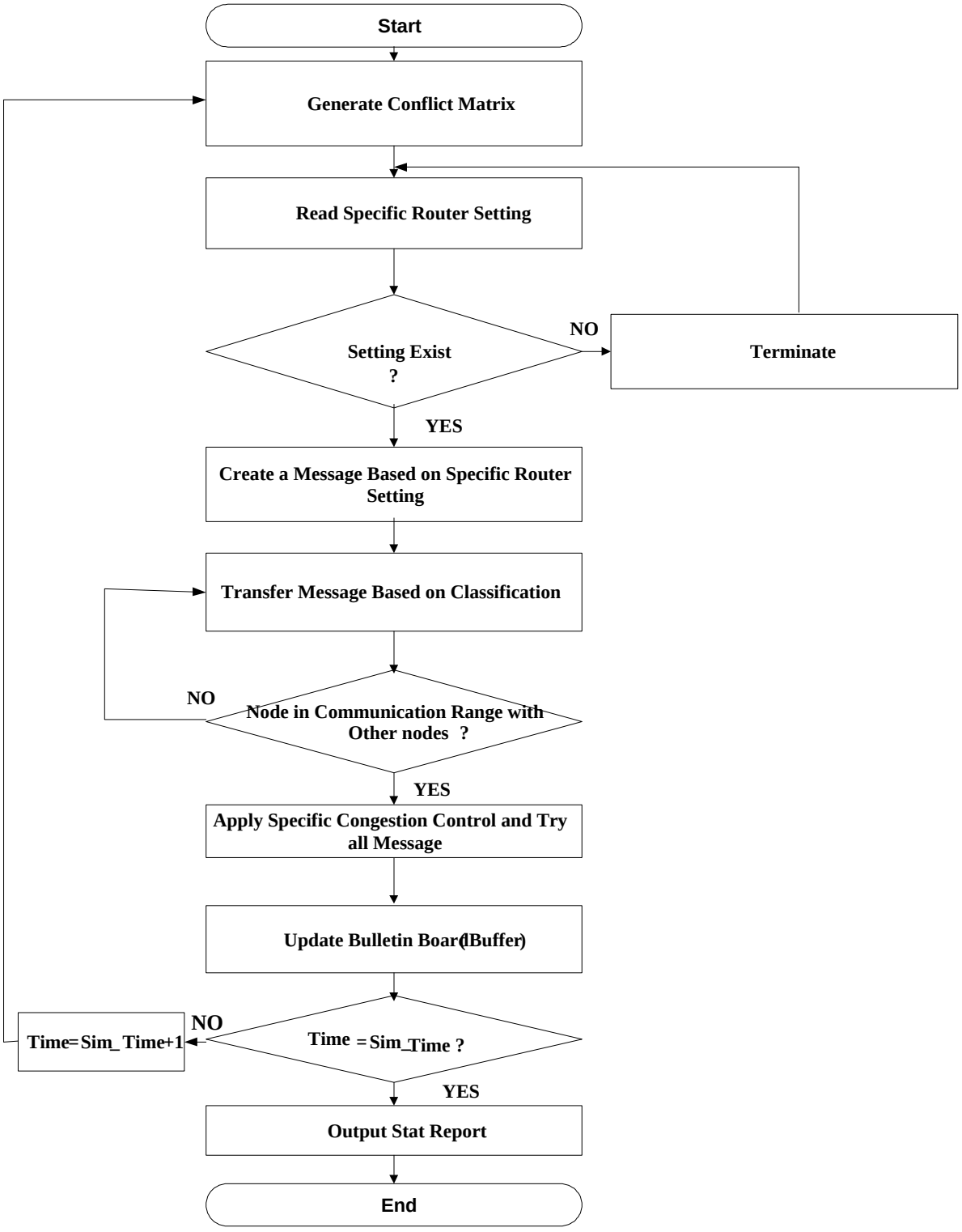


Figure 3.18: Flow Chart of the Routing Algorithm with Congestion Control.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

This section evaluates the performance of the improved integrated routing protocol in the ONE simulator. The following variables were used in the simulation:

Sim_time: This refers to the total time used for the simulation.

Delivery probability: Refers to total probability of the messages delivery.

Packet loss: This refers to the total number of messages that were aborted during the simulation time.

4.1The Performance Evaluation of the Congestion Control Strategies

The congestion control strategies were first implemented on the 10-node test scenario in order to establish their workability. The ONE simulator is capable of generating statistical results from the simulation performed when configured to do so. This is achieved using the following commands:

Report.nrofReports = 1

Report.reportDir = reports/Baseerah

Report.report1 = MessageStatsReport

The ONE simulator keeps the report in a directory named report/Baseerah.The summary of results obtained for the epidemic, prophet and the integrated routing protocol is shown in the

message statistics report in Table 4.1.The simulation time is in seconds (s) and 43200s was used in order to benchmark it with the standard simulation time used on Helsinki simulations by Verma & Srivastava, (2012).

Table 4.1: Message Statistical Report

Variables	Prophet	Epidemic	Integrated
Sim_time (s)	43200.0000	43200.0000	43200.0000
Aborted	251	278	247
Delivery_prob	0.1922	0.1899	0.2008

The integrated routing protocol performed better thanboth the epidemic and prophet routing protocol because it has a higher delivery probability and less aborted messages. The integrated routing protocol is better than the prophet and the epidemic routing protocol with 4.3% and 5.4% respectively. The significance of this result is that more messages will get to their final destination in the integrated routing protocol than the epidemic and the prophet routing protocol.

The result obtained for the integrated routing protocol on the Helsinki map is shown in Table 4.2.

Table 4.2: Message Statistical Report for the Helsinki Integrated Routing.

Variables	Helsinki_Integrated
Sim_time (s)	43200.0000
Aborted	670
Delivery_prob	0.048

For the ten nodes used, the delivery probability of the integrated routing protocol implemented is higher than that of Helsinki integrated routing because nodes on Helsinki follow a predefined map path. In the implemented integrated routing protocol, the nodes contact probability is reduced, resulting in a decrease in the delivery probability as shown in equation (2.2).

The summary of results obtained for the integrated routingsimulation without congestion control in Table 4.1 was expanded by running the simulation at discrete point of the simulation time in order to see how packet loss and delivery probability vary with time from the initial stage. This is shown in Table 4.3, where PL denotes the number of packet loss and DP denotes delivery probability.

Table 4.3: Results for Integrated Routing Simulation(IRP) for the 10-Node Test Case

Integrated Routing		
Sim time (s)	**PL**	**DP**
0	0	0
1000	0	0
2000	0	0.116
3000	5	0.165
4000	6	0.183
5000	8	0.191
6000	14	0.194
8000	23	0.196
10000	33	0.194
12000	49	0.202
14000	66	0.203
16000	76	0.206
18000	97	0.202
20000	96	0.2031
22000	114	0.2011
24000	128	0.202
26000	142	0.202
28000	159	0.205
30000	179	0.206
32000	168	0.209
34000	195	0.209
36000	199	0.214
38000	227	0.215
40000	241	0.199
42000	243	0.201
43200	248	0.2008

The results obtained for the improved integrated routing with congestion control (using acknowledgement, buffer size advertisement, data centric method and the avoidance of duplication)are shown in Table 4.4.

Table 4.4: Results of Simulating the Improved Integrated Routing with Congestion Control(ITPCC) for the 10-Node Test Case

	Acknowledgment		**Advertisement**		**Data Centric**		**Duplication Avoidance**	
Sim time (s)	**PL**	**DP**	**PL**	**DP**	**PL**	**DP**	**PL**	**DP**
0	0	0	0	0	0	0	0	0
1000	0	0.099	0	0.099	0	0.099	0	0.099
2000	0	0.099	0	0.099	0	0.099	0	0.099
3000	2	0.147	2	0.157	2	0.154	3	0.167
4000	2	0.135	2	0.143	2	0.143	3	0.15
5000	5	0.146	4	0.158	4	0.152	4	0.17
6000	9	0.152	7	0.163	9	0.158	8	0.163
8000	13	0.161	11	0.176	14	0.165	10	0.188
10000	18	0.17	13	0.193	17	0.178	15	0.204
12000	28	0.165	24	0.185	27	0.167	25	0.202
14000	36	0.17	32	0.187	38	0.179	36	0.2
16000	39	0.167	34	0.183	39	0.174	39	0.202
18000	42	0.169	40	0.184	47	0.174	42	0.199
20000	58	0.178	51	0.19	55	0.178	55	0.205
22000	59	0.175	57	0.189	60	0.181	65	0.207
24000	64	0.178	62	0.196	62	0.185	66	0.205
26000	66	0.174	64	0.19	69	0.176	68	0.203
28000	76	0.182	80	0.195	75	0.188	80	0.204
30000	79	0.183	82	0.197	76	0.189	86	0.208
32000	95	0.185	90	0.2	88	0.189	87	0.207
34000	98	0.18	91	0.192	92	0.187	92	0.207
36000	105	0.192	96	0.205	102	0.197	100	0.215
38000	120	0.195	116	0.213	103	0.199	116	0.217
40000	122	0.201	120	0.21	109	0.202	124	0.229
42000	130	0.202	126	0.215	127	0.206	126	0.226
43200	145	0.202	141	0.217	126	0.208	126	0.227

In Table 4.4, it can be seen that the duplication avoidance performed better with a DP of 22.70% and a PL of 126 as compared to the use of acknowledgement, buffer size advertisement and data centric method with DP of 20.2%, 21.7% and 20.8%, respectively and PL of 145, 141 and126, respectively at the end of the simulation time.

Figures 4.1 and 4.2 showthe graphical representation of improved integrated routing with congestion control as compared with the conventional integrated routing protocol for the 10-node test case (that is, the graphical representation of Tables 4.3 and 4.4). Figure 4.1 shows the variation of packet loss with time while Figure 4.2 shows the variation of delivery probability with time. Figures 4.2 and 4.1 are generated using the Matlab (Matlab 2013Rb) script of Appendices F1 and F2, respectively.

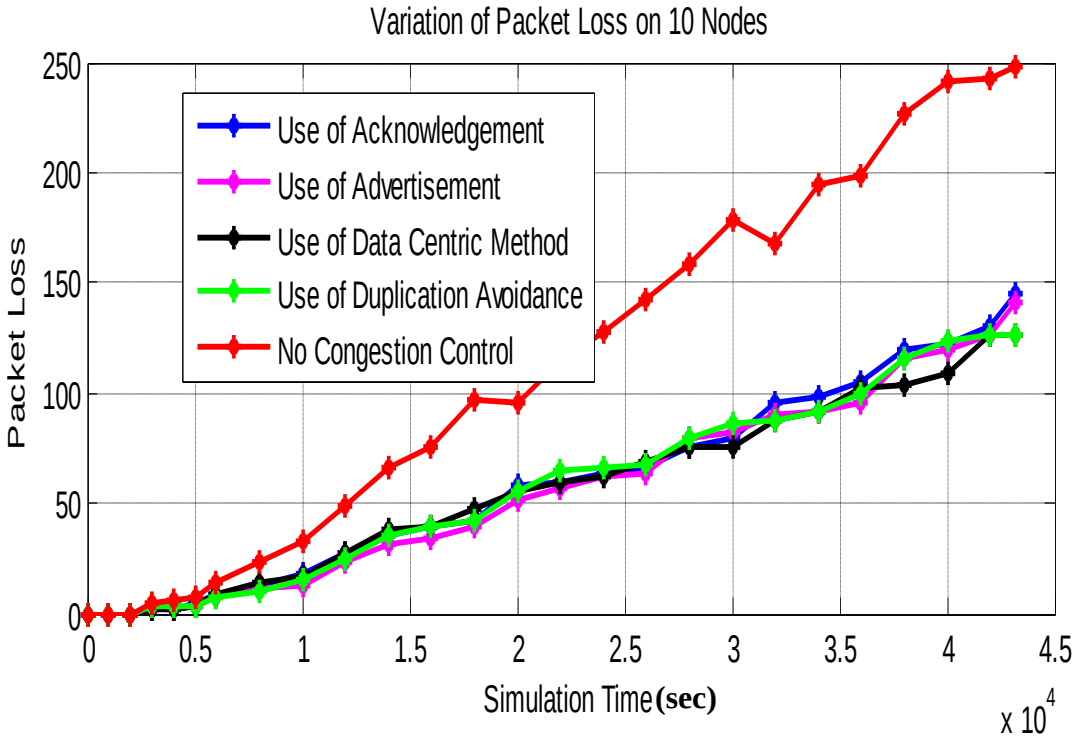


Figure 4.1: Variation of Packet Loss with Time

It can be observed from Figure 4.1 that integrated routing without congestion control has higher packet loss as compared to those with congestion control. This shows that the congestion control strategies have improved the performance of the integrated routing protocol by decreasing the number of packet loss.

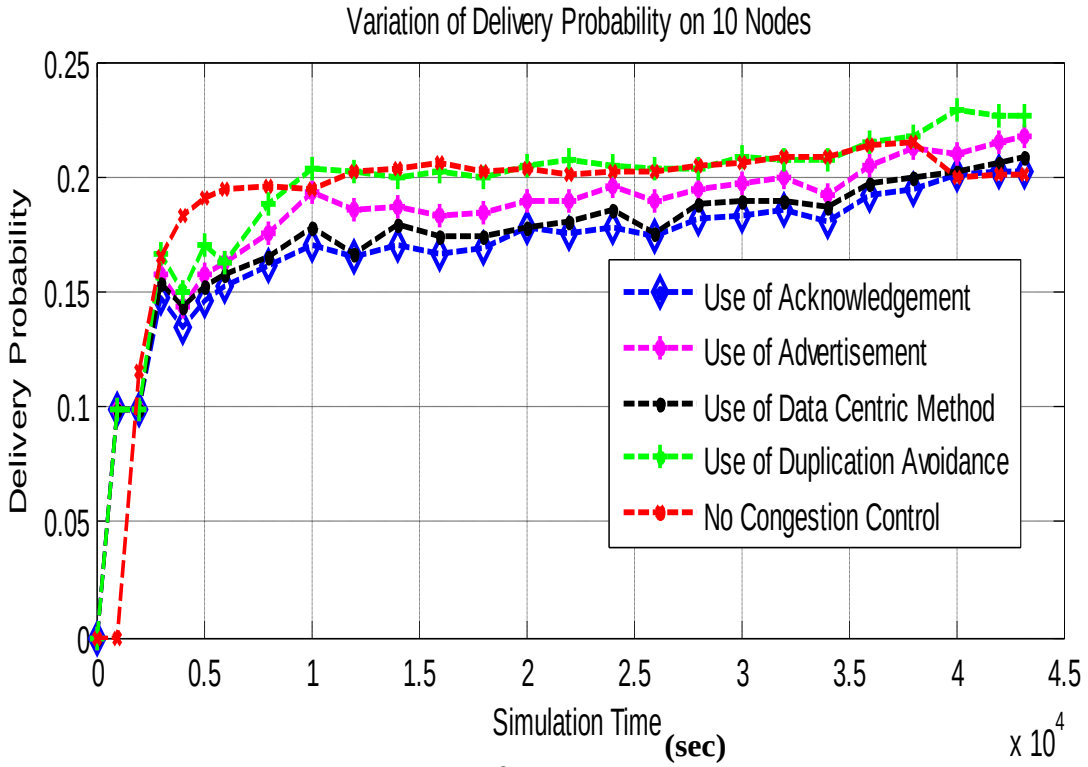


Figure 4.2: Variation of Delivery Probability with Time

In Figure 4.2, it can be seen that congestion control strategies did not have much impact on the delivery probability. However, the congestion control strategies outperformed the integrated routing protocol without congestion control towards the end of the simulation time. Duplication avoidance outperformed other strategies almost at all time.

Results obtained from the validation of the model onthe benchmark Helsinki simulation areaare shown in Tables 4.5 and 4.6.Table 4.5 is an expanded version of Table 4.2 achieved by running

the simulation at discrete point of the simulation time in order to see how packet loss (PL) and delivery probability (DP) vary with time from the initial stage.

Table 4.5: Integrated Routing Simulation(IRP) without Congestion Control for Helsinki Region

Integrated Routing		
Sim time (s)	**PL**	**DP**
0	0	0
1000	0	0.078
2000	0	0.099
3000	6	0.082
4000	25	0.082
5000	36	0.099
6000	48	0.102
8000	74	0.103
10000	84	0.102
12000	119	0.101
14000	155	0.103
16000	180	0.103
18000	192	0.103
20000	215	0.104
22000	245	0.103
24000	280	0.104
26000	320	0.105
28000	385	0.106
30000	420	0.106
32000	452	0.104
34000	485	0.104
36000	519	0.103
38000	564	0.103
40000	602	0.104
42000	648	0.1052
43200	670	0.1048

Table 4.5 shows the validation result obtained when the integrated routing is applied to the Helsinki simulation region. It can be observed that the proposed method has packet loss and delivery probability of 670 and 10.48%, respectively at the end of the simulation time.

Table 4.6: Improved Integrated Routing With Congestion Control (IRPCC) forHelsinki Region.

	Acknowledgement		**Advertisement**		**Data Centric**		**Duplication Avoidance**	
Sim time (s)	**PL**	**DP**	**PL**	**DP**	**PL**	**DP**	**PL**	**DL**
0	0	0	0	0	0	0	0	0
1000	0	0.099	0	0.099	0	0.099	0	0.099
2000	5	0.099	2	0.099	0	0.099	0	0.099
3000	11	0.099	2	0.099	2	0.099	1	0.099
4000	24	0.099	2	0.099	3	0.099	2	0.099
5000	36	0.099	2	0.099	5	0.099	4	0.099
6000	48	0.099	2	0.099	5	0.099	3	0.099
8000	84	0.099	3	0.102	6	0.102	3	0.099
10000	117	0.102	9	0.102	8	0.102	7	0.102
12000	155	0.101	19	0.102	18	0.101	15	0.102
14000	180	0.101	30	0.101	28	0.1	25	0.101
16000	216	0.102	42	0.101	40	0.101	38	0.101
18000	242	0.106	54	0.1	53	0.1	53	0.103
20000	280	0.105	76	0.102	78	0.101	74	0.106
22000	320	0.106	101	0.103	105	0.101	98	0.107
24000	350	0.105	120	0.103	122	0.103	111	0.106
26000	380	0.105	143	0.102	148	0.102	130	0.107
28000	415	0.105	158	0.102	160	0.103	152	0.107
30000	432	0.106	178	0.102	179	0.104	172	0.11
32000	462	0.106	191	0.102	195	0.102	187	0.112
34000	480	0.102	210	0.102	208	0.102	205	0.107
36000	513	0.105	217	0.102	220	0.102	213	0.107
38000	564	0.106	237	0.103	240	0.103	231	0.108
40000	594	0.106	261	0.103	265	0.104	256	0.108
42000	632	0.105	279	0.103	280	0.104	275	0.109
43200	653	0.105	286	0.103	290	0.105	283	0.109

Like in the 10-node scenario, it can be seen from Table 4.6 that the duplication avoidance performs better with a final PL of 283 and DP of 10.9% as compared to the use of

acknowledgement, buffer size advertisement and data centric method with PL of 653, 286 and 290, respectively and DP of 0.105, 0.103 and 0.105, respectively.

The graphical representation of the improved integrated routing protocol with congestion control on Helsinki as compared with that of the conventional integrated routing protocol is shown on Figures 4.3 and 4.4 (that is, the graphical representation of Tables 4.5 and 4.6).Figure 4.3 shows the variation of packet loss with time, while Figure 4.4 shows the variation of delivery probability with time. Figures 4.4 and 4.3 are generated using the Matlab (Matlab 2013Rb) script of Appendices F3 and F4, respectively.

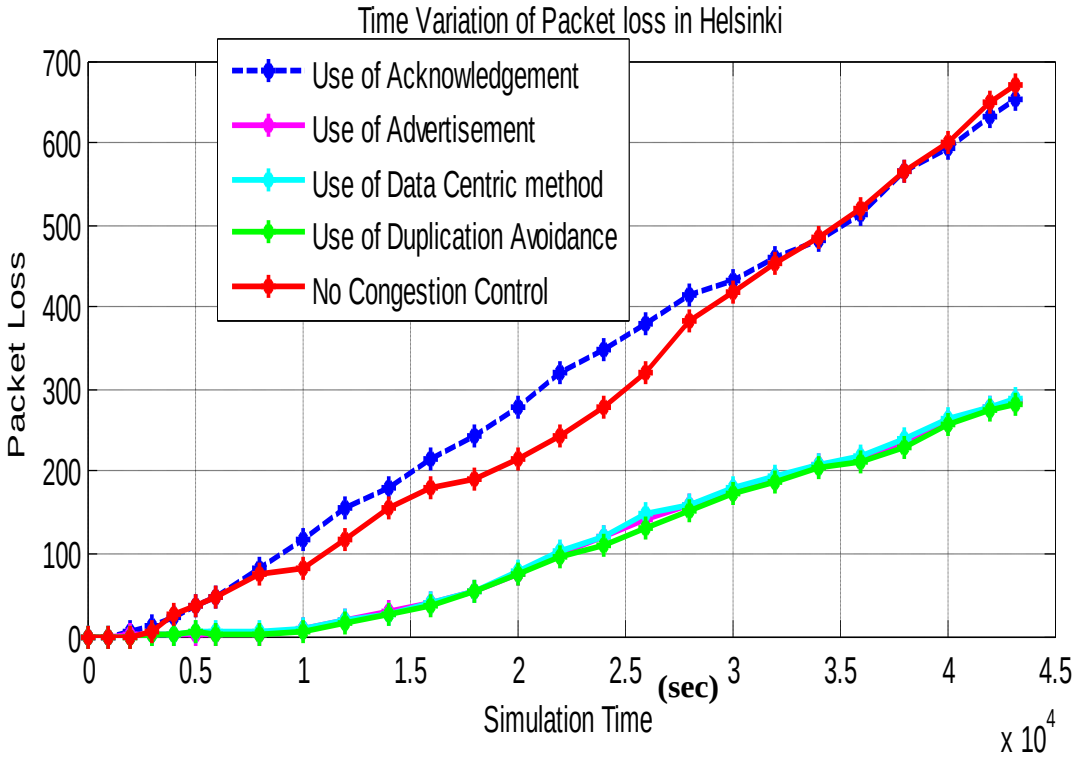


Figure 4.3: Variation of Packet Loss with Time for Helsinki Region

Figure 4.3 shows that buffer size advertisement, data centric method and the avoidance of duplication reduced the packet loss as compared to the integrated routing protocol without

congestion control. The use of acknowledgement reduced the packet loss onlytowards the end of the simulation time.

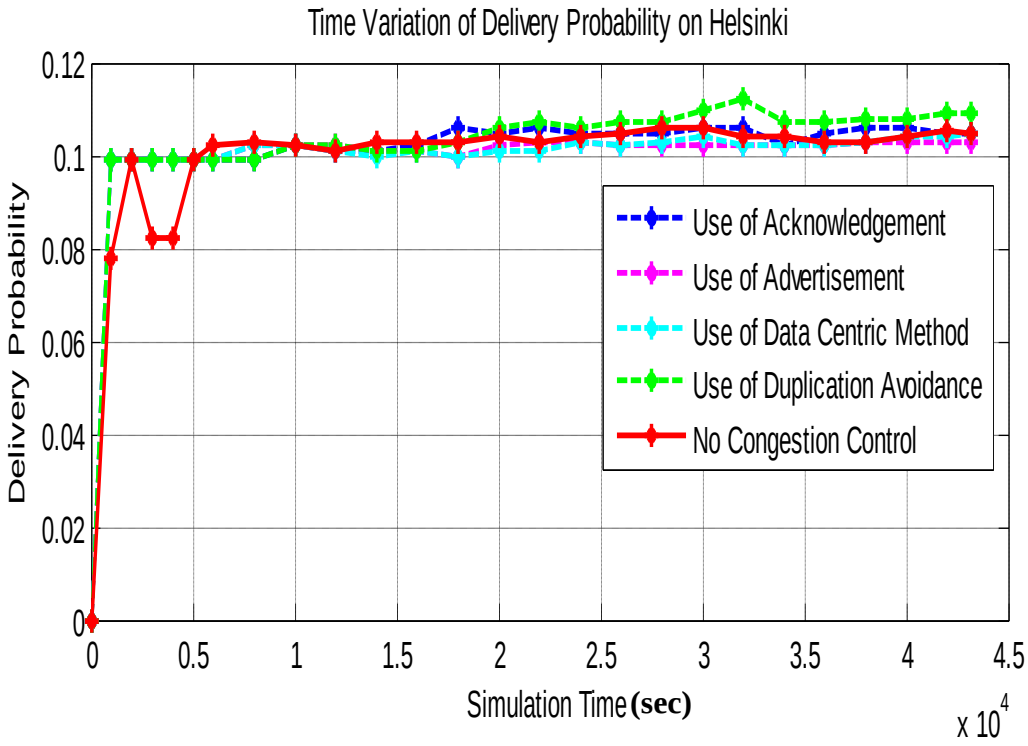


Figure 4.4: Variation of Delivery Probability with Time for Helsinki Region

From Figure 4.4, duplication avoidance outperformed other strategies as it has higher delivery probability almost throughout the simulation time.

4.3 General Discussion

The integrated routing protocol performed better than the epidemic and the prophet routing protocols because it overcame the problem encountered when new nodes join the network (their context information is not yet available in the network) in the prophet routing protocol. Also, the integrated routing protocol reduced the wastage of resources (continuous flooding of packet waste network resources for example, buffer space and bandwidth) of the epidemic routing protocol.

Congestion control using buffer size advertisement and the data centric methods reduced the number of packet loss but did not improve the delivery probability of the messages significantly. This is because there is no end-to-end connection in an opportunistic network.Hence, nodes are using any slight opportunity available to send messages. These forms of congestion control prevent nodes from accepting all messages, (due to buffer size or based on interest), as a result reducing the forwarding opportunity (forwarding probability) and consequently the delivery probability.

The use of acknowledgement did not improve the delivery probability because of the absence of end-to-end connectivity in an opportunistic network. The acknowledgement is either not received at all or not received in reasonable time. Hence, nodes still keep the messages in their buffer long after it has reached its destination. Thisis not an effective way of preventing congestion in an opportunistic network because of theabsence of end-to-end connectivity in an opportunistic network.

The duplication avoidance reduced the number of packet loss, as well as improved the delivery probability. This is because this method did not block any chance of forwarding messages except those messages thatalready exist in the node. Hence, it did not decrease forwarding probability and it saved buffer space by preventing duplication. This resulted in an improved delivery probability, as well as reduced the number of packet loss.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

This chapter presents the conclusion and limitationsencountered during the course of this research. Areas of future research have also been recommended.

5.2 Conclusions

The improved integrated routing protocol was developed by incorporating congestion control strategies into the conventional integrated routing protocol.Four pre-emptive congestion control strategies (acknowledgement, buffer size advertisement, data centric method and the avoidance of duplication) were used. The improved integrated routing protocol was simulated and compared with the conventional integrated routing protocol. Results showed that the duplication avoidance improved the integrated routing protocol better because it reduced the packet loss and improved the delivery probability. Duplication avoidance reduced the packet loss by 58% and improved the delivery probability by 4% at the end of the simulation time. The use of acknowledgement, buffer size advertisement, data centric method reduced the packet loss by 2.5%, 57% and 57%, respectively, but did not improve on the delivery probability significantly. Their delivery probabilities are 0.105, 0.103 and 0.105, respectively at the end of the simulation time. The delivery probability and packet loss of the integrated routing protocol without congestion control are 0.1048 and 670, respectively. These results obtained from validating the model on Helsinki simulation region conformed to the results obtained using the 10-node test case, which shows that, duplication avoidance is the preferred congestion control strategy for the integrated routing considered in this research.

5.3 Limitations

The limitations of this research work are itemized as follows:

i) This research work did not consider security on the integrated routing protocol.

ii) A simplified assumption that all devices in the opportunistic network have the same storage space was made.

iii) This research work did not consider bandwidth utilization of the routing protocols.

5.4 Recommendations for Further Work

The following possible areas are recommended for future work:

i. Security measures can be considered in the integrated routing of the opportunistic networks since malicious devices are likely to join the opportunistic network because of lack of initial authentication.

ii. All the congestion control strategies can be integrated as a hybrid congestion control strategy and used to further improve the performance of the integrated routing protocol.

iii. Extensive experimental setup (like the vehicular mobility model) can be used to evaluate the performance of the integrated routing protocol.

REFERENCES

Amin, R., Wang, K.-C., & Ramanathan, P. (2007). An integrated routing and schedulingapproach for persistent vehicle communication in mobile wimax mesh networks.*Paperpresented at the Military Communications Conference, 2007. MILCOM 2007. IEEE.*

Asgari, C., Zareie, A., & Torkashvand, R. R. (2013). Intelligent Routing for Opportunistic Networks Based on Distributed Learning Automata. *Journal of Basic and Applied Science Research,* 3(7) 117-126.

Basalamah, A., Kim, S. M., Guo, S., He, T., & Tobe, Y. (2012). Link correlation aware opportunistic routing.*Paper presented at the INFOCOM, 2012 Proceedings IEEE.*

Bjurefors, F. (2014). opportunistic networking: Congestion, Transfer Ordering and Resilience.*http://uu.diva-portal.org/smash/get/diva2:713179/FULLTEXT01.pdf*

Benamar, N., Singh, K.-D., Benamar, M., Ouadghiri, & D., Bonnin, J. Routing protocols in Vehicular Delay Tolerant Networks:A comprehensive survey, *Computer Communications, Elsevier. 48 (2014) 141–158*

Chen, L.-J., Yu, C.-H., Sun, T., Chen, Y.-C., & Chu, H.-h. (2006).A hybrid routing approach for opportunistic networks.*Paper presented at the Proceedings of the 2006 SIGCOMM workshop on Challenged networks,* pp 213-220.

Dinakar, S., R.M.Bhavadharini, & S.Karthik. (2013). study of opportunistic network and MANET. *International journal of software & Hardware Reasearch in Engineering, 1*(4).

Hu, W., Xie, J., & Zhang, Z. (2013). Practical opportunistic routing in high-speed multi-rate wireless mesh networks.*Paper presented at the Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing.*

Huang, C.-M., Lan, K.-c., & Tsai, C.-Z. (2008). A survey of opportunistic networks.*Paper presented at the Advanced Information Networking and Applications-Workshops, 2008. AINAW 2008. 22nd International Conference on.*

Islam, M. A., & Waldvogel, M. (2011). Questioning flooding as a routing benchmark in Opportunistic Networks. *Paper presented at the Internet Communications (BCFIC Riga), 2011 Baltic Congress on Future,* pp. 128-133

Jones, E. P., & Ward, P. A. (2006). Routing strategies for delay-tolerant networks. *Submitted to ACM Computer Communication Review (CCR).*

Journi, K., Jorg, O. (2008). Time scales and delay-tolerant routing protocols. *Proceedings of CHANTS'08, San Francisco, California, USA,* pp. 13-19.

Kaur, E. U., & Kaur, E. H. (2009). Routing techniques for opportunistic networks and Security Issues. *National Conference on Computing, communication and control*.

Keranen, A., & Ott, J. (2007). Increasing Reality for DTN Simulation.*Technical report, Helsinki University of Technology, Networking Laboratory.*

Keranen, A., & Ott, J. (2009). The ONE Simulator for DTN Protocol Evaluation. *Special report , Helsinki University of Technolog , Networking Laboratory.*

Lilien, L., Kamal, Z. H., Bhuse, V., & Gupta, A. (2006). Opportunistic networks: the concept and research challenges in privacy and security. *Proc. of the WSPWN*, 134-147.

Lin, Y., Li, B., & Liang, B. (2008). *CodeOR:* Opportunistic routing in wireless mesh networks with segmented network coding.*Paper presented at the Network Protocols, 2008. ICNP 2008. IEEE International Conference on.*

Lindgren, A., Doria, A., & Schelen, O. (2003) a. Epidermic Routing for Partially Connected Adhoc Networks. *ACM Mobile Computing and Communications Rrview, 7*, 19-20.

Lindgren, A., Doria, A., & Schelen, O. (2003) b. Probablistic Routing in intermittentlyconnected Network, *ACM Mobile Computing and Communications Review,* Vol.7, pp.19-20.

MartíN-Campillo, Abraham, Crowcroft, Jon, Yoneki, Eiko, & Martí, Ramon. (2013). Evaluating opportunistic networks in disaster scenarios. *Journal of Network and Computer Applications, 36*(2), 870-880.

Moreira, W., Mendes, P., & Sargento, S. (2011). Assessment model for opportunistic routing.*Paper presented at the Communications (LATINCOM), 2011 IEEE Latin-American Conference on,* 1-6.

Nguyen, H. A., Giordano, S., Puiatti, A. (2007). Probabilistic routing protocol for intermittently connected mobile Ad hoc network. *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, Espoo, Finland, pp.1-6.*

Oliveira, A. B., del Duca Almeida, V., da Hora, D. N., & Macedo, D. F. (2014). Evaluating contacts in opportunistic networks over more realistic simulation models. *Journal of Applied Computing Research, 3*(1), 54-63.

Orozco, J., Santos, R., Ochoa, S. F., & Meseguer, R. (2013) Stochastic Performance Evaluation of Routing Strategies in Opportunistic Networks. *www.dcc.uchile.cl/TR/2013/TR_DCC-20131029-009.PDF.*

Pelusi, L., Passarella, A., & Conti, M. (2006). Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. *Communications Magazine, IEEE, 44*(11), 134-141.

Pan, D., Ruan, Z., Zhou, N., Liu, X & Song, Z. (2013). A Comprehensive-integrated buffer management strategy for opportunistic Network. *EURASIP Journal on Wireless Communication and Networking.*2013: 103

Ristanovic, N. (2012). Modeling and Measuring Performance of Data Dissemination in Opportunistic Networks. *http://infoscience.epfl.ch/record/180634/files/EPFL_TH5448.pdf*

Santos, R.M., Orozco, J., Ochoa, S. (2011). A real-time analysis approach in opportunistic networks. ACM SIGBED Rev. 8(3), 2011.

Silva, Aloizio P, Burleigh, Scott, Hirata, Celso M, & Obraczka, Katia. (2015). A survey on congestion control for delay and disruption tolerant networks. *Ad Hoc Networks,Elsevier 25*, 480-494.

Spyropoulos, T., Psounis, K., Raghavendra, C. S.(2005) Spray and wait: an efficient routing in intermittently connected mobile networks. *Proceeding of ACM SIGCOMM Workshop on Delay Tolerant Networking (WDTN), Philadelphia, USA*, pp. 252-259.

Tsapeli, Fani, & Tsaoussidis, Vassilis. (2012). Routing for opportunistic networks based on probabilistic erasure coding *Wired/Wireless Internet Communication* (pp. 257-268): Springe

Trivini-Cabrera, A., & Canadas- Hurtalo, S. (2011). Survey on opportunistic routing in multihop wireless networks. *International Journal of Communication Networks and Information Security. 3(2), pp. 170-177*

Vahdat, A., & Becker, D. (2000). Epidemic Routing for Partially Connected Ad Hoc Networks. *Technical Report CS-2000-06, Computer Science Departmenr. Duke University.*

Verma, A., & Srivastava, D. (2012). Integrated routing protocol for opportunistic networks. *arXiv preprint arXiv:1204.1658*.

Vidya, K., Hemanth, S. R. (2014) Routing and Congestion Control Strategies in Opportunistic Networks: A Survey. *International Journal For Technical Research In Engineering. 2(11), 2347-4718.*

Yogi,M. K.,& Chinthala, V. (2014). A Study of Opportunistic Networks for Efficient Ubiquitous Computing. *International Journal of Advanced Research in Computing and Communication Engineering.*3(1), 5187-5191

Zhao, Z., Mosler, B. R., & Braun, T. (2012). Performance evaluation of opportunistic routing protocols: a framework-based approach using OMNeT++.*Paper presented at the Proceedings of the 7th Latin American Networking Conference*, pp 28-35

http://www.netlab.tkk.fi/tutkimus/dtn/theone/down/one_1.5.1.tar.gz

http://www.eclipse.org/downloads/

https://netbeans.org/community/releases/73/index.html

Appendix A1

The Program Class for Epidemic Router

```
package routing;

import core.Settings;

//Epidemic message router

public class EpidemicRouter extends ActiveRouter {

	public EpidemicRouter(Settings s) {

		super(s);

		//TODO: read&use epidemic router specific settings (if any)

	}

	protected EpidemicRouter(EpidemicRouter r) {

		super(r);

		//TODO: copy epidemic settings here (if any)

	}

	@Override

	public void update() {

		super.update();

		if (isTransferring() || !canStartTransfer()) {

			return; // transferring, don't try other connections yet

		}

		if (exchangeDeliverableMessages() != null) {

			return; // started a transfer, don't try others (yet)

		}

		// then try any/all message to any/all connection

		this.tryAllMessagesToAllConnections();

	}

	@Override

	public EpidemicRouter replicate() {

		return new EpidemicRouter(this);

	}
```

Appendix A2

Epidemic Routing Settings

Scenario.name = Baseerah_Epidemic

Scenario.simulateConnections = true

Scenario.updateInterval = 1.0

Scenario.endTime = 43200

Scenario.nrofHostGroups = 1

btInterface.type = SimpleBroadcastInterface

btInterface.transmitSpeed = 250k

btInterface.transmitRange = 10

highspeedInterface.type = SimpleBroadcastInterface

highspeedInterface.transmitSpeed = 10M

highspeedInterface.transmitRange = 1000

Group.movementModel = RandomWaypoint

Group.router = EpidemicRouter

Group.bufferSize = 10M

Group.nrofInterfaces = 1

Group.interface1 = btInterface

Group.speed = 0.5, 1.5

Group.msgTtl = 300

Group.nrofHosts = 10

Group.groupID = n

MovementModel.rngSeed = 1

MovementModel.worldSize = 450, 340

Events.nrof = 1

Events1.class = MessageEventGenerator

Events1.interval = 25,35

Events1.size = 500k,1M

Events1.hosts = 0,4

Events1.prefix = M

Report.nrofReports = 1

Report.reportDir = reports/Baseerah

Report.report1 = MessageStatsReport

Optimization.cellSizeMult = 5

Optimization.randomizeUpdateOrder = true

Appendix B1

Program for Prophet Routing

```
public class ProphetRouter extends ActiveRouter {
        public static final double P_INIT = 0.75;
        public static final double DEFAULT_BETA = 0.25;
        public static final double GAMMA = 0.98;
        public static final String PROPHET_NS = "ProphetRouter";
    public static final String SECONDS_IN_UNIT_S ="secondsInTimeUnit";
        public static final String BETA_S = "beta";
    private int secondsInTimeUnit;
    private double beta;
    private Map<DTNHost, Double> preds;
    private double lastAgeUpdate;
public ProphetRouter(Settings s) {
        super(s);
        Settings prophetSettings = new Settings(PROPHET_NS);
        secondsInTimeUnit = prophetSettings.getInt(SECONDS_IN_UNIT_S);
        if (prophetSettings.contains(BETA_S)) {
                beta = prophetSettings.getDouble(BETA_S);
        }
        else {
                beta = DEFAULT_BETA;
        }

        initPreds();
    }
    protected ProphetRouter(ProphetRouter r) {
        super(r);
        this.secondsInTimeUnit = r.secondsInTimeUnit;
        this.beta = r.beta;
```

```
        initPreds();
}
private void initPreds() {
        this.preds = new HashMap<DTNHost, Double>();
}
@Override
public void changedConnection(Connection con) {
        super.changedConnection(con);
        if (con.isUp()) {
                DTNHost otherHost = con.getOtherNode(getHost());
                updateDeliveryPredFor(otherHost);
                updateTransitivePreds(otherHost);
        }
}
private void updateDeliveryPredFor(DTNHost host) {
        double oldValue = getPredFor(host);
        double newValue = oldValue + (1 - oldValue) * P_INIT;
        preds.put(host, newValue);
}
public double getPredFor(DTNHost host) {
        ageDeliveryPreds(); // make sure preds are updated before getting
        if (preds.containsKey(host)) {
                return preds.get(host);
        }
        else {
                return 0;
        }
}
private void updateTransitivePreds(DTNHost host) {
        MessageRouter otherRouter = host.getRouter();
```

```
            assert otherRouter instanceof ProphetRouter : "PRoPHET only works " +
                    " with other routers of same type";
            double pForHost = getPredFor(host); // P(a,b)
            Map<DTNHost, Double> othersPreds =
                    ((ProphetRouter)otherRouter).getDeliveryPreds();
            for (Map.Entry<DTNHost, Double> e : othersPreds.entrySet()) {
                    if (e.getKey() == getHost()) {
                            continue; // don't add yourself
                    }
                    double pOld = getPredFor(e.getKey()); // P(a,c)_old
                    double pNew = pOld + ( 1 - pOld) * pForHost * e.getValue() * beta;
                    preds.put(e.getKey(), pNew);
            }
    }
    private void ageDeliveryPreds() {
            double timeDiff = (SimClock.getTime() - this.lastAgeUpdate) /
                    secondsInTimeUnit;
            if (timeDiff == 0) {
                    return;
            }
            double mult = Math.pow(GAMMA, timeDiff);
            for (Map.Entry<DTNHost, Double> e : preds.entrySet()) {
                    e.setValue(e.getValue()*mult);
            }
            this.lastAgeUpdate = SimClock.getTime();
    }
    private Map<DTNHost, Double> getDeliveryPreds() {
            ageDeliveryPreds(); // make sure the aging is done
            return this.preds;
    }
```

```
@Override
public void update() {
	super.update();
	if (!canStartTransfer() ||isTransferring()) {
		return; // nothing to transfer or is currently transferring
	}
	if (exchangeDeliverableMessages() != null) {
		return;
	}
	tryOtherMessages();
}
private Tuple<Message, Connection> tryOtherMessages() {
	List<Tuple<Message, Connection>> messages =
		new ArrayList<Tuple<Message, Connection>>();
	Collection<Message> msgCollection = getMessageCollection();
	for (Connection con : getConnections()) {
		DTNHost other = con.getOtherNode(getHost());
		ProphetRouter othRouter = (ProphetRouter)other.getRouter();
		if (othRouter.isTransferring()) {
			continue;
		}
		for (Message m : msgCollection) {
			if (othRouter.hasMessage(m.getId())) {
				continue;
			}
			if (othRouter.getPredFor(m.getTo()) > getPredFor(m.getTo())) {
				// the other node has higher probability of delivery
				messages.add(new Tuple<Message, Connection>(m,con));
			}
		}
```

```
        }

        if (messages.size() == 0) {
                return null;
        }
        Collections.sort(messages, new TupleComparator());
        return tryMessagesForConnected(messages);
}
private class TupleComparator implements Comparator
        <Tuple<Message, Connection>> {

        public int compare(Tuple<Message, Connection> tuple1,
                        Tuple<Message, Connection> tuple2) {
                double p1 = ((ProphetRouter)tuple1.getValue().
                getOtherNode(getHost()).getRouter()).getPredFor(
                                tuple1.getKey().getTo());
                double p2 = ((ProphetRouter)tuple2.getValue().
                                getOtherNode(getHost()).getRouter()).getPredFor(
                                tuple2.getKey().getTo());

                if (p2-p1 == 0) {
                return compareByQueueMode(tuple1.getKey(), tuple2.getKey());
                }
                else if (p2-p1 < 0) {
                        return -1;
                }
                else {
                        return 1;
                }
        }
```

```
}
@Override
public RoutingInfo getRoutingInfo() {
        ageDeliveryPreds();
        RoutingInfo top = super.getRoutingInfo();
        RoutingInfo ri = new RoutingInfo(preds.size() +
                        " delivery prediction(s)");
        for (Map.Entry<DTNHost, Double> e : preds.entrySet()) {
                DTNHost host = e.getKey();
                Double value = e.getValue();
                ri.addMoreInfo(new RoutingInfo(String.format("%s : %.6f",
                                host, value)));
        }
        top.addMoreInfo(ri);
        return top;
}
@Override
public MessageRouter replicate() {
        ProphetRouter r = new ProphetRouter(this);
        return r;
}
```

AppendixB2

Prophet Settings

Scenario.name = Baseerah_Prophet

Scenario.simulateConnections = true

Scenario.updateInterval = 1.0

Scenario.endTime = 43200

Scenario.nrofHostGroups = 1

btInterface.type = SimpleBroadcastInterface

btInterface.transmitSpeed = 250k

btInterface.transmitRange = 10

highspeedInterface.type = SimpleBroadcastInterface

highspeedInterface.transmitSpeed = 10M

highspeedInterface.transmitRange = 1000

Group.movementModel = RandomWaypoint

Group.router = ProphetRouter

ProphetRouter.secondsInTimeUnit = 1

Group.bufferSize = 10M

Group.nrofInterfaces = 1

Group.interface1 = btInterface

Group.speed = 0.5, 1.5

Group.msgTtl = 300

Group.nrofHosts = 10

Group.groupID = n

MovementModel.rngSeed = 1

MovementModel.worldSize = 450, 340

Events.nrof = 1

Events1.class = MessageEventGenerator

Events1.interval = 25,35

Events1.size = 500k,1M

Events1.hosts = 0,4

Events1.prefix = M

Report.nrofReports = 1

Report.reportDir = reports/Baseerah

Report.report1 = MessageStatsReport

Optimization.cellSizeMult = 5

Optimization.randomizeUpdateOrder = true

Appendix C1

Setting for Integrated Routing on 10-node test case

Scenario.name = Baseerah_Integrated

Scenario.simulateConnections = true

Scenario.updateInterval = 1.0

Scenario.endTime = 43200

Scenario.nrofHostGroups = 1

btInterface.type = SimpleBroadcastInterface

btInterface.transmitSpeed = 250k

btInterface.transmitRange = 10

highspeedInterface.type = SimpleBroadcastInterface

highspeedInterface.transmitSpeed = 10M

highspeedInterface.transmitRange = 1000

Group.movementModel = RandomWaypoint

Group.router = [ProphetRouter;EpidemicRouter;]

ProphetRouter.secondsInTimeUnit = 1

Group.bufferSize = 10M

Group.nrofInterfaces = 1

Group.interface1 = btInterface

Group.speed = 0.5, 1.5

Group.msgTtl = 300

Group.nrofHosts = 10

Group.groupID = n

MovementModel.rngSeed = 1

MovementModel.worldSize = 450, 340

Events.nrof = 1

Events1.class = MessageEventGenerator

Events1.interval = 25,35

Events1.size = 500k,1M

Events1.hosts = 0,4

```
Events1.prefix = M

Report.nrofReports = 1

Report.reportDir = reports/Baseerah

Report.report1 = MessageStatsReport

Optimization.cellSizeMult = 5

Optimization.randomizeUpdateOrder = true
```

Appendix D1

Settings for integrated routing on Helsinki

Scenario.name = Helsinki_Integrated

Scenario.simulateConnections = true

Scenario.updateInterval = 0.1

Scenario.endTime = 43200

btInterface.type = SimpleBroadcastInterface

btInterface.transmitSpeed = 250k

btInterface.transmitRange = 10

highspeedInterface.type = SimpleBroadcastInterface

highspeedInterface.transmitSpeed = 10M

highspeedInterface.transmitRange = 1000

Scenario.nrofHostGroups = 6

Group.movementModel = [RandomWalk; RandomWaypoint;]

Group.router = [EpidemicRouter; ProphetRouter;]

Group.bufferSize = [2M; 5M; 10M; 15M; 20M;]

Group.waitTime = 0, 120

Group.nrofInterfaces = 1

Group.interface1 = btInterface

Group.speed = 0.5, 1.5

Group.msgTtl = 300

Group.nrofHosts = 40

Group1.groupID = p

Group2.groupID = c

Group2.okMaps = 1

Group2.speed = 2.7, 13.9

Group3.groupID = w

Group4.groupID = t

Group4.bufferSize = 50M

Group4.movementModel = MapRouteMovement

Group4.routeFile = data/tram3.wkt

Group4.routeType = 1

Group4.waitTime = 10, 30

Group4.speed = 7, 10

Group4.nrofHosts = 2

Group4.nrofInterfaces = 2

Group4.interface1 = btInterface

Group4.interface2 = highspeedInterface

Group5.groupID = t

Group5.bufferSize = 50M

Group5.movementModel = MapRouteMovement

Group5.routeFile = data/tram4.wkt

Group5.routeType = 2

Group5.waitTime = 10, 30

Group5.speed = 7, 10

Group5.nrofHosts = 2

Group6.groupID = t

Group6.bufferSize = 50M

Group6.movementModel = MapRouteMovement

Group6.routeFile = data/tram10.wkt

Group6.routeType = 2

Group6.waitTime = 10, 30

Group6.speed = 7, 10

Group6.nrofHosts = 2

Events.nrof = 1

Events1.class = MessageEventGenerator

Events1.interval = 25,35

Events1.size = 500k,1M

Events1.hosts = 0,126

Events1.prefix = M

MovementModel.rngSeed = 1

MovementModel.worldSize = 4500, 3400

MovementModel.warmup = 1000

MapBasedMovement.nrofMapFiles = 4

MapBasedMovement.mapFile1 = data/roads.wkt

MapBasedMovement.mapFile2 = data/main_roads.wkt

MapBasedMovement.mapFile3 = data/pedestrian_paths.wkt

MapBasedMovement.mapFile4 = data/shops.wkt

Report.nrofReports = 1

Report.warmup = 0

Report.reportDir = reports/Baseerah

Report.report1 = MessageStatsReport

ProphetRouter.secondsInTimeUnit = 30

SprayAndWaitRouter.nrofCopies = 6

SprayAndWaitRouter.binaryMode = true

Optimization.cellSizeMult = 5

Optimization.randomizeUpdateOrder = true

GUI.UnderlayImage.fileName = data/helsinki_underlay.png

GUI.UnderlayImage.offset = 64, 20

GUI.UnderlayImage.scale = 4.75

GUI.UnderlayImage.rotate = -0.015

GUI.EventLogPanel.nrofEvents = 100

Appendix E

program class for Helsinki

```
\* BASEERAH YAHAYA
AHMADU BELLO UNIVERSITY ZARIA */
	public MessageStatsReport() {
		init();
	}
	@Override
	protected void init() {
		super.init();
		this.creationTimes = new HashMap<String, Double>();
		this.latencies = new ArrayList<Double>();
		this.msgBufferTime = new ArrayList<Double>();
		this.hopCounts = new ArrayList<Integer>();
		this.rtt = new ArrayList<Double>();
		this.nrofDropped = 0;
		this.nrofRemoved = 0;
		this.nrofStarted = 0;
		this.nrofAborted = 0;
		this.nrofRelayed = 0;
		this.nrofCreated = 0;
		this.nrofResponseReqCreated = 0;
		this.nrofResponseDelivered = 0;
		this.nrofDelivered = 0;
	}
	public void messageDeleted(Message m, DTNHost where, boolean dropped) {
		if (isWarmupID(m.getId())) {
			return;
		}
```

```
		if (dropped) {
			this.nrofDropped++;
		}
		else {
			this.nrofRemoved++;
		}

		this.msgBufferTime.add(getSimTime() - m.getReceiveTime());
	}
	public void messageTransferAborted(Message m, DTNHost from, DTNHost to) {
		if (isWarmupID(m.getId())) {
			return;
		}
		this.nrofAborted++;
	}
	public void messageTransferred(Message m, DTNHost from, DTNHost to,
			boolean finalTarget) {
		if (isWarmupID(m.getId())) {
			return;
		}
		this.nrofRelayed++;
		if (finalTarget) {
			this.latencies.add(getSimTime() -
				this.creationTimes.get(m.getId()) );
			this.nrofDelivered++;
			this.hopCounts.add(m.getHops().size() - 1);

			if (m.isResponse()) {
```

```
                        this.rtt.add(getSimTime() -m.getRequest().getCreationTime());
                        this.nrofResponseDelivered++;
                }
public void newMessage(Message m) {
        if (isWarmup()) {
                addWarmupID(m.getId());
                return;
        }
        this.creationTimes.put(m.getId(), getSimTime());
        this.nrofCreated++;
        if (m.getResponseSize() > 0) {
                this.nrofResponseReqCreated++;
        }
}
public void messageTransferStarted(Message m, DTNHost from, DTNHost to) {
        if (isWarmupID(m.getId())) {
                return;
        }
        this.nrofStarted++;
}
public void done() {
        write("Message stats for scenario " + getScenarioName() +
                        "\nsim_time: " + format(getSimTime()));
        double deliveryProb = 0; // delivery probability
        double responseProb = 0; // request-response success probability
        double overHead = Double.NaN;     // overhead ratio
        if (this.nrofCreated > 0) {
                deliveryProb = ((1.0 * this.nrofDelivered) / this.nrofCreated
        }
```

```
if (this.nrofDelivered > 0) {
        overHead = (1.0 * (this.nrofRelayed - this.nrofDelivered)) /
                this.nrofDelivered;
}
if (this.nrofResponseReqCreated > 0) {
        responseProb = (1.0* this.nrofResponseDelivered) /
                this.nrofResponseReqCreated;
}
String statsText = "created: " + this.nrofCreated +
        "\nstarted: " + this.nrofStarted +
        "\nrelayed: " + this.nrofRelayed +
        "\naborted: " + this.nrofAborted +
        "\ndropped: " + this.nrofDropped +
        "\nremoved: " + this.nrofRemoved +
        "\ndelivered: " + this.nrofDelivered +
        "\ndelivery_prob: " + format(deliveryProb) +
        "\nresponse_prob: " + format(responseProb) +
        "\noverhead_ratio: " + format(overHead) +
        "\nlatency_avg: " + getAverage(this.latencies) +
        "\nlatency_med: " + getMedian(this.latencies) +
        "\nhopcount_avg: " + getIntAverage(this.hopCounts) +
        "\nhopcount_med: " + getIntMedian(this.hopCounts) +
        "\nbuffertime_avg: " + getAverage(this.msgBufferTime) +
        "\nbuffertime_med: " + getMedian(this.msgBufferTime) +
        "\nrtt_avg: " + getAverage(this.rtt) +
        "\nrtt_med: " + getMedian(this.rtt)
        ;
write(statsText);
super.done();}     }
```

Appendix F1

Variation of Delivery Probability on 10 Nodes

```
clc
%Sim_time: Is the time of simulation
%PD_Ack: is the probability of delevery for acknowledgemenet Congestion
%Control in 10 nodes
%PD_Adv:is the probability of delevery for Advertesement Congestion Control
%in 10 nodes
%PD_DC: is the probability of delevery for Data Centric Congestion Control
%in 10 nodes
%PD_RAv: is the probability of delevery for Duplication Avoidance
%Congestion Control in 10 nodes

Sim_time=input('Please Provide Time of simulation: ');
PD_Ack=input('Please provide the delivery probability: ');
PD_Adv=input('Please provide the delivery probability: ');
PD_DC=input('Please provide the delivery probability: ');
PD_RAv=input('Please provide the delivery probability: ');
PD_Int=input('Please provide the delivery probability: ');
plot(Sim_time,PD_Ack)
hold on
plot(Sim_time,PD_Adv,'m')
hold on
plot(Sim_time,PD_DC,'c')
hold on
plot(Sim_time,PD_RAv,'g')
hold on
plot(Sim_time,PD_Int,'r')
title('Variation of Delivery Probability on 10 Nodes')
xlabel('Simulation Time')
ylabel('Delivery Probability')
legend('Use of Acknowledgement','Use of Advetisement', 'Use of Data Centric
Method', 'Use of duplication Avoidance', 'No Congestion Control')
grid on
```

Appendix F2

Variation of Packet Loss on 10 Nodes

```
%Sim_time: Is the time of simulation
%PL_Ack: Packet Loss for Acknowledgement Congestion Control in 10 nodes
%PL_Adv: Packet Loss for Advertesment Congestion Control in 10 nodes
%PL_RAv: Packet Loss for Duplication Avoidance Congestion Control in 10
%nodes
%PL_Int: Packet Loss for Integrated Congestion Control in 10 nodes
Sim_time=input('Please Provide the Packet Loss: ');
PL_Ack=input('Please provide the Packet Loss: ');
PL_Adv=input('Please provide the Packet Loss: ');
PL_DC=(input('Please provide the Packet Loss: ');
PL_RAvinput('Please provide the Packet Loss: ');
PL_Int=input('Please provide the Packet Loss: ');
plot(Sim_time,PL_Ack)
hold on
plot(Sim_time,PL_Adv,'m')
hold on
plot(Sim_time,PL_DC,'c')
hold on
plot(Sim_time,PL_RAv,'g')
hold on
plot(Sim_time,PL_Int,'r')
title('Variation of Packet Loss on 10 Nodes')
xlabel('Delivery Probability')
ylabel('Simulation Time')
legend('Use of Acknowledgement','Use of Advetisement', 'Use of Data Centric
Method', 'Use of duplication Avoidance', 'No Congestion Control')
grid on
```

Appendix F3

Time Variation of Delivery Probability on Helsinki

```
%Sim_time: Is the time of simulation
%PD_Ack: is the probability of delevery for acknowledgemenet Congestion
%Control in Helsinki
%PD_Adv:is the probability of delevery for Advertesement Congestion Control
%in Helsinki
%PD_DC: is the probability of delevery for Data Centric Congestion Control
%in Helsinki
%PD_RAv: is the probability of delevery for Duplication Avoidance
%Congestion Control in Helsinki
Sim_time=input('Please Provide the simulation time');
PD_HAck=input('Please provide the Delivery Probability:');
PD_HAdv=input('Please provide the Delivery Probability:');
PD_HDC=input('Please provide the Delivery Probability:');
PD_HRAv=input('Please provide the Delivery Probability:');
PD_HInt=input('Please provide the Delivery Probability:');
plot(Sim_time,PD_HAck)
hold on
plot(Sim_time,PD_HAdv,'m')
hold on
plot(Sim_time,PD_HDC,'c')
hold on
plot(Sim_time,PD_HRAv,'g')
hold on
plot(Sim_time,PD_HInt,'r')
grid on
title('Time Variation of Delivery Probability on Helsinki')
xlabel('Simulation Time')
ylabel('Delivery Probability')
legend('Use of Acknowledgement','Use of Advertisement', 'Use of Data Centric
Method', 'Use of Duplication Avoidance','No Congestion Control')
```

Appendix F4

Time Variation of Packet loss in Helsinki.

```
%Sim_time: Is the time of simulation
%PL_Ack: Packet Loss for Acknowledgement Congestion Control in Helsinki
%PL_Adv: Packet Loss for Advertisement Congestion Control in Helsinki
%PL_RAv: Packet Loss for Duplication Avoidance Congestion Control Helsinki
%PL_Int: Packet Loss for Integrated Congestion Control in Helsinki
Sim_time=input('Please provide the simulation time: ');
PL_Hack=input('Please provide the packet loss:');
PL_HAdvinput('Please provide the packet loss:');
PL_HDCinput('Please provide the packet loss:');
PL_HRAvinput('Please provide the packet loss:');
PL_HIntinput('Please provide the packet loss:');
plot(Sim_time,PL_HAck)
hold on
plot(Sim_time,PL_HAdv,'m')
hold on
plot(Sim_time,PL_HDC,'c')
hold on
plot(Sim_time,PL_HRAv,'g')
hold on
plot(Sim_time,PL_HInt,'r')
xlabel('Simulation Time')
ylabel('Packet Loss')
title('Time Variation of Packet loss')
legend('Use of Acknoeledgement','Use of Advertisement','Use of Data Centric
method', 'Use of Duplication Avoidance','No Congestion Control')
grid on
```