JIMMA UNIVERSITY
JIMMA INSTITUTE OF TECHNOLOGY
FACULTY OF COMPUTING AND INFORMATICS

Contact Duration Aware Buffer Management Forwarding Policy in Sparse Delay Tolerant Networks

By: Yadesa Bedasa

Principal Advisor: Mr. Kebebew Ababu (Ass.Prof.)

Co-Advisor: Mr. Gemechu Birhanu (M.Sc.)

THESIS SUBMITTED TO JIMMA UNIVERSITY FACULTY OF COMPUTING AND INFORMATICS IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER NETWORKING

December 8, 2022

Jimma University

Approval sheet

This Independent Research entitled "**Contact Duration Aware Buffer Management Forwarding Policy in Sparse Delay Tolerant Networks**" has been read and approved as meeting the preliminary research requirements of the Faculty of Computing in partial fulfillment for the award of the degree of Master in Computer Networking, Jimma University, Jimma, Ethiopia.

Principal Investigator

Mr. Yadesa Bedasa

Sign ____________

Principal Advisor

Mr. Kebebew Ababu (Assistant. Professor)

Sign: ____________

Co-Advisor

Mr. Gemechu Birhanu (MSc.)

Sign:

External Examiner

Dr. Henok Mulugeta

Sign: ____________

Internal Examiner

Mr. Temesgen Derejaw (MSc)

Sign: Dec 8/2022 ____________

Chair Person

Mrs. Kasech Tsegaye (MSc.)

Sign: ____________

Dedication

First and foremost, I dedicate this thesis to the Almighty God, thanking you for your guidance, strength, mental power, protection, and skill in providing us with a healthy life. I offer you all of these.

This study is entirely dedicated to our beloved parents, who have been our source of inspiration and strength when we felt like giving up, and who continue to provide their moral, spiritual, and emotional support.

And Lastly, I dedicate this thesis to our brothers, sister, relatives, mentor, friends, and teachers who shared their words of advice and encouragement to finish this study.

Abstract

Delay-Tolerant Networks (DTNs) are target environments suffering from instability or no complete path between source and destination nodes. The contact duration between pairs of network nodes is limited due to node movement and limited range of peer-to-peer wireless communication. The pairs of node takes a long time to contact again in sparse delay-tolerant networks, which is longer than contact duration. Therefore limited contact duration needs more consideration to design buffer management forwarding policy. Several work has been done in the design of forwarding algorithms, but little work has focused on studying forwarding under the presence of limited contact duration. The pairs of network nodes require waiting till they come into communication range again when do not finish exchanging the message during the contact time. Consequently, the message which is started to transmit may drop when expired before the nodes contact again and destination node waits till to receive complete message. Also, receiver node drops a complete message to create space for incoming partial message items upon buffer overflow. To solve the raised problem, we proposed a novel contact duration aware buffer management forwarding policy. Since all intended buffered messages may not be exchanged between nodes within a single contact duration, our proposed scheme select and forward only complete transmit messages.

To evaluate the performance of proposed work we have used Opportunistic Network Environment (ONE) network simulator. The proposed algorithm is integrated with the MaxProp routing protocol buffer management forwarding policy and they were analyzed on two different metrics namely bundle delivery probability and average delivery delay. Also, we considered number of node, buffer size and simulation time to test performance of proposed work. Consequently, across all the experiments the simulation results obtained in this thesis show that our proposed algorithm outperforms than existing forwarding policy.

Keywords: DTN, Sparse DTN, Buffer Management Forwarding Policy, Contact Duration

Acknowledgment

Above all, I would like to thank GOD for giving me the strength to achieve whatever I have achieved so far. Next to this, I would like to express my sincere appreciation to my main advisor, Assistant Prof. Mr. Kebebew Ababu for giving me an excellent and constrictive comment and suggestion throughout my thesis work. Additionally, I would like to express my gratitude to my co-advisor, Mr. Gemechu Birhanu, who has provided positive encouragement from the outset of this work, beneficial clarifications, ongoing supervision, and assistance with relevant chores to help me complete my research program on time.

My warm appreciation goes to my friends and classmates for the family relationship we have that shape and encourages me for completing my study. I would like to thank all School of Computing staff members of Jimma University for their support.

Finally, I would like to say thanks to my beloved parents who encourage and appreciate me in innumerable way in every aspect of my work.

Table of Contents

List of Figures I

List of Tables II

Acronyms, Abbreviation, and Terminology III

CHAPTER ONE 1

INTRODUCTION 1

1.1. Background 1

1.2. Statement of the Problem 3

1.3. Objectives of the Research 4

1.3.1. General Objective 4

1.3.2. Specific Objective 4

1.4. Scope and Limitation of the Study 4

1.5. Methodology 5

1.5.1. Literature Review 5

1.5.2. Design and Implementation 5

1.5.3. Evaluation of the Proposed Work 5

1.6. Significance of the Study 5

CHAPTER TWO 7

LITERATURE REVIEW 7

2.1. Overview 7

2.2. History of Delay Tolerant Networks 7

2.3. Delay-Tolerant Network Architecture 8

2.4. Characteristics of Delay-Tolerant Networks 10

2.5. Application Areas of Delay-Tolerant Networks 11

2.6. Connectivity in Delay Tolerant Network 12

2.7. Mobility Model 13

2.7.1. Random Mobility Models 14

2.7.2. Map-Constrained Mobility Models 14

2.7.3. Social Mobility Models 14

2.7.4. Composite Mobility Models 15

2.8. Routing Protocol 15

2.8.1. Encounter Based Routing 16
2.8.2. Time-Based Routing 17
2.8.3. Infrastructure Based Routing 17
2.8.4. Hybrid and Other Routing 18
2.9. Buffer Management Policy 18
2.9.1. Buffer Management Approaches 19
2.9.2. Classifications of Buffer Management Policy 21
2.9.2.1. Protocol-Specific and Non-Protocol-Specific 21
2.9.2.2. Static Dropping and Dynamic Dropping 21
2.10. Challenges in Delay Tolerant Network 22
CHAPTER THREE 23
RELATED WORK 23
3.1. Heuristic-Based Policy 23
3.2. Optimal-Based Policy 25
CHAPTER FOUR 29
DESIGN OF THE PROPOSED SOLUTION 29
4.1. The Architecture of Proposed Solution 30
4.1.2. Forwarding Policy 32
4.1.3. Contact Duration 32
4.1.4. Calculate DSM and DAM 33
4.1.5. Compare CD with DSM and DAM 33
4.1.6. Select Complete Transmit Messages 34
4.2. Summary 36
CHAPTER FIVE 37
IMPLEMENTATIONS AND RESULT EVALUATION 37
5.1. Overview 37
5.2. Experimental Environment Setup 38
5.2.1. The Simulation Scenario 39
5.2.2 Message Generation 40
5.2.3. Mobility Model 40
5.2.4. Reporting and Visualization 41
5.3. Simulation Experiment and Result Analysis 42

5.3.1. Performance Evaluation Metrics 42
5.3.2. Result Analysis and Discussion 43
5.3.2.1. Performance Evaluation Concerning Number of Node 44
5.3.2.2. Performance Evaluation Concerning Simulation Time 45
5.3.2.3. Performance Evaluation Concerning Buffer Size 46
5.4. Summary 47
CHAPTER SIX 48
CONCLUSION, CONTRIBUTION, AND FUTURE WORKS 48
6.1. Conclusion 48
6.2. Contribution 49
6.3. Future works 49
References 50

List of Figures

Figure 2.1: Difference between TCP/IP and DTN Architecture [3]. 9
Figure 2.2: The DTN Architecture Implemented within a Single Node [13]. 10
Figure 2.3: Applications of DTNs [18]. 12
Figure 2.4: Categories of DTNs routing protocols [23]. 16
Figure 2.5: Node architecture [38]. 19
Figure 4.1: Architecture of the proposed solution. 31
Figure 5.1: Simulated Map of a Part of Helsinki, Finland [46]. 38
Figure 5.2: Screenshot of the ONE Simulator GUI. 41
Figure 5.3: Bundle delivery probability by varying number of nodes. 44
Figure 5.4: Bundle average delivery delay by varying number of nodes. 45
Figure 5.5: Bundle delivery probability by varying simulation time. 45
Figure 5.6: Bundle average delivery delay by varying simulation time. 46
Figure 5.7: Bundle delivery probability by varying buffer sizes. 46
Figure 5.8: Bundle average delivery delay by varying buffer sizes. 47

List of Tables

Table 3.1: Summary of related work. 27

Table 5.1: Simulation Environment Parameters. 39

Acronyms, Abbreviation, and Terminology

CDBR: Contact Duration Based Routing

CG: Contact Graph

CLA: Convergence Layer Adaptor

CMM: Community Mobility Model

DAER: Distance Aware Epidemic Routing

DD: Direct Delivery

DF: Delegation Forwarding

DOS: Denial of Service

DSL: Digital Subscriber Line

DTN: Delay Tolerant Network

FC: First Contact

FIFO: First In First Out

GeSoMo: General Social Mobility Model

ICR: Inter Contact Routing

IP: Internet Protocol

IPN: Inter Planetary Internet

LAN: Local Area Network

MANET: Mobile Ad-hoc Network

MBM: Map Based Movement

MF: Message Ferrying

MOVE: Movement of Vehicle

MPAD: Mobility Prediction Based Adaptive Data

MULE: Mobile Ubiquitous LAN Extensions

ONE: Opportunistic Network Environment

PDA: Personal Digital Assistant

PDM: Post-Disaster Mobility

PER: Predict and Relay

PROPHET: Probabilistic Routing Protocol using History of Encounters and Transitivity

RBMM: Route Based Map Movement

RD: Random Direction

RW: Random Way

RWP: Random Way Point

SLAN: Self-similar Least-Action Walk

SPBMM: Shortest Path Based Map Movement

SPMBMM: Shortest Path Map Based Movement Model

TCT: Transmission Control Protocol

WDM: Working Day Movement

WKT: Well-Known Text

WMBR: World Model Based Routing

CHAPTER ONE

INTRODUCTION

1.1. Background

The rapid growth of technology and infrastructure has made communication extremely important and made our lives easier. In modern communication, Internet architecture has been successfully used to connect various networks on Earth. Internet is heavily built upon the TCP/IP protocol stack, and the connections are characterized by low latency, end-to-end connectivity, and low packet loss [1]. The services are typically expected to be always available and always-on. However, Internet protocols rely on certain assumptions that may not be true in the Inter-Planetary Internet (IPN). As an effort to standardize communications for the IPN, the Delay-Tolerant Networking (DTN) architecture and protocols were proposed. Initially, as a concept, it was proposed by NASA for interplanetary communication [2]. Then after the researchers began to consider how the architecture could be applied to other situations where communications were subject to delays and disruptions that would make conventional Internet protocols (especially TCP) ineffective. Also, the concept of Delay-Tolerant Network is extended to mobile ad-hoc networks with dynamic topology changes, network partitions, node mobility, sparse network topologies, terrain obstacles, and resource constraints.

In a challenging environment of delay-tolerant networks, popular ad hoc routing protocols fail to establish routes. Therefore DTN routing protocols must take to a "store and forward" approach, when instantaneous end-to-end paths are difficult or impossible to establish, where data is incrementally moved and stored throughout the network in hopes that it will eventually reach its destination [3]. Based on this concept multiple routing protocols were proposed for delay tolerant-networks. These routing protocols are classified into four types based on considered knowledge to predict the future relay node [4]. Encounter-based routing protocol represents the categories of protocols that differentiate based on encounter property. Also, these protocols are further divided into flooding and prediction-based. The time-based category defines the relay selection prediction based on their time metric. Time metric represents the interval, duration, and inter-meeting time. The infrastructure-based protocol can also be differentiated based on how they use the location information of nodes like route information or measuring distance between nodes. Hybrid and others category belongs to those protocols which have more than one type of information for relay selection or do not belong to any category.

Delay-tolerant networks use multi-copy routing mechanisms in the process of forwarding messages to improve the performance of the network and overcome sudden difficulties. Although multi-copy routing protocols can improve the delivery ratio and reduce the propagation delay. However, they also easily exhaust the limited storage resource of nodes and resulting in buffer overflows. To deal with this issue, multiple protocol-specific and non-protocol-specific buffer management policies have been proposed. The aspect that needs to be taken into consideration for an effective and efficient delay-tolerant network is the buffer management policy. Which is required to select and forward a messages when pair of nodes encounter and drop upon buffer overflow. Delay-tolerant network buffer management includes both forwarding and dropping policies. The forwarding policy is used to determine the messages to be forwarded during a meeting of limited contact duration with another node.

The existing buffer management forwarding policies cover heuristic-based and optimal-based. These policies are not considered limited contact duration between pairs of nodes. However, the contact duration of time between the pair of nodes is limited due to node movement and the limited range of peer-to-peer wireless communication [5]. Hence complete message may not transmit to the requester node within contact duration. Because the message can complete transmit only if the available contact duration is greater than or equal to the message size divided by the available communication bandwidth [6]. Also, pairs of nodes require to wait till they come into communication range again when do not finish exchanging the message during the contact time. The partial message transmission makes an unexpected message drop and destination node waits till to receive complete message. The issue of limited contact duration mostly affects the performance of sparse delay-tolerant networks because the nodes take a long time to meet again. To deal with the issue of limited contact duration in sparse delay-tolerant networks, important to forward only complete transmit messages within contact duration.

We aim to determine the forwarding solution for each node to fully utilize the network resources, such as the limited storage spaces that each node is willing to provide, and the limited node contact duration. To achieve this, only complete transmit messages are must forward within contact duration between pairs of nodes. The number and order of complete transmit messages are determined based on available contact duration between pairs of network nodes. To solve the specified problem of the existing schemes, we proposed a contact duration aware buffer management forwarding policy in sparse delay tolerant networks.

1.2. Statement of the Problem

Delay-Tolerant Networks target environments suffering from the instability or lack of end-to-end connectivity. To handle the issue of connectivity of mobile nodes, the store-carry-forward principle aims to sustain message sessions, and replication to increase the probability of on-time delivery [3]. However, these techniques require efficient buffer management forwarding and dropping policy, to comply with limited resources availability like contact duration and storage. Buffer management forwarding policy is used to determine which message should be replicated during a meeting of limited contact duration with another node. Therefore forwarding policy needs more consideration for effective and efficient delay-tolerant networks.

To this end, a lot of multiple copy routing protocols incorporate distributed buffer management forwarding policy. The existing forwarding policies are mainly considered message states and network knowledge but ignored the limited contact duration. They simply assume that the complete message can always be transmitted as long as a requester node contacts a node storing the message. However, the contact duration is limited due to node movement and the limited range of peer-to-peer wireless communication [5]. Hence complete message item may not be transmitted to the requester node. Therefore pairs of nodes require waiting till they come into communication range again when do not finish exchanging the message during the contact time. Consequently, the message which is started to transmit and interrupted due to limited contact duration may drop at sender node when expired before the nodes contact again and receiver node drops complete messages upon buffer overflow to create a space for incoming partial message items. In addition, the destination node waits till to receive a complete message. Due to limited contact duration partial message transmission has major impact on the performance of sparse delay-tolerant networks because pair of nodes takes a long time to contact again. Therefore buffer management forwarding policy must forward only complete transmit messages within contact duration to enhance performance of sparse delay tolerant networks.

To overcome the stated problem, we proposed a contact duration aware forwarding strategy for buffer management policy in sparse delay tolerant networks. Our proposed scheme forward only complete transmit messages within contact duration to enhance the performance of sparse delay tolerant networks.

Generally, the following research questions are to be answered in this thesis:

- How many messages are complete transmit within a limited contact duration?
- How order complete transmit messages should forward within contact duration?

1.3. Objectives of the Research

1.3.1. General Objective

The general objective of this study is the performance enhancement of buffer management forwarding policy using a contact duration aware forwarding strategy in sparse delay tolerant networks.

1.3.2. Specific Objective

The following specific objectives have to be accomplished to achieve the general objective of the research.

- Investigate and recognize the current buffer management schemes
- Design architecture for the proposed forwarding strategy in the buffer management scheme
- Implement the proposed solution in the sparse scenario on the simulator DTN environment
- Test and Evaluate through simulations the performance of the proposed solution to demonstrate that it enhances the existing buffer management policy
- Compare and contrast the new scheme with existing schemes.

1.4. Scope and Limitation of the Study

The scope of this thesis is delimited to designing a contact duration aware forwarding strategy to enhance the performance buffer management policy in sparse delay tolerant networks. The proposed strategy allows forwarding only complete transmit intended buffered messages within contact duration between pairs of nodes. The proposed scheme operates following to existing forwarding strategy of buffer management policy. Thus the proposed scheme takes prioritized intended messages by existing forwarding policy as input. We apply our proposed new scheme to prioritize intended messages to select and forward only complete transmit messages within a limited contact duration.

This thesis will not cover the following issues:

- Routing protocol algorithm
- Buffer management dropping policy

1.5. Methodology

The research methodologies used to achieve the specified objective of the thesis are described as follows.

1.5.1. Literature Review

To understand the various buffer management techniques in the domain, the research gap in the domain, and the evaluation metrics used to test the performance of the proposed work, related literature works are reviewed in depth. The reviewed related works also enable us to identify the research gap in the domain and to define the scope of the proposed thesis.

1.5.2. Design and Implementation

In the design phase, the proposed solution in sparse delay-tolerant network scenarios that are specified in the objectives of this thesis is designed. Due to the prohibitive costs of employing DTNs, different wireless access technologies, and nodes in real-world test beds, we have implemented the proposed solution using a simulation tool Opportunistic Network Environment (ONE).

1.5.3. Evaluation of the Proposed Work

A simulation experiment was conducted to test the effectiveness of our proposed approach. After conducting the simulation experiments, the results are discussed and interpreted, and finally, triangulation was conducted to see the effectiveness and efficiency of the proposed work. The proposed approach is integrated with the MaxProp routing protocol buffer management policy. It was then compared through visualization and reports tools with the existing MaxProp buffer management policy in terms of message delivery probability and average delivery delay.

1.6. Significance of the Study

The proposed scheme used to make buffer management policy aware of available contact duration between pairs of network nodes. The primary advantage of the scheme is to enhance the performance of buffer management policy in sparse delay-tolerant networks. Since all intended buffered messages may not be exchanged between nodes within a single contact duration, our scheme forward only complete transmit messages within contact duration. Specifically proposed approach will address waiting time till to receive a complete message at the destination node, inappropriate message drop, and consumed network bandwidth by partial message transmission. Generally, sparse delay-tolerant networks can use this finding to facilitate the services.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overview

Networks that can tolerate delays differ from typical networks. DTNs use store and forward message switching to get around the issues caused by sporadic connectivity, protracted or changeable delay, unequal data rates, and high error rates. Likewise created to function effectively over heterogeneous networks without an end-to-end connectivity. In this kind of setting, long delay, measured in hours or days, may be anticipated. There has been a wide range of DTN research. According to [7], DTN is characterized as a novel networking paradigm that permits device interconnection that is not possible with present networking technology; nodes can connect and share information when circumstances permit. As a result, in certain situations, data delivery using conventional routing protocols is not possible.

According to the authors of [8], their wireless networking approach requires a delay-tolerant network to fulfill four key functions. The first, called DTN, addresses the possibility that source-to-destination pathways may not be entirely full in mobile edge networks. It makes use of data MULEs (Mobile Ubiquitous LAN Extensions), drops boxes, and opportunistic linkages. Next, unlike traditional networks where end-to-end connectivity is expected and node in the network is optimized individually, DTN enables each node in the network to be optimized distinctively and separately. As a result, it handles local congestion, delay, and loss as well as the cached material at each node. The third is Nodes supply to and request content from the network using the same format network is aware of information, not just addresses. Cognitive management decides on data storage, replication, and other attributes of a router. The last is DTN hides internal network information (routing, protocols, and name services).

2.2. History of Delay Tolerant Networks

The basic concepts behind the DTN architecture as it is today came from looking at how to extend the internet into interplanetary space [8]. This new network paradigm was started in 1994 by the researcher named Vinton Grey Cerf, and Adrian Hooke. They began thinking about how to extend the internet into an interplanetary network and use Internet-style communications for links to spacecraft. The first phase of the Interplanetary Internet (IPN) project ran for about four years and its main output was a description of the problems and a proposed architecture for a communication overlay network that would support the transmission of messages in the IPN environment. This was published in 2001 as IPN: Architectural Definition [9] and set the architectural basis for much of the DTN work that has taken place since that time. The scenarios that the IPN architecture targeted were based on the sorts of extensive delays resulting from interplanetary distances and scheduled communication opportunities that are typical of spacecraft operations.

Following the release of the IPN document, academics started to think about how the architecture might be used in other scenarios where communications were exposed to lags and interruptions that would render traditional Internet protocols, particularly TCP, ineffective. The architectural work in 2002-3 looked at other scenarios, especially terrestrial wireless networks such as wireless sensor networks and Wi-Fi-based local area networks, where communications opportunities were much less predictable. It also provided a framework for dealing with interconnected heterogeneous networks, such as occurs at the gateway between a sensor network which does not usually use IP-based addressing/communications, and a conventional IP-based network. By the middle of 2002, when an updated version of the IPN architecture was published [2], Kevin Fall of Intel Research had coined the name Delay-Tolerant Networking (giving the initial use of the acronym DTN).

2.3. Delay-Tolerant Network Architecture

A communications network with substantially longer delays, connection interruptions, and communication latencies can be supported by a delay-tolerant networking architecture. In the beginning, the DTN architecture was suggested as a method for making the Interplanetary Internet [10] a workable networking environment. It sought to put in place a global network of networks. The bundle layer's forwarding capabilities regarding custody transfers, name, addressing schemes, and the store-and-forward capabilities of intermediary nodes were therefore given more attention. The author of an early study [9] discusses crucial difficulties with the creation of network protocols for difficult environments and suggests that the DTN architecture should take the form of an overlay that would cover all potentially difficult situations. A large number of the DTN's capabilities are driven by heterogeneity. The architecture in [11] is designed to support a more extensive form of ‘internet’ where the authors use the original meaning of the word that implied an interconnection of networks. The network elements are not expected to have common addressing formats or addressing semantics, and disparate routing methodologies may be employed.

The architecture of DTNs has been exploited in many works and its architectural layers compared with the conventional TCP/IP (internet) layers [12]. DTN layers and internet layers are identical except that DTN architecture implements store and forward message switching by overlaying a new protocol layer called bundle layer. It is placed on top of transport layers and it ties together the transport layers and below, with application programs to communicate across multiple regions. The bundle layer is proposed to function above the existing protocol layers and provide the function of a gateway when two nodes meet each other. Due to its flexibility, it can be easily linked with the already existing TCP/IP protocol networks or link two or more networks together. The difference between the TCP/IP layer and the DTN layer is shown in Figure 2.1.

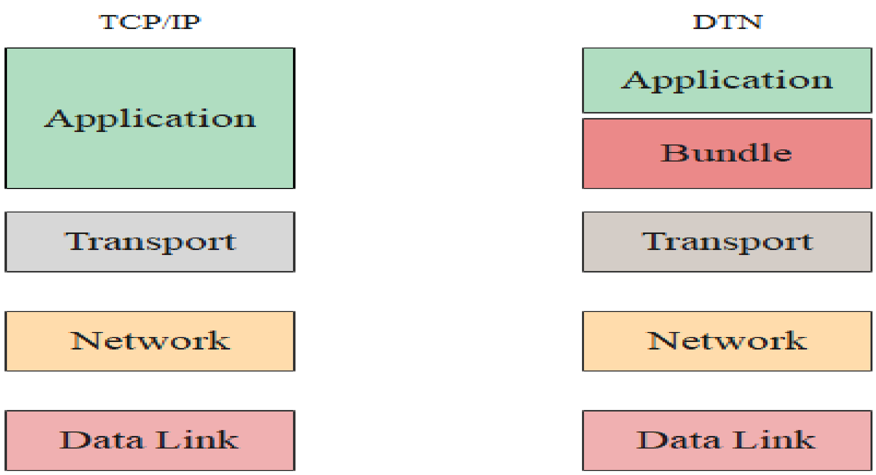

Figure 2.1: Difference between TCP/IP and DTN Architecture [3].

A DTN overlay network is produced by DTN, which functions on top of the current link layer and network protocols. The main advantage of DTN over IP is that it can join networks with a wide range of characteristics since it can handle extended delays, large error rates, and protracted link outages.

In-network data storage, retransmission, custody transfer with authenticated forwarding, and variable node naming are the main features offered by the DTN architecture. In order to use different delivery protocols, Figure 2.2 depicts an example implementation architecture that demonstrates how a bundle forwarder interacts with storage, routing choices, and convergence layer adapters (creating the convergence layer) [13].

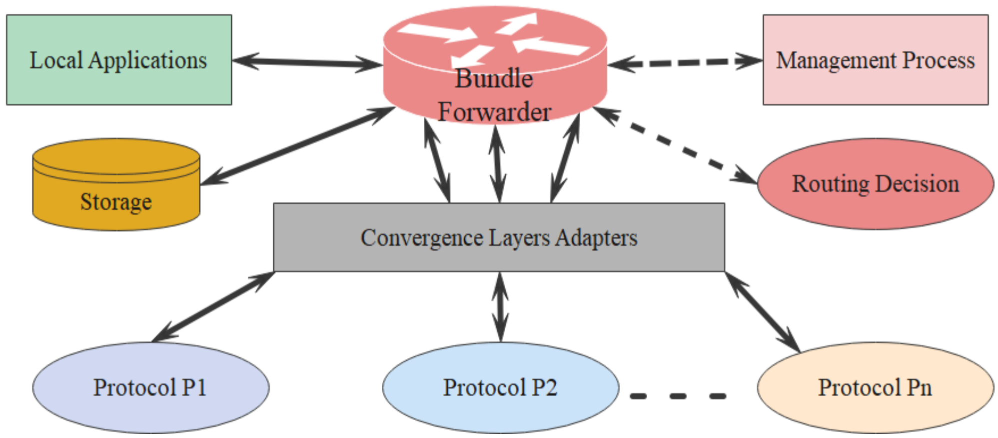

Figure 2.2: The DTN Architecture Implemented within a Single Node [13].

The bundle forwarder is at the center of the system that controls how bundles are managed within the node, as depicted in Figure 2.2, which is cited in [3]. The node can connect to other nodes during communication utilizing a wide variety of delivery methods, including as TCP/IP, Bluetooth, Wi-Fi, or hand-carried storage devices. A group of convergence layer adaptors (CLAs) translates the capabilities of the various protocols to the tasks required to transfer bundles to a peer node during communication opportunities, hiding the conflicting semantics of the various protocols from the bundle forwarder. The bundle forwarder will start and manage a link to the peer node using the relevant CLA when a communications opportunity arises, either because it has been planned by the node management process or because a peer node has been discovered, often by some wireless mechanism. In accordance with the routing decisions made by the routing decision process, the link will be used to transmit some of the bundles that are stored on this node to the peer node. Before being forwarded, the bundle may be encrypted or given integrity protection, as required by the policy [12].

2.4. Characteristics of Delay-Tolerant Networks

DTNs have some unique characteristics as compared to other forms of wireless networks such as MANETs or cellular networks. Some of the differences are mainly in terms of:

- **Intermittent Connection-** DTN gets disconnected frequently due to limited energy and node mobility thus continuously changing the DTN topology [14]. DTN doesn't guarantee to achieve an end-to-end route by keeping the track of its intermittent connections and the area where no path exists.

- **High delay, low efficiency, and high queuing delay-** The delay is comprised of waiting time, queuing time, and transmission time. It is the summated value of the total delay of each hop on the route. Due to intermittent connection, hop delay might be very high. Further, it leads to slow data rates and frequent fragmentations. It leads to increased queuing delays [15].
- **Limited resource-** Due to constraints on node's resources like processing power, storage, battery life, etc., the packet loss rate is very high.
- **Limited lifetime of a node-** Since a node has limited battery life, in special circumstances of the restricted network when the power is off the normal course of action is not possible. A message transferred at this time is generally lost.
- **Dynamic topology-** Due to high network mobility, the topology changes as nodes enters different regions or areas [16].
- **Poor Security-** Nodes in DTN are vulnerable to attacks like eavesdropping, message modification, routing spoofing, Denial of Service attack (DoS), and other security threats. A node may get corrupted due to a lack of specialized services and maintenance in the real world. In some cases nodes themselves deny forwarding the message, further restricting the transmission of a message from the source to the destination.
- **Heterogeneous interconnection-** DTN may be a heterogeneous network. The interconnection of such devices may require special arrangements at various stages of deployment [17].

2.5. Application Areas of Delay-Tolerant Networks

DTNs span a wide spectrum of application settings. As noted in [7] the study shows that DTNs have been applied to various settings such as interplanetary networks (IPNs), underwater networks, wildlife tracking sensor networks, and networks for developing communities. The application of DTNs which is referred from [18] is shown in Figure 2.3.

As described in [19], DTN improves network access and the distribution of educational and entertainment content in rural environments, despite the relative underdevelopment of more traditional kinds of infrastructure like roads and electricity. Thus, DTN does compromise network reliability and quality of service by introducing unpredictable delays as data is physically relayed from one wireless device to another.

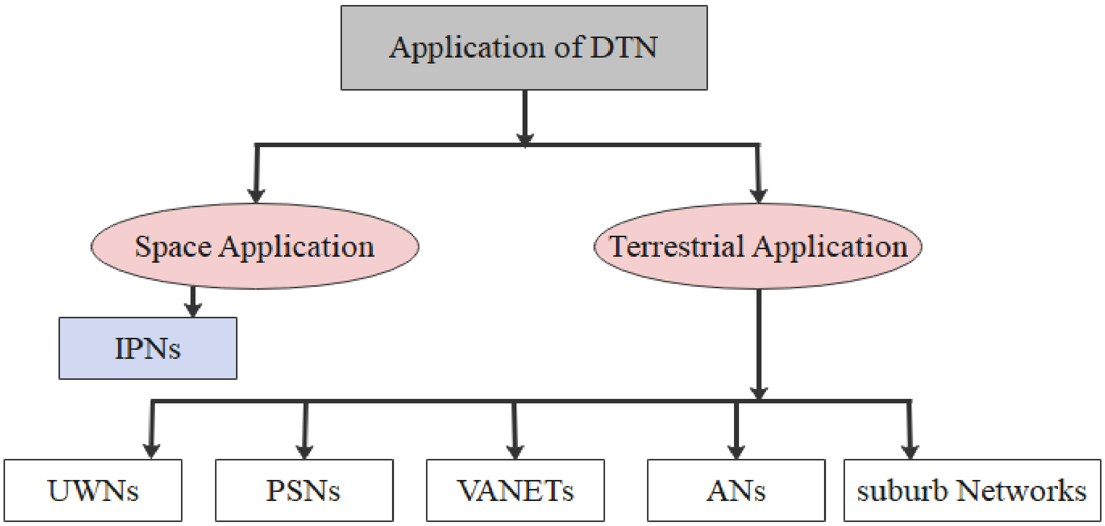

Figure 2.3: Applications of DTNs [18].

2.6. Connectivity in Delay Tolerant Network

In DTN applications, message transmission only happens when the right relays get in touch with one another. When the separation between two nodes is less than their communication range, contact occurs. The primary criterion for assessing mobility in DTNs is the amount of time between such two sessions. The meeting time is the length of time it takes for a node to first connect with another node. Two nodes begin processes to exchange messages as soon as they are within communication range of one another. The length of time they will communicate data depends on how long they are in each other's communication range. The contact time is known as this period of time. They must wait until they are once again within communication range if they do not complete the message exchange during the contact time. The inter-meeting time is the name given to this period of time. Different types of contacts between network nodes are defined by the DTN architecture. Request for comments of DTN describes five types of contacts [20], as follows:

- **Persistent Contacts:** Consistent contacts are constantly accessible (i.e., no connection initiation action is required to instantiate a persistent contact). An example of this type would be a "always-on" Internet connection, such as a Digital Subscriber Line (DSL) or Cable Modem connection.

- **On-Demand Contacts:** On-Demand contacts need to be initiated, but once they are, they remain active until they are terminated. An example of an on-demand contact is a dial-up connection.
- **Intermittent - Scheduled Contacts:** An agreement to establish a contact at a specific time and for a specific length of time is known as a scheduled contact. A connection to a low-Earth orbiting satellite is an illustration of a predetermined interaction. The satellite's schedule of view hours, capacities, and latencies can be used to create a node's list of interactions with the satellite.
- **Intermittent - Opportunistic Contacts:** Opportunistic connections don't show up on schedule; instead, they appear suddenly. Unscheduled aircraft passing overhead while beaconing to signal its communication readiness is an excellent example of this kind of interaction. An infrared or Bluetooth communication link between personal digital assistants could be another method of opportunistic contact.
- **Intermittent - Predicted Contacts:** The times and lengths of predicted connections are not based on a set timetable, but rather are forecasts based on a pattern of previously observed contacts or some other data. If there is a high enough degree of confidence in a predicted contact, routes can be selected using this data. We took into account the anticipated contact duration for the suggested solution based on the contact kinds covered above.

2.7. Mobility Model

Communication is typically made easier by the nodes' mobility in a delay-tolerant network because it is through their movement that formerly unreachable nodes link. Messages are sent to initially unreachable nodes by either waiting for a direct contact opportunity or by passing the messages through reachable nodes with the aid of DTN routing algorithms. Mobility models show the location, velocity, and acceleration of a mobile person as well as how these variables vary over time. The movement of the network nodes affects how well delay-tolerant networks perform (DTN). Practical traces and certain synthetic theories that aim for realism are used to construct mobility models. The underlying mobility patterns strongly influence the number, length, and other aspects of contacts. Therefore, understanding the statistics of these three values is necessary for a meaningful performance study of any mobility-assisted routing method. The most popular mobility models in delay-tolerant networks are discussed here, according to [21].

2.7.1. Random Mobility Models

Many traditional mobility models are still routinely used to enable straightforward DTN protocol evaluation. Node mobility models that are synthetically created can be adjusted in a variety of ways, although they typically only address a few mobility features. Random Mobility Models are the most basic conventional models. There are numerous such mobility models in the literature. Among them, the following popular models are discussed.

- **Random Walk (RW) Model** A simple mobility model that relies on random speeds in random directions.
- **Random Walkway Point (RWP)** A model that adds pause times when nodes assume a new destination and speed.
- **Random Direction (RD) Mobility Model** A model that drives nodes up to the boundary of the simulation area before changing direction and speed.

2.7.2. Map-Constrained Mobility Models

Another mobility model class with the random pattern confined within the map was developed through applications of the synthetic models on the map. These models use a portion of the Well Known Text (WKT) format to include map data. Geographic Information System (GIS) software typically uses the ASCII format, on which WKT is based [30]. These models are categorized as Map-Constrained Mobility Models. The following are some of the popular map-based mobility models.

- **Map-Based Mobility Models (MBM)** A model that moves nodes with random speed and directions following a map.
- **Shortest Path-Based Map Mobility Model (SPBMM)** A model that is similar to MBM, but selects a certain destination in the map for all nodes and finds the shortest path to reach the destination.
- **Route-Based Map Mobility Model (RBMM)** A model that has predetermined routes on the map that they travel.

2.7.3. Social Mobility Models

In many DTN applications, wireless nodes are carried by humans, animals, vehicles, or planets which possess their movement patterns. DTN routing protocols often exploit these patterns to implement applications [36]. In the human community, people meet strangers by chance or familiar strangers due to having similar mobility patterns while family, friends, and colleagues by intention. Therefore, human mobility pattern also has characteristics like macroscopic structure along with microscopic mobility properties. Thus, many researchers have developed mobility models based on human behavior. We have grouped these models as Social Mobility Models. We have analyzed the following popular models.

- **Community-based Mobility Model (CMM)** A model that is directly driven by a social network.
- **Working Day Mobility (WDM) Model** A model that accounts for all activities done by a human during a working week.
- **General Social Mobility Model (GeSoMo)** A model that separates the core mobility models from the structural explanation of the social network.

2.7.4. Composite Mobility Models

Since mobility models can be configured on a per-node basis, it is possible to combine multiple mobility models in one simulation. Consequently, researchers have been motivated to create mobility models where, for example, in a shopping Centre or a park some nodes move the following map along roads, and others walk around randomly within. We have included such mobility models in the Composite Mobility Models group. The following are the very familiar and successful mobility models of this type.

- **Self-similar Least-Action Walk (SLAW)** A model that accounts for all the five statistical features as well as user-created virtual ones.
- **Post-Disaster Mobility (PDM) model** A mode that mimics real scenarios and focuses on role-based movements based on a map.
- **Contact Graph-Based (CG) Mobility Model** A model that involves tools and metrics from social network analysis and graph theory.

2.8. Routing Protocol

The Inter-Planetary Network (IPN) launched a project for establishing connectivity between randomly located nodes on different solar system planets. Then they realized IPN is a special case of challenging networks where due to excessive propagation delay and frequent disruption traditional routing protocols failed. A significant amount of work has been done by researchers in the area of DTN routing. Most of these algorithms collect the knowledge of node encounter, duration of the encounter, and a relative location of nodes for better relay selection and try to reduce overhead in the network. In this section, we will discuss the major body of the work done for DTNs routing by categorizing it under four categories. This section specifies previous work done in the DTN routing algorithm based on information they used for relay selection. The following figure shows the categorization along with sub-categories of the DTNs routing protocols diagrammatically. Now, we will discuss some of the prevalent routing algorithms in each of the categories in detail.

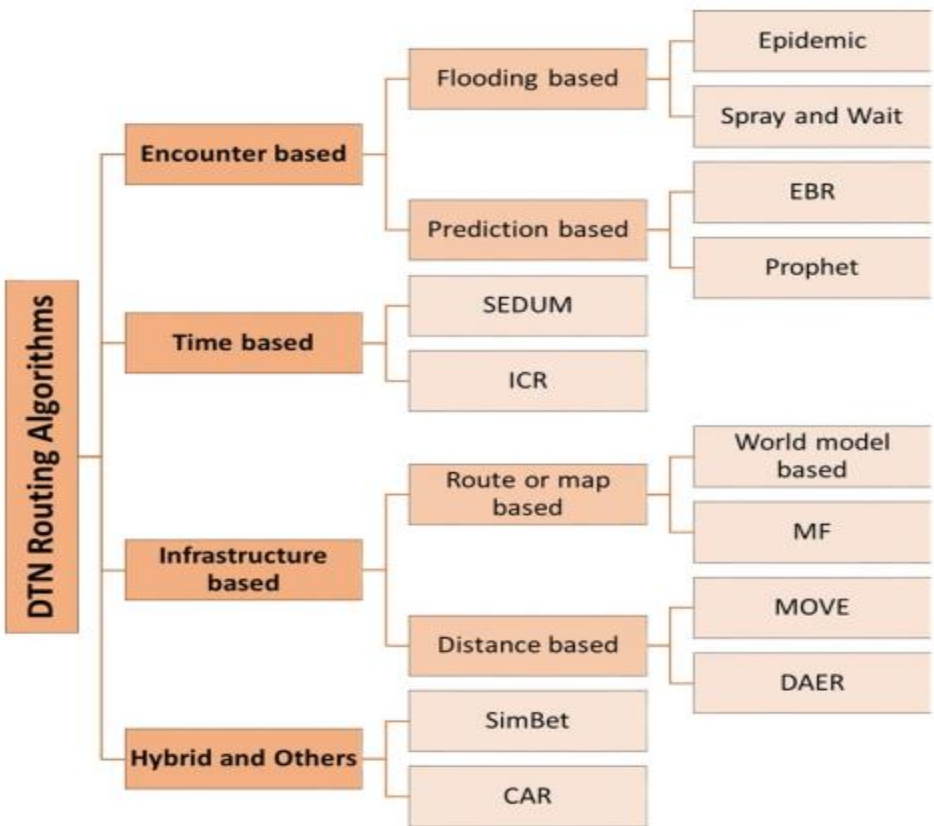

Figure 2.4: Categories of DTNs routing protocols [23].

2.8.1. Encounter Based Routing

The term "encounter-based routing protocol" refers to a group of protocols that forward messages based on the history of previous encounters between nodes. Two nodes are said to have encountered one other when they are within range of one another, and the node keeps track of these encounters in its history. This category can be further subdivided as:

- **Flooding Based Routing-** Flooding based category defines the routing protocols when there is zero knowledge about the encounter history. This means these protocols do not use any strategy for relay selection [24].
- **Encounter Prediction Based-** this category defines the relay selection prediction based on the encounter history. Thus, these protocols harness the encounter behavior of the nodes to determine the best nodes to forward the message further. Some of the seminal work in these sub-categories are discussed below [25] [26].

2.8.2. Time-Based Routing

This category defines the relay selection prediction based on any time-related metric such as the interval, contact duration, inter-meeting time or inter-contact time, etc.

Social network-oriented and duration utility-based distributed multi-copy routing protocol achieve high throughput in a dynamic setting as instead of using only contact frequency for the delivery utility in probabilistic routing, it also considers cumulative contact duration within a time period [27]. Contact frequency f of node A to a node B is the number of times they encountered over a time period T.

The author in [28] proposed a seek-and-focus protocol, which is based on the latest encounter time. This protocol initially seeks a relay node based on delivery utility and then moves to the focus phase if a better relay node with the latest encounter time for the destination node comes into range. In Space and Time routing protocol [29] next-hop selection is based on the information of the current and future neighbors. Further, in this protocol, a space timing graph that depicts the mobility of the nodes is used.

2.8.3. Infrastructure Based Routing

This category represents the routing protocols, which use the infrastructure information for routing decisions. These protocols used location information of nodes, routes they follow, map information, moving direction of nodes, etc. for better decision power to decide on forwarding node. Now, we will discuss the algorithms developed based on distance and route information [30] [31] [32] [33] [34].

2.8.4. Hybrid and Other Routing

The protocols are under the hybrid and other categories since they are different from those under the listed categories and use data from more than one category. Predict and relay (PER) routing calculates the node's movement activities and fills the transition matrix with the likelihood that it will visit a particular location [35]. The utility metric for calculating the likelihood of arriving at the destination is calculated by PER using a time probability distribution matrix. To provide a novel reputation-based incentive strategy for routing, the authors of [36] have done so. Disobedient nodes are identified and excluded from DTN routing in this proposed scheme. Epidemic forwarding and Delegation forwarding are the two versions they created. In epidemic forwarding, messages are transferred to primarily encountered nodes and in Delegation forwarding, messages are transferred according to the node's forwarding capabilities.

Simbet [37] is a protocol that applies similarity and betweenness properties for relay selection. Simbet falls in this category because between-ness is a property of distance (shortest path) and similarity measures the common neighbors. When a node A interacts with another node B, A calculates its relative similarity SimUtilA(D) and betweenness utility BetUtil A to the node B for message delivery to the destination node D.

2.9. Buffer Management Policy

DTN systems are often composed of resource-constrained wireless or mobile devices. So, when configuring a delay tolerant network (DTN), there are many aspects that need to be taken into consideration for an effective and efficient network. One aspect is a buffer management strategy. Since the buffer space and internode wireless resources in a system are always limited, it is important to enforce buffer management policies to maintain message delivery in DTNs. The goal of an efficient buffer management algorithm can be summarized as deciding in which order the messages should be forward within limited contact durations and deciding which messages should be discarded when the node's buffers operate close to their capacity. The buffer should be managed properly since it may be full due to the store-carry-forward property. In such a circumstance, some messages stored in the buffer should be dropped to hold the new message coming in.

The internal architecture of a DTN node is shown in Figure 2.5. Upon contact, a node exchanges a summary of the information necessary to update its knowledge on the environment in which it operates. As an example, it can exchange the list of messages in the buffer or the list of prior encountered nodes. This knowledge drives all of the subsequent actions and is used by buffer management to assign a rank to the messages in the buffer. If the buffer is full, buffer management applies a suitable eviction policy to make room for new incoming messages. The forwarding policy selects the messages to be forwarded, which are possibly duplicated to improve the effectiveness of the protocol.

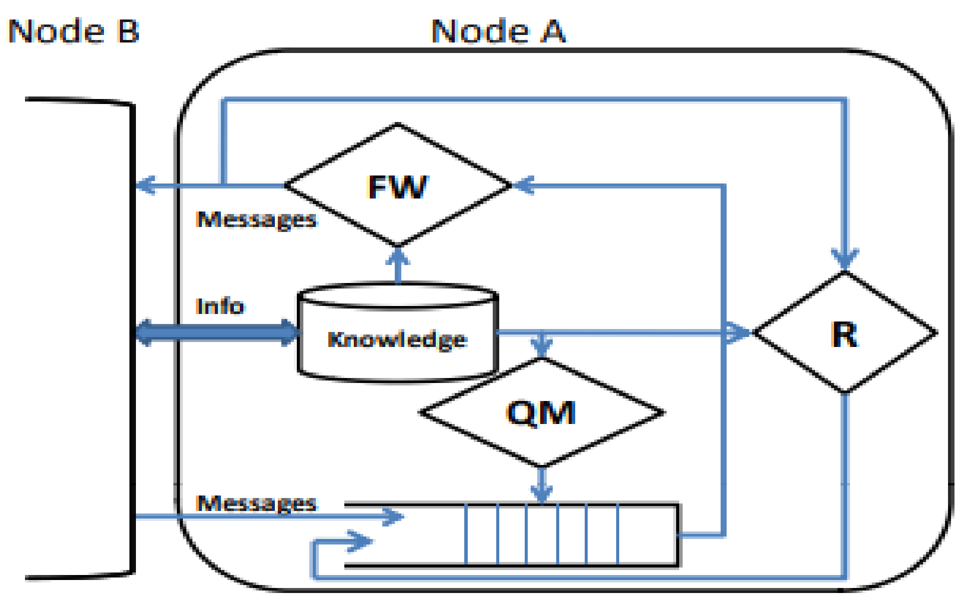

Figure 2.5: Node architecture [38].

2.9.1. Buffer Management Approaches

Buffer management policy defines a total ordering of the messages in the buffer based on the node knowledge. Buffer management orders all of the messages, also those that will not be taken into consideration as possible candidates to be forwarded. When new messages must be accommodated in the buffer and there is no room, the eviction policy selects the messages to be discarded according to that ordering. The intrinsic delay of DTN applications implies that messages can be enquired for a long time. Due to the limited buffer size of DTN nodes, an efficient queuing management policy must be adopted to avoid the eviction of important messages. In some cases, more than one policy is combined to produce more efficient message ordering [38]. In the following, we briefly discuss the main buffer management policies:

- First-In-First-Out (FIFO) is the default buffer management policy.
- Destination independent, where uses only parameters that are not related to the destination, such as message hop count, forward transmission count, and message size, and other types.
- Destination dependent: In this case, buffer management uses parameters dependent on the destination.

Forwarding- selects the subset of messages in the buffer to be forwarded. In most cases, the messages are selected on the basis of the same ordering defined by the buffer management policy, but in some cases, a new ordering among messages can be employed. Contrary to what happens for buffer management, this new ordering is defined only with respect to the current contact and lasts only for the contact time. In other words, the new ordering is only used to define a priority on the messages to be forwarded to the current contact. Briefly, the main forwarding policies are:

- Direct-delivery: The message is only delivered to the destination (i.e., no intermediate nodes are employed).
- Always: Upon contact, the message is always forwarded. This policy requires low computational power but can lead to a very high number of transmissions.
- Knowledge-based: Messages are forwarded based on the current knowledge in terms of contextual, historical, or social information. Contextual information regards the knowledge that can be inferred by the node only based on its current status, such as battery level, speed, direction, etc. Historical information is obtained over time and is used to evaluate future behaviors of the network; inter-contact time, history of encounters, age of encounters, and contact time are examples of such information. Finally, considering that in many DTN scenarios, nodes are carried by humans, social information is used to describe the relationships among the users in order to predict social behaviors that can be used to improve the effectiveness of the forwarding policy.

Replication- controls, and bounds the number of copies of a message in the network and is used to increase the robustness of the protocols. Observe that a message that has been selected by the

FW policy to be delivered to the current contact is normally deleted from the queue, but the R policy can queue it again in order to have multiple copies in the network.

- Single copy: Messages are never replicated. Once a message has been delivered to an encounter, it is deleted from the queue.
- Limited: The total number of copies of a message in the network is bounded.
- Controlled: A message is replicated only if a condition holds.
- Unlimited: There are no constraints on the number of replicas in the network.

2.9.2. Classifications of Buffer Management Policy

Multiple buffer management strategies have been proposed to match different protocols. Some of them are general strategies that can be used for any routing protocol, while some strategies can be used for specific routing protocols. In addition, according to their policies on dropping messages, these strategies can use static or dynamic algorithms to select which message should be dropped. The difference between static and dynamic algorithms is whether they need to gather information at runtime.

2.9.2.1. Protocol-Specific and Non-Protocol-Specific

Protocol-specific buffer management techniques are those that were created specifically for a given routing protocol or that won't function properly under any other routing protocol. Non-protocol specific buffer strategies are those that can be utilized with any protocol or perform well with any protocol. Customized design can be used in protocol-specific techniques to take into account the particulars of various protocols. Benefits include a higher ratio of deliveries and shorter delivery delays.

2.9.2.2. Static Dropping and Dynamic Dropping

As was previously said, when the buffer is full, some messages should be deleted to make room for incoming messages. In this context, a static buffer strategy is a strategy that uses no data from the present environment to decide which message should be deleted. A dynamic buffer strategy is one that needs to acquire data in order to determine the best method for dumping messages. It is clear that static dropping can require less processing time. Dynamic dropping, however, might gather additional data for a more precise judgment.

2.10. Challenges in Delay Tolerant Network

Based on the likelihood that, out of multiple copies of a message, one of them will reach a destination, routing protocols developed for DTN are based on this assumption. A distance function is used to measure the cost of transmitting communications from one place to another. The main difficulty in DTN is routing because it only requires a little knowledge of the network to do so. Some constraints are taken into considerations.

Resource allocation:- The primary objectives of the traditional routing system are to maximize message delivery and reduce resource usage. Replicate copies of messages to all network nodes to maximize message delivery. Each intermediary node must store messages on all hosts since DTN cannot ensure that a message will reach its destination. As a result, more resources are used.

Buffer space:- All messages must be stored by the intermediate nodes in the allocated buffer space. More buffer consumption will result from more pending messages (which are not being sent to the destination). DTN networks must preserve all messages until they reach their destinations because disconnection may happen often.

Limited power:- Due to sending, receiving, storing, and calculating activities, routing in DTN results in a higher energy resource use. Therefore, it is recommended to use an energy-efficient routing protocol. More energy will be used by nodes that are mobile and disconnected than by nodes that use the standard routing method.

Contacts available:- Nodes in a network make an effort to connect when advantageous contacts are available. DTN has a connectivity issue, therefore transmitter and recipient can only communicate for short periods of time.

Securit:- Security is a crucial concern for both DTN and traditional networks. Users must demand a particular level of assurance regarding the validity of messages because messages are traversed among an arbitrary path of hosts in the network. Here, different cryptographic methods are used to guarantee secure end-to-end routing.

CHAPTER THREE

RELATED WORK

As described in the literature review the effective operation of delay tolerant networks depends on the buffer management policy among pairs of network nodes to exchange messages within sparse and challenging network environments. Since the network resources such as node buffer size, network bandwidth and contact duration are always restricted conducting efficient buffer management at each forwarder node is necessary to improve the performance of the networks. There are several forwarding policies have been proposed in buffer management to enhance the performance of DTNs. In this chapter, we present the forwarding strategy of recently introduced buffer management policies related to our work. These policies are characterized under two main categories based on the information they considered to forward intend buffered messages.

3.1. Heuristic-Based Policy

The heuristic-based policies employ different states of a message to estimate the probability that a message can be delivered to its destination. Some of the message states used in this policy are, the hop counts of a message, the number of message copies, remaining time-to-live (TTL), and the duration that a message has been kept in a node. The authors in [39] proposed an efficient buffer management scheme for DTN routing, particularly in the context of real DTN deployment. The study provides the scheme comprises intelligent decisions for message transfer and message drop for buffer management. According to the scheme the messages which have the lowest hop count are scheduled to be transmitted first. This is because a low hop count depicts that the message hasn’t traveled far from the source and is still far from the destination. Therefore, such messages must be prioritized for transmission to create more copies so as to achieve a higher delivery rate and lower delivery delay. In the scheme, both components of buffer management policy are considered to determine message forwarding and dropping decisions. Regarding message forwarding strategy only the hop-count of a message is considered to schedule buffered messages to be forwarded when the node encounters other nodes.

Also, the other authors in [40] proposed a weight-based buffer management policy for DTN routing protocols. The introduced scheme by the study is used to effectively manage the buffer of nodes in the network whenever buffer overflows happen. According to this scheme buffered messages in the node are assigned to one of three queues; the priority message queue, the high-weight message queue, and the low-weight message queue. Among these queues, the most recently received messages are put in the priority message queue. When buffer overflows happen, the messages in the priority queue will never be discarded to avoid discarding the latest received messages. In addition, when buffer overflows happen, if the node is not the new message's destination node, messages in the high-weight message queue are discarded. If all the messages in the high-weight queue are discarded and the node buffer still overflows, the new message will not be received. If the current node is the destination node of the new message and the sum of the size of the free buffer space and the buffer space occupied by the messages in the high-weight queue is still less than the size of the new message, messages in the low-weight message queue are discarded until the buffer space is sufficient to receive the new message. Besides the message forwarding strategy, the study considered only the duration that the message has been kept in the node. Accordingly, the messages stored in the priority message queue have a high priority to be forwarded when the node encounters the other appropriate nodes. The study contributes to how to take care of global information regarding the network when dropping the message and the effects of discarding the message on the performance of the network. Moreover, the study proposed scheme mostly considered dropping the policy of buffer management, which is intended to address buffer overflow.

In another way, the authors in [41] introduced priority-based buffer management techniques for opportunistic networks. As the authors suggested message priority allows for multiple messages to have their corresponding types based on the nature of these messages. The message type is set according to the confidentiality of the message. When the sender transmits the messages to the destination, the message type is constructed during it. Also, the relay nodes follow the same pattern for buffer management that is, allowing high-priority messages to be transmitted before the lower ones. When a source node transmits a message to the destination, it, firstly, transmits to the neighboring nodes in its proximity based on message priority. The intermediate nodes, when exchanging the summary vectors, queue the messages based on the message priority in their buffer. On the opportunistic encounter of other intermediate nodes for the exchange of the summary vector, the most prioritized message is transmitted based on message priority. Thus, the destination node finally receives the most prioritized message. The primary advantage of this study is that, without any losses, the messages reach the destination faster, based on the message priority assigned initially to these messages. Also, it's more secure and efficient than traditional buffer management policies for opportunistic networks. However, the proposed scheme in the study considered only message confidentiality to prioritize buffered messages to be forwarded.

3.2. Optimal-Based Policy

The optimal-based policies commonly formulate the message scheduling and dropping task as a message replica allocation problem concerning delivery ratio or delay. The author in [42] proposed joint scheduling and buffer management policies for DTN applications of different traffic classes. According to this scheme first of all average delivery rate and average delivery delay calculates for each buffered message by considering message states and network knowledge. Therefore messages residing inside a nodes buffer can be separated into two dynamic groups; the first group contains all messages whose predicted delivery probability or delay hasn’t reached the desired quality of service threshold and the second group consists of messages which have reached their threshold. In the case of delivery delay optimization, the second group includes also the messages whose elapsed time since creation is higher than the desired threshold. The messages of the first group are always prioritized over the messages of the second group. Regarding message forwarding decisions in this scheme, the classic utility function is used to schedule the buffered message. So, the message which has a high utility function value has a high priority to be forwarded. This approach guarantees the satisfaction of the individual constraints, when this is feasible given the available resources, and allocates any remaining resources optimally, to maximize the desired performance metric.

Also, the author in [43] introduced a max delivery approach to DTN buffer management. In the study, a specific policy for buffer management is implemented, which is designed by considering contact history and message states to schedule and drop the messages. According to this work, each node has three priority queues prepared for the transmission of messages. The first queue contains messages that have the contact node as a destination. This queue is sorted in descending order by message size and it has the highest transmission priority. The second queue contains messages that belong to a neighbor of the contact node. This queue has the next level of priority. The messages are sorted too in descending order by size. The third queue contains the rest of the messages in the buffer, sorted by a utility function which is calculated by considering message states. Besides the forwarding strategy of the scheme, node contact history and message size are considered. Generally, this study contributes an intermediate phase of an algorithm that optimized the message transfer rate and resource consumption in DTN networks.

In another way, the authors in [44] proposed a buffer management policy based on the congestion control metric in vehicular delay-tolerant networks. Also, this approach is intended to address congestion in the node buffer to maximize the average delivery ratio and minimize the average delivery delay. The metric takes into account several attributes of nodes and messages, including the remaining lifetime of messages, delivery probability, and buffer overhead ratio. Also takes into account the encounter probability between nodes and the number of copies of the message in the forwarding decision. This policy combines the delivery successful estimation, time measurement, and buffers overhead ratio to calculate the congestion control metric value. Based on the value, the proposed scheme efficiently controls the message drop policy to reduce the impact of message drop on the delivery ratio. The study mostly considered the dropping component of buffer management, which is not enough to achieve needed performance without giving attention to forwarding policy.

In addition, the author in [45] proposed optimizing bulk transfer size and scheduling for efficient buffer management in mobile opportunistic networks. The study first developed a generalized probabilistic forwarding model where the forwarding probability denotes the likelihood of a message to be forwarded to the encountered node. Based on the forwarding probability, develop a congestion indicator and predict the point of congestion. Using this, a node can decide the optimal number and the exact set of messages to replicate, which leads to an optimal performance with minimal message drop and overhead. According to this scheme, the messages are optimally scheduled, first by calculating the average delivery probability of the individual messages in the buffer and then maximizing the average delivery probability of all the messages subject to the buffer overflow constraint. Using this, a node can decide the optimal number and the exact set of messages to replicate. The approach optimized performance with minimal packet drop and overhead.

In the following table, recent works related to our work are discussed with their forwarding mechanism, contributions, and limitations.

Table 3.1: Summary of related work.

Year	Forwarding Mechanism	Contribution	Limitation
2016 [39]	Message hop count	Scheduling and dropping policy	Lowest hop count message scheduled to be transmitted first without considering available contact duration which leads to partial message transmission
2017 [40]	Duration the message has kept in the node	Take care of global information regarding the network when dropping the message and the effects of discarding the message on the performance of the network	A recently received message has high priority to be forwarded but complete message may not transmit due to limited contact duration between pair of nodes.
2018 [42]	Message states and inter contact time	Derive a distributed algorithm that adapts to the available resources and provide a framework to applying this algorithm in real life mobility conditions	At a time all message items may not reach the destination node but the node waits till to receive complete message which leads to more delays
2019 [41]	Message type based on confidentiality	Without any losses, the messages reach the destination faster Secure and efficient	The message which needs high confidentiality has a high priority to be forwarded but the complete message may not transmit to the requester node within contact duration which leads to partial message transmission
2020 [43]	Node contact history and message states	Optimized the message transfer rate and resource consumption	The message intended to direct contact node prioritized by descending order by message size to forward first but within available

			contact duration complete message may not transmit
2020 [44]	Encounter probability between nodes and the number of copies of the message	Buffer management policy based on the congestion control metric	A complete message may not transmit to the requester node due to limited contact duration but the receiver node drop some of the buffered messages to create room for incoming message
2021 [45]	Message states and inter contact time	Accurate prediction of the point of congestion Devise a proactive buffer management policy that incurs minimal overhead Assess the impact of mobility parameters on the network performance	The messages are forward to the node which has less inter contact time with the destination node but waiting till to contact with such nodes leads to more delay.

According to the above analysis of existing buffer management forwarding policies, we find that most current studies are considered the message states and network knowledge to schedule buffered messages to be forwarded. However these studies are not considered limited contact duration between pair of nodes whenever the node forwards the message, therefore blindness exists.

In this paper, we proposed a contact duration aware forwarding strategy for buffer management policy in spare delay tolerant networks. Our scheme determines the forwarding probability of buffered messages based on contact duration between pairs of network nodes. Therefore our proposed scheme easily integrates with the existing forwarding strategy of buffer management and addresses the problem of limited contact duration. We aim to forward only complete transmit messages whenever the nodes encounter each other, which is important to avoid inappropriate message drop and reduce the destination waiting time till to receive complete message.

CHAPTER FOUR

DESIGN OF THE PROPOSED SOLUTION

The contact duration between pairs of network nodes is limited due to node mobility in a delay-tolerant network. Hence always complete message may not transmit to the requester node within contact duration. The nodes require to wait till they come into communication range again when do not finish exchanging the message during the contact time. In sparse delay-tolerant networks, the pairs of network nodes take a long time to contact again. Thus proper management of contact duration is important to enhance the performance of networks. Due to limited contact duration, there is partial message transmission which is make inappropriate message drop and destination node waits till to receive complete message. Therefore further investigation needs to be conducted to reduce destination node waiting time to till to receive a complete message and avoid inappropriate message drops due to partial message transmission in sparse delay-tolerant networks.

The existing buffer management forwarding approaches are mainly considered message states and network knowledge. However, these policies ignored the limited contact duration between pairs of network nodes. Thus complete message may not transmit to the requester node within the contact duration as prioritized by buffer management forwarding policy. In our new scheme, the specific problem of the existing schemes is considered. To address the problem we proposed a contact duration aware forwarding strategy for buffer management policy in sparse delay-tolerant networks. Since all intended buffered messages may not be exchanged between nodes within a single contact duration, our scheme forward only complete transmit messages when pair of nodes encounter. Accordingly, we can avoid inappropriate message drops and reduce the waiting time to receive a complete message at a destination node. Moreover, our goal is to determine the forwarding solution for each node to wisely use available contact duration.

In this study, we are going to add a contact duration aware forwarding module to the forwarding policy of the existing buffer management scheme. The new module is used to identify the number and orders of complete transmit messages within contact duration. However, our proposed solution does not modify the existing forwarding strategy of buffer management policy. Rather our proposed scheme considered the output of the existing forwarding strategy as an input. Also, operates at the top of the existing buffer management forwarding strategy. The proposed solution algorithm is run in every node in the network to manage limited contact duration between pairs of network nodes. In a delay tolerant network, when pair of nodes contact the existing forwarding strategy of buffer management policy selects and sorts intended buffered messages to forward. These sorted intended messages are input for our proposed scheme because the forwarding strategy of the existing buffer management policy has no information on whether the message is complete transmit or not within contact duration. The other new idea in the proposed solution is each node calculates and registers the current contact duration with the encountered nodes. Also, the sender node read past contact duration with the receiver node to identify types of available contact duration with the receiver node based on the intended buffered message size. Identifying the types of available contact duration is used to determine the number and order of complete transmit messages.

4.1. The Architecture of Proposed Solution

The proposed solution architecture shows the flow operations of the proposed work in detail. We proposed a contact duration aware forwarding strategy for buffer management policy to enhance the performance of sparse delay tolerant networks. The proposed scheme is used to select and forward only complete transmit messages within a limited contact duration. To do this the scheme perform two main tasks when pair of network nodes contact. The first task is calculate the current contact duration and register to use the next time when the nodes contact again. The second task is read the past contact duration with the encountered node and compare it with the duration required to transmit the smallest size message and all intended messages. These tasks aim to select and forward complete transmit messages within contact duration. According to the proposed solution architecture, we show that the intended messages are forward based on available contact duration. Our scheme forward only the complete transmit intended messages, if available contact duration is enough to complete transmit some intended messages. Also, reduce waste of the time and resource to transmit partial items of the message. The proposed scheme wisely uses the available contact duration between pair of network nodes. Therefore unexpected message drop is avoided and also waiting time to receive a complete message is reduced. Moreover, the performance of the buffer management policy is optimized in sparse delay tolerant networks.

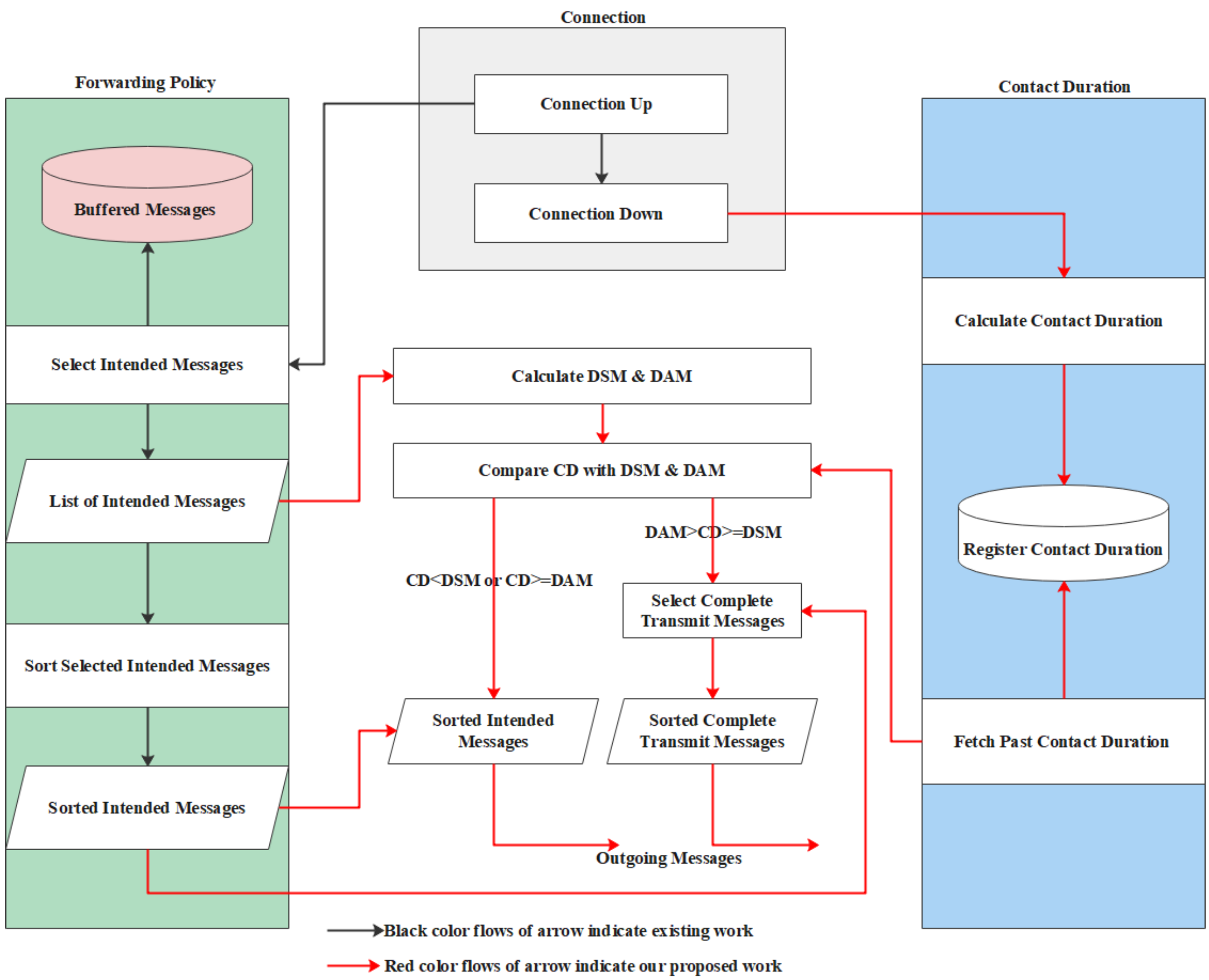

Figure 4.1: Architecture of the proposed solution.

In the following, we discussed the components which show the public information about the architecture of the proposed solution.

4.1.1. Connection

In delay tolerant network connection between pairs of nodes is not stable due to node mobility, dynamic topology, and network partitioning. Thus there are two states of connection between pair of network nodes. The first state is connection up, it refers to when the pair of nodes inter the same communication range. The existing buffer management forwarding policy select and sorts intended messages when the connection is up. However, our proposed scheme fetches past contact duration while selecting and sorting intended messages. The past contact duration is used to decide the number and order by which sorted selected intended messages are forward. The second state is connection down, it refers to when connected pairs of network nodes are left out of the same communication range. Our proposed scheme start to calculate and register the duration of time the two pair of nodes stayed in the same communication range.

4.1.2. Forwarding Policy

Buffer management policy has forwarding and dropping components. The forwarding policy is used to select and prioritize intended messages to be forwarded when pairs of nodes contact. All buffered messages may not forward because the same copy of the message may exist at the receiver node. Thus in DTNs, each node maintains a list of message IDs that it is currently holding in its buffer called the summary vector. When two nodes meet, they exchange their summary vector with each other. By comparing this summary vector, the sender node selects intended messages for the receiver node. After selecting intended messages, the messages are sorted based on different criteria to be forwarded to the encountered node. The existing schemes considered message states and network knowledge to sort the intended messages. However limited contact duration between pair of network nodes has an impact on the successful transmission of the message. Since the messages have an equal chance to transmit within contact duration, we cannot sort intended messages by considering contact duration. Therefore the new contact duration aware module needs to integrate with the existing forwarding queue of buffer management to select and forward complete transmit messages within contact duration. Our proposed scheme considered both selected and sorted intended messages by the existing scheme as an input. Therefore the proposed scheme operates at the upper layer of the forwarding queue to determine the number and order of complete transmit messages.

4.1.3. Contact Duration

In our proposed scheme we considered the predicted contact type between pair of network nodes. The predicted contact refers to predictions of likely contact times and durations based on the history of previously observed contacts or some other information. The past contact duration between pair of network nodes will occur next time when the nodes contact again. Therefore as a basic parameter, we considered contact duration in our proposed solution which refers to the duration that pair of network nodes stayed in the same communication range. A contact duration between pair of network nodes indicates the range of time from connection up to connection down. Also, contact duration is used to determine whether the message complete transmit or not when pairs of nodes contact. According to our proposed solution, each node in the network calculates and registers the duration stay with other nodes in the same communication range. Also, fetch the past contact duration with the encountered node to determine the number and order of messages to be forwarded. The contact duration between encountered nodes is calculated by:

Contact Duration = Connection Down Time – Connection Up Time

The contact duration is updated within contact frequency to handle some changes like node movement speed and direction. The calculated contact duration is registered by mapping with the hostname of encountered node. Thus when next time the nodes contact again easily identify the past contact duration with the connected node.

4.1.4. Calculate DSM and DAM

The proposed scheme considered the output of the existing scheme as an input to calculate the duration required to complete transmit the smallest size intended message and all intended messages. A message is complete transmit if contact duration is greater than or equal to message size divided by communication bandwidth. Therefore the duration required to complete transmit the smallest size message is calculated by:

DSM = Message Size / Bandwidth, where DSM is the duration required to complete transmit a smallest size intended message.

Also the duration required to complete transmit all intended messages is calculated by:

DAM = MS1 + MS2 + MS3 + - - - + MSn / Bandwidth, where MS is message size and DAM is duration required to complete transmit all intended messages.

4.1.5. Compare CD with DSM and DAM

According to our proposed scheme when pair of nodes contact each of them read the past contact duration they have before starting to forward intended messages. The duration require to complete transmit all intended messages and the smallest size intended message are compared with past contact duration between pair of nodes. The comparison is used to identify types of past contact duration. The identification of contact duration type is used to know the number of complete transmit intended messages within available contact duration. Also, the order of messages to be forwarded is determined by considering the type of contact duration between nodes. Besides message forwarding, when pairs of nodes contact again first of all sender node read the past contact duration with the receiver node. The past contact duration is classified into four types based on DSM and DAM values. The number and order of messages to be forwarded determined based on the type of contact duration. These types of contact duration are enough, not enough, and partial enough. Accordingly, the not enough contact duration refers to a contact duration that is less than the transmission duration of a small size intended message, then the first message is forward according to the existing forwarding strategy. Also, enough contact duration refers to contact duration is greater than or equal to all intended message transmission duration, then all intended messages are forward as prioritized by the existing forwarding strategy. Otherwise, partial enough contact duration is refers to contact duration is greater than or equal to the transmission duration of the smallest size intended message and less than the transmission duration of all intended messages, then only complete transmit messages are selected and forwarded by keeping their priority.

4.1.6. Select Complete Transmit Messages

Types of available contact duration are identified by comparing with the duration required to complete transmit the smallest size intended message and all intended messages. If available contact duration is greater than or equal to the transmission duration of the smallest size intended message and less than the transmission duration of all intended messages. Mainly in DTNs contact duration between pair of network nodes is enough to complete transmit some intended messages. Our proposed scheme starts to select complete transmit sorted messages by existing buffer management policy within contact duration. A complete transmit messages are selected by keeping its priorities and sorted again as selected. The message is complete transmit if the available contact duration is greater than or equal message size divided by communication bandwidth. Our scheme select complete transmit message starting from high priority to low as sorted by BMP. Accordingly, our scheme select the message if contact duration greater than or equal message size divided by bandwidth, and calculates the remaining contact duration to check the next message. The remaining contact duration is calculated by:

Remaining Contact Duration = Contact Duration - Message Size/Bandwidth

Jump and continue to check the next message if the available contact duration is less than message size divided by communication bandwidth.

In the following, we discussed the algorithm developed in this thesis to tackle the stated problem. The proposed work algorithm is run on the top of forwarding decisions to identify the number and order of complete transmit intended messages to be forwarded.

Algorithm: Our proposed work algorithm executes at the sender node when come in contact with the receiver node.

1. ***Forwarding policy module***
2. ***List of intended messages***
3. ***Procedure (Sorted intended messages)***
4. ***Begin***
5. *If CD is less than DAM & greater than or equal to DSM*
6. *For sorted intended messages M*
7. *Continue to iterate till to get complete transmit messages*
8. *If CD is greater than or equal to M Size / Bandwidth*
9. *Forward M*
10. *CD= CD - M Size / Bandwidth*
11. *End if*
12. *Else*
13. *Jump and continue the next M*
14. *End for*
15. *Else*
16. *Forward messages as sorted by BMP*
17. *End if*
18. *Stop forward*
19. ***End Procedure***

Forwarding policy module:- is used to select intended messages to the receiver node and sort the selected intended messages.

List of intended messages:- is the messages which are not with the receiver node prioritized by the existing buffer management policy to be forwarded**.**

If CD is less than DAM & greater than or equal to DSM:- when contact duration is partial enough, which means available contact duration between encountered nodes is only enough to complete transmit some intended messages.

For intended message M in buffer:- check each of the intended messages whether complete transmit or not within contact duration

If CD is greater than or equal to M size / bandwidth:- the current message can complete transmit within the available contact duration. So the current message forward and remaining contact duration is calculated to check the next messages.

Else Jump and continue the next M:- when contact duration is not enough to complete transmit the currently intended message jump and check the next messages.

Else Forward message as sorted by BMP:- when contact duration between encountered nodes is zero, not enough and enough the intended messages are forward as sorted by buffer management policy.

4.2. Summary

To come up the problem of limited contact duration, we proposed a contact duration aware forwarding strategy for buffer management policy in sparse delay tolerant networks. Our proposed solution considered contact duration between pairs of network nodes and prioritized intended messages by the existing strategy. The output of the existing strategy is used as an input for our proposed scheme. Therefore our scheme operates on the list of intended messages selected and prioritized by the existing approach. Since all intended buffered messages may not be exchanged between nodes within a single contact duration, our scheme forward only complete transmit messages.

CHAPTER FIVE

IMPLEMENTATIONS AND RESULT EVALUATION

5.1. Overview

In the previous chapter four, we have designed a contact duration aware message forward scheme for buffer management policy in sparse DTNs. The proposed scheme aimed to address the problem of partial message transmission due to limited contact duration between pairs of network nodes. Our proposed scheme aims to reduce destination node waiting time till to receive a complete message and avoid unexpected message drops. To achieve our objective we have proposed a contact duration aware message forwarding strategy, which is used to forward only complete transmit messages within contact duration. The order of messages to be forwarded between pair of network nodes is determined based on types of contact duration. If the contact duration between sender and receiver nodes is not enough and enough, the sender node forward intended messages to the receiver node as ordered by the buffer management policy. Otherwise, our scheme selects and forward only complete transmit messages by keeping the order of messages prioritized by the buffer management policy.

Due to the excessive costs of DTN entities and the wireless access network technologies in real-world test beds, our proposed solution has been implemented and evaluated using a network simulator. We have used the Opportunistic Networking Environment (ONE) simulator to evaluate the result of our proposed scheme with the existing approach. The ONE simulator implemented in Java and available as open source has been used to demonstrate the proposed system. The simulator supports for generating different node movement models, routing messages between nodes with different DTN routing algorithms, visualizing both mobility and message passing in real time in its graphical user interface. In our proposed solution simulation experiment, we have considered part of the city of Helsinki; Finland scenario of sparse DTN nodes. A detailed description of the implementation of our proposed work is presented under subsections of this Chapter. Section 5.2 describes the experimental environment setup employed to implement the scheme. In section 5.3, Prototype Implementation. Finally, in section 5.4 the simulation experiment and evaluation results are described.

5.2. Experimental Environment Setup

The selection of the experimental environment and simulation tool that was used for the implementation and evaluation of our proposed solution is described below. The proposed solution implementation was conducted through simulation considering the ONE simulator tool [46]. The simulator is specifically designed for evaluating DTNs routing and application protocols. Also supports generating different node movement models, inter-node contacts, routing, and message handling. Therefore the ONE simulator is the right tool used to demonstrate our proposed solution in this thesis. Many of the routing algorithms applicable to the DTN environment are pre-implemented in the simulator. Also, these routing protocols work under the store-carry-forward mechanism. In this study, the proposed solution algorithm is incorporated into the MaxProp routing algorithm because the protocol has more performance than the other routing protocols in the considered simulation scenario. Besides the node movement model, map route movement and bus movement models were provided with the ONE simulator used across all the simulation experiments. The Well-Known Text (WKT) file of the map in ONE simulator which represents a part of the city of Helsinki, Finland (Figure 5.1) was used for simulation. Accordingly, the movement model obtains its configuration data using files formatted with a subset of the well-known text format, and with the mobility models specified above; the mobile nodes move using roads from the map data.

Figure 5.1: Simulated Map of a Part of Helsinki, Finland [46].

5.2.1. The Simulation Scenario

The analysis phase uses the following scenario for describing the proposed contact duration aware forwarding strategy using the ONE simulator modeling on node movement; inter-node contact duration, routing, and message handling. In this study, the following points were considered for the simulation scenario. To share the created messages with the help of mobile nodes that store carry and forward, they stored messages when the nearby nodes are encountered opportunistically, and it defines two types of node mobility models; map route movement and bus movement model. The scenario was built by defining the simulated nodes and their capabilities. The defined nodes and their capabilities include the basic parameters such as storage capacity, transmit range, bit-rates, movement, routing models, and simulation duration. In the simulation, we used a different group of nodes moved by different movement models. The considered scenario consists of 12 groups of nodes, 8 groups of nodes move by bus movement model, and the other groups move by map route movement model. Since in both considered movement models there is the regularity of movement path by contact speed, it is assumed that past contact duration between pair of nodes is the same next time when the nodes contact again. The detail of various simulation parameters for the simulation scenario is shown in table 5.1.

Table 5.1: Simulation Environment Parameters.

Parameter	**Value**
Simulation time	1,2,3,4,5,6 hours
Node number	15,20,25,30,35,40
Simulation scenario size	10000, 8000m
Mobility model	Bus movement and Map route movement
Buffer size	20,30,40,50,60,70 MB
Routing	MaxProp
Number of groups	12
Transmission range	100m
Packet transmission speed	1375k
Message TTL	5 minutes
Message size	5 MB,10 MB
Message generation interval	10,20 minutes

5.2.2 Message Generation

The ONE simulator provides two ways to generate application messages inside the simulation. These are message generators and external event files. The first way of message generation is built in the ONE simulator. It creates messages with a random or fixed source, destination, size, and interval. The second way of message generation is external event files. For the simulation scenario, the message generated by the message generator which is pre-implemented in ONE simulator was used. The message event generator is set in such a way that it generates the messages every 10 to 20 minutes and every time message size can also be varied from 5MB to 10MB.

5.2.3. Mobility Model

This section provides the movement models of the node opted in the ONE simulator for node mobility. As DTN networks are contingent greatly on the physical movement of devices that can store carry and forward messages with other DTN devices, the performance of a DTN would seem to depend on the underlying mobility assumed of the smart vehicle that has onboard units for communication. In this study, the smart vehicles were considered to buffer and exchange messages with other smart vehicles.

For the proposed contact duration aware forwarding strategy, the smart nodes have a communication board, and each node has IEEE 802.11b interface to exchange information when it encounters other nodes. The information exchange takes place when the mobile nodes encounter any other mobile nodes in the network. The node mobility extends the range of the network and allows data paths to exist over time in networks that suffer from long periods of disconnection. Thus, the onboard unit on the nodes acts as the communication infrastructure for the network, being opportunistically explored to collect, carry and disseminate data. Once the generated message is delivered to the terminal node, it is assumed that the end user can access the data from the local device.

In this work, the map route movement and bus movement models which are pre-implemented in the ONE simulator are chosen for simulating the node mobility. The map route movement model has predetermined routes on the map that they travel. In this model, nodes are assigned predetermined routes that they must travel on the map. This kind of model shows better performance in simulating nodes' movement, especially, on the bus and tram routes. In these node movement models there is the regularity of motion path and node contact duration, so past contact duration between pair of nodes will occur next time when the nodes contact again. Therefore these movement models are the right models to test our proposed scheme.

5.2.4. Reporting and Visualization

This section describes the way how the results of the simulation were visualized. These are through an interactive Graphical User Interface (GUI) and with generated images from the information gathered during the simulation. Figure 5.2 shows the GUI displaying the simulation in real-time and it has been used for demonstration purposes. In the main window the node locations, current paths, connections between nodes, number of messages carried by a node, etc. are all visualized. Also, the GUI produces a filtered log of simulation events, such as contacts and message transfer. While the GUI is good for getting an overall picture of what is happening during the simulation, more rigorous ways to visualize node relations, message paths, and performance summaries are provided by post-processed report files.

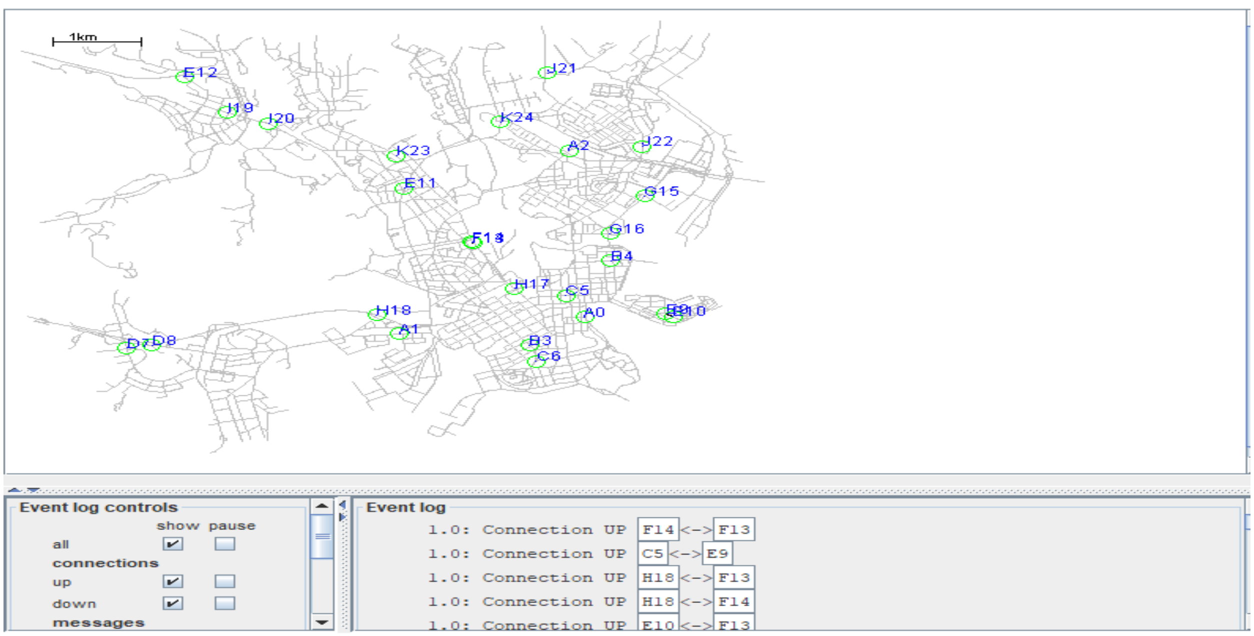

Figure 5.2: Screenshot of the ONE Simulator GUI.

Besides the simulation results of the GUI, the majority of the result used for the analysis was collected primarily through reports generated by report modules during the simulation run. Report modules receive events like messages or connectivity events from the simulation engine and generate results based on them. During the simulation run, the simulator generates the message statistics file in the report module that gathers statistics on overall performance (i.e., the number of created messages, message delivery ratio, how long messages stay in node buffers, etc.). Accordingly, the report module's output was used to plot a graph for the various message statistics report generated by the ONE simulator.

5.3. Simulation Experiment and Result Analysis

To test our scheme we use a simulation experiment according to different parameters. To implement our scheme initially we made a simulation set up to conduct the simulation. After that, we identify and define network parameters. Finally, we analyze and compare our scheme with the existing buffer management forwarding schemes.

5.3.1. Performance Evaluation Metrics

Here we described the metrics adopted for evaluating the forwarding strategy of considered routing algorithms. Routing in the communication networks depends on the definition of a performance indicator called routing metrics. As noted in [47], the popular routing metrics in traditional networks are the number of hops, end-to-end delay, jitter, route lifetime, and route throughput. However, in delay tolerant networks, these metrics are problematic due to asynchronous specificities. Therefore, DTNs use different routing metrics from the traditional networks as it is clearly defined in [48]. The proposed forwarding algorithm considered in this thesis was analyzed based on two routing metrics namely delivery probability and average latency. The detailed description of each is explained as follows:

Delivery Probability: It is the fraction of generated messages that are correctly delivered to the final destination within a given time. It is calculated according to Equation 5.1:

Notation: DP = is the bundle delivery probability

DB = is the total number of unique delivered bundles

CB = is the total number of unique created bundles

Equation 5-1 Delivery Probability

$$Dp = \frac{\sum DB}{\sum CB} \quad \text{-----} \quad (5.1)$$

Average Latency: It is defined as the average time taken from the source node to transfer the message to the destination node. It is the measure of the average time between messages is generated and when it is received by the destination. In other words, the average latency is the time it takes for a delivered message to reach its destination. Average latency includes all types of delay such as buffer delay, route discovery process, delay during retransmission of the message, propagation time, etc. The small value of the end-to-end delay means a better performance of the protocol. It is calculated according to Equation 5.2:

Notation: Dd = is the bundle average delivery delay

Tdi = is the time when the bundle i was delivered

Tci = is the time when the bundle i was created

DB= is the total number of unique delivered bundles

Equation 5-2 Average Delay

$$Dd = \frac{\sum_{i=1}^{DB}(Tdi - Tci)}{DB} \quad \text{------} \quad (5.2)$$

5.3.2. Result Analysis and Discussion

In this thesis, the network performance is studied from the application performance point of view. Based on different simulation parameters, each simulation run generates a trace file containing all the data packets that are sent between the nodes during the course of the simulated scenario. Accordingly, the performance of the considered routing protocol forwarding strategy and a proposed scheme was determined through analysis of the generated trace file of the simulation. In the considered simulation scenario MaxProp routing protocol has high performance than other DTN routing protocols. Thus our proposed algorithm is integrated and evaluated with the forwarding policy of MaxProp routing protocol. We presents a detailed analysis of the simulation result using a chart based on the generated trace file of the simulation. The results observed in simulation experiments with MaxProp routing protocol proved the importance of the proposed contact duration aware forwarding strategy.

5.3.2.1. Performance Evaluation Concerning Number of Node

Figure 5.3 shows the existing forwarding scheme and proposed contact duration aware forwarding mechanism. Both the existing and the proposed mechanisms help to reduce the impact of node mobility on message transmission by increasing the number of delivery bundles. The same figure also shows that the proposed mechanism overtakes message delivery probability more than the existing one because our proposed scheme forward complete transmit messages and monitored limited contact duration between pairs of network nodes before affecting the network performance. Thus we avoided inappropriate message drop due to partial message transmission. More accurately, the proposed contact duration aware forwarding schema presents an improvement of approximately 0.54%, 1.2%, 2.34%, 2.97%, 4.33% and 5.44% of bundles (for a number of nodes equal to 15, 20, 25, 30, 35 and 40, respectively) when compared to an existing scheme.

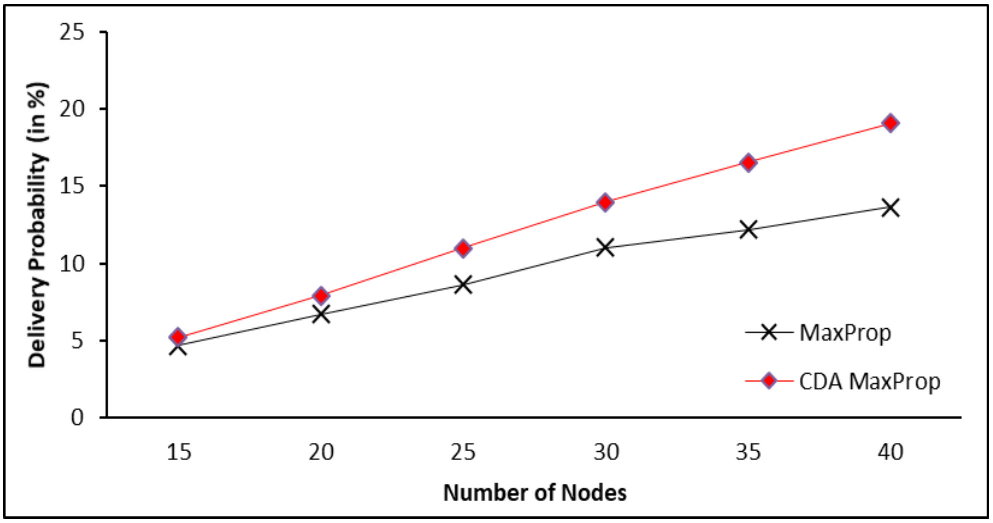

Figure 5.3: Bundle delivery probability by varying number of nodes.

Also, Figure 5.4 approves the results observed in Figure 5.3. The proposed contact duration aware forwarding scheme reward and punish in a disproportional way not only delivering more bundles but also delivering bundles faster. Since our scheme forwarded only complete transmit message, we reduced destination node waiting time till to receive complete message. Which is more applicable when few number of nodes deployed in large areas. Comparatively, with the existing scheme, the proposed scheme delivers bundles approximately 3.34, 3.08, 2.79, 2.69, 2.02 and 1.39 minutes sooner (for a node number equal to 15, 20, 25, 30, 35 and 40, respectively).

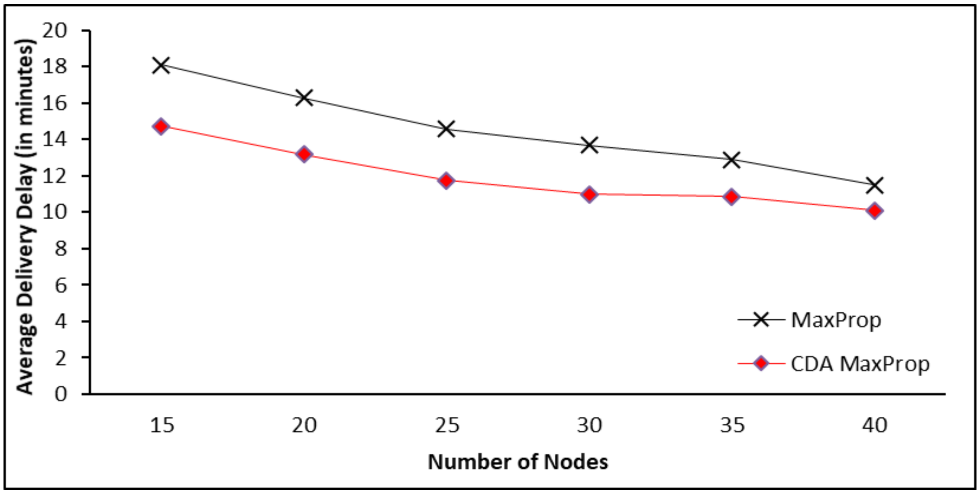

Figure 5.4: Bundle average delivery delay by varying number of nodes.

5.3.2.2. Performance Evaluation Concerning Simulation Time

Figure 5.5 show that the proposed contact duration aware forwarding scheme increases about 0.84%, 1.1%, 1.51%, 1.71%, 1.73% and 2.04% the bundle delivery probability (for the considered simulation time of 1, 2, 3, 4, 5 and 6 hours respectively) when compared with the existing forwarding scheme.

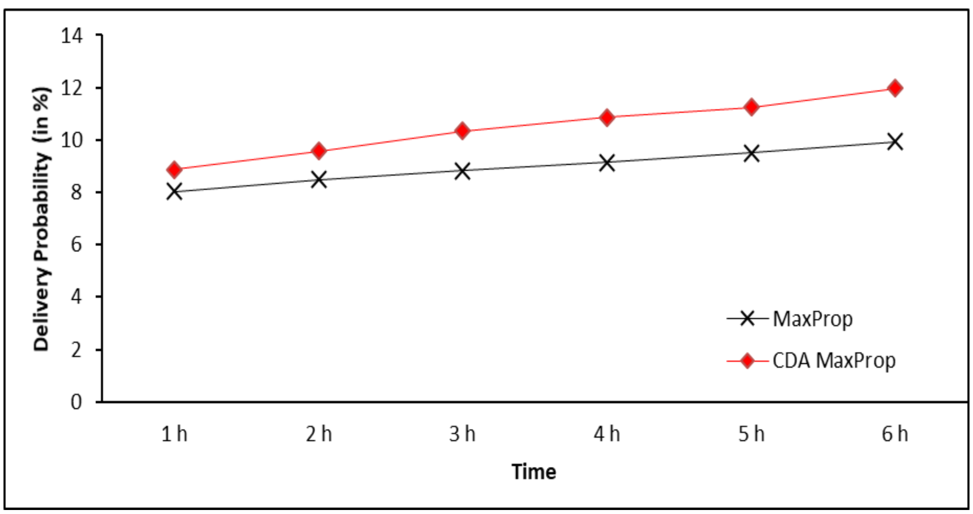

Figure 5.5: Bundle delivery probability by varying simulation time.

In addition, increasing the simulation time supports the improvement in the average delay that was introduced by the proposed contact duration aware forwarding scheme. Figure 5.6 shows that bundles will arrive at their final destination approximately 1.16, 1.24, 1.81, 2.12, 2.73, and 3.27

minutes sooner (for the considered simulation time of 1, 2, 3, 4, 5, and 6 hours respectively) when compared with the existing forwarding scheme.

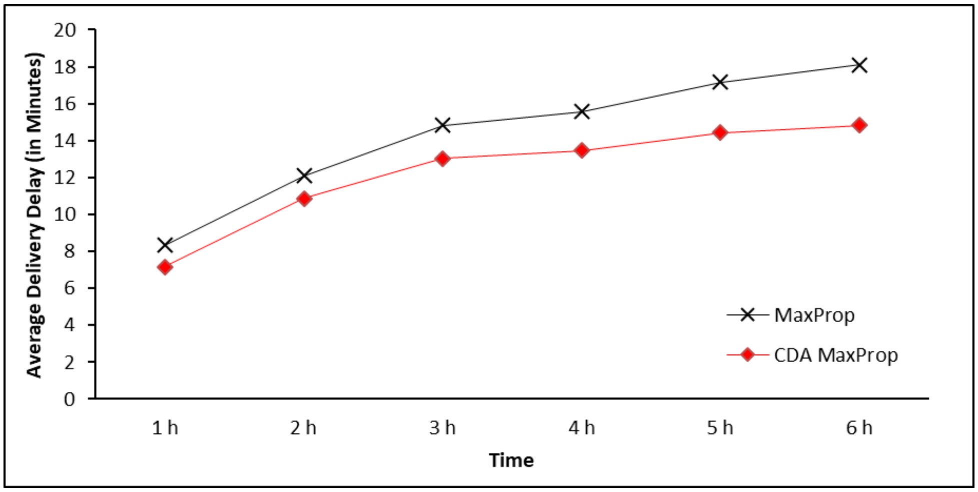

Figure 5.6: Bundle average delivery delay by varying simulation time.

5.3.2.3. Performance Evaluation Concerning Buffer Size

Figure 5.7 show that the proposed contact duration aware forwarding scheme increases about 3.54%, 3.15%, 2.91%, 2.67%, 2.47% and 2.13% the bundle delivery probability (for the considered Buffer Size of 20MB, 30MB, 40MB, 50MB, 60MB, and 70MB respectively) when compared with the existing forwarding scheme. The gains observed in the bundle delivery probability performance metric are raised when mobile nodes have a bigger Buffer Size.

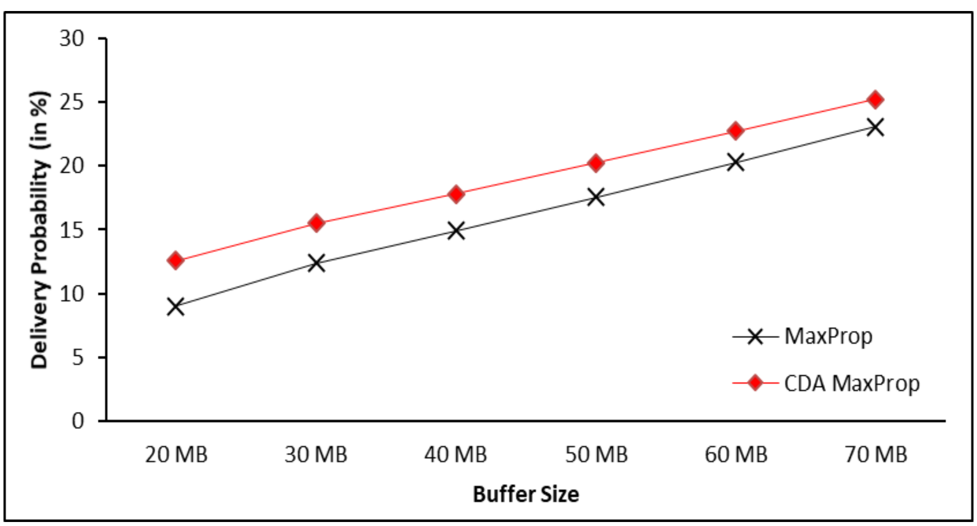

Figure 5.7: Bundle delivery probability by varying buffer sizes.

In addition, increasing the Buffer Size supports the improvement in the average delay that was introduced by the proposed contact duration aware forwarding scheme. Figure 5.8 shows that bundles will arrive at their final destination approximately 7.77, 5.77, 4.53, 3.87, 3.25 and 2.65 minutes sooner (for the considered Buffer Size of 20MB, 30MB, 40MB, 50MB, 60MB, and 70MB respectively) when compared to the existing mechanism.

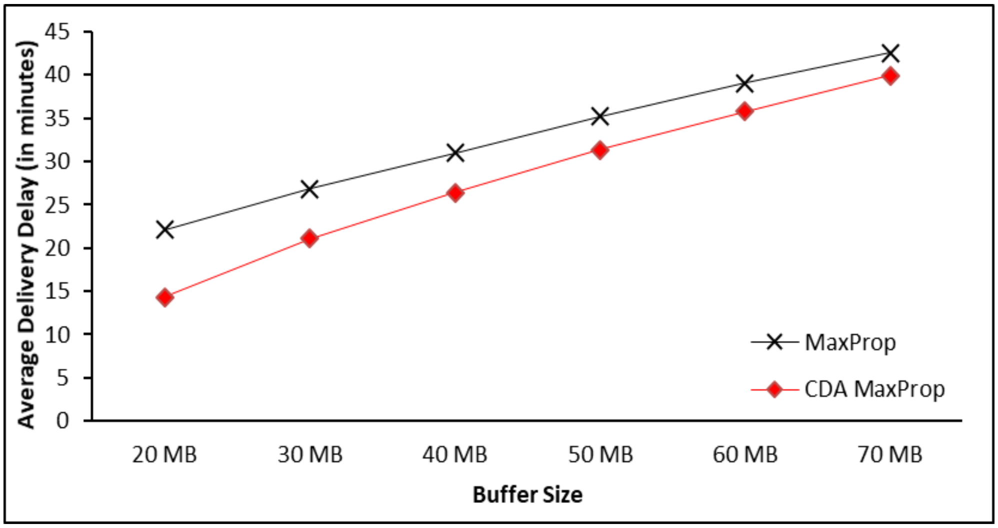

Figure 5.8: Bundle average delivery delay by varying buffer sizes.

5.4. Summary

Generally, we tested our proposed solution by using ONE simulator. After extensive experiments, we analyzed the performance of our proposed solution, and finally, we compared our work with the existing forwarding strategy of MaxProp routing protocol. The proposed contact duration aware forwarding strategy has been chosen to evaluate the performance of the MaxProp routing protocol concerning the number of nodes, simulation time, and node buffer size. From the simulation result, we concluded that our scheme has realized better in terms of delivery probability and average delivery delay. The objective of our work is to design the scheme to enhance the performance of buffer management policy using contact duration aware forwarding strategy in sparse delay tolerant networks. Thus we improved the listed above metrics used and obtain expected results from the simulation.

CHAPTER SIX

CONCLUSION, CONTRIBUTION, AND FUTURE WORKS

6.1. Conclusion

This study has focused on limited contact duration among pairs of delay-tolerant network nodes. The contact duration between any pair of network nodes is limited due to node movement and the limited range of peer-to-peer wireless communication. In sparse delay tolerant networks the pairs of network nodes take a long time to contact again, which is much longer than contact duration. The nodes require waiting till they come into communication range again when do not finish exchanging the message during the contact time. Consequently, the message which is started to transmit may drop when expired before the nodes contact again and the destination node waits till to receive complete message. Also, the receiver node drops a complete message to create a room for incoming partial message items upon buffer overflow. While much work has been done in the design of forwarding algorithms, little work has focused on studying forwarding under the presence of limited contact duration. Thus limited contact duration needs more consideration to design buffer management forwarding policy.

The purpose of this study was to enhance the performance of buffer management policy in sparse delay-tolerant networks. To reduce the impact of limited contact duration in message transmission, an efficient contact duration aware forwarding approach was proposed. The basic idea of the proposed forwarding principle was to forward complete transmit messages between pair of nodes within contact duration. Our proposed scheme considered contact duration to identify the number and order of complete transmit messages to be forwarded. The proposed algorithm was integrated with the forwarding policy of MaxProp routing protocol and performance evaluated with the protocol. The ONE simulator and scripting languages were used as tools to simulate the proposed scheme.

Finally, we have tested, evaluated, and proved our proposed scheme with the existing ones. From the conducted simulation, results considering bundle delivery probability and average delivery delay showed that the proposed scheme outperforms than existing forwarding strategy. In general, using a contact duration aware forwarding strategy on buffer management policies provides a novel result.

6.2. Contribution

In this thesis, we study limited contact duration between pairs of network nodes which is a fundamental problem in sparse delay tolerant networks. The core contribution of this thesis is bringing the concepts of limited contact duration as a core mechanism to manage forwarding messages between pairs of network nodes. Finally, we provided a novel contact duration aware forwarding policy which has better performance than the existing policy in sparse delay tolerant networks.

6.3. Future works

Our proposed scheme can address the limited contact duration between pairs of network nodes in sparse delay-tolerant networks. Also, it can be a good forwarding strategy for buffer management policy in sparse delay tolerant networks. However, in delay tolerant networks inter-contact time between pairs of network nodes is one of the key metrics central to forwarding algorithms and end-to-end delay. Thus a clear deferral of our work could be extended to consider also the inter-contact time between pairs of network nodes in buffer management forwarding policy.

References

[1] K. Charles. *The TCP/IP guide: a comprehensive, illustrated Internet protocols reference*. No Starch Press, 2005.

[2] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis, H. Weiss, and K. Fall, "Delay-Tolerant Network Architecture: The Evolving Interplanetary Internet", draft-irtf-ipnrg-arch-01 (work in progress), August 2002.

[3] C. Vinton, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. *Delay-tolerant networking architecture*. No. rfc4838. 2007.

[4] V.Ankita, and S.Kumar. "Routing protocols in delay tolerant networks: Comparative and empirical analysis." *Wireless Personal Communications* 118, no. 1 (2021): 551-574.

[5] X. Zhuo, Q. Li, W. Gao, G. Cao and Y. Dai, "Contact duration aware data replication in Delay Tolerant Networks," 2011 19th IEEE International Conference on Network Protocols, 2011, pp. 236-245, doi: 10.1109/ICNP.2011.6089057.

[6] T. Le and M. Gerla, "Contact Duration-Aware Routing in Delay Tolerant Networks," 2017 International Conference on Networking, Architecture, and Storage (NAS), 2017, pp. 1-8, doi: 10.1109/NAS.2017.8026849.

[7] Fall and Kevin. "A delay-tolerant network architecture for challenged internets." In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 27-34. 2003.

[8] D. Elwyn. "DTN-The State of the Art." Networking for Communications Challenged Communities. April 2009.

[9] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, E. Travis, and H. Weiss, " Interplanetary Internet (IPN): Architectural Definition", draft-irtfipnrgarch-00 (work in progress), May 2001.

[10] InterPlanetary Networking Special Interest Group. "Interplanetary internet project." International Journal of Grid Distribution Computing, Vol.8, No.1, pp.169 – 178, 2015

[11] V. Cerf, S. Burleigh, and A. Hookeetal. "Delay-tolerant networking architecture." Available: https://tools.ietf.org/html/rfc4838, [Accessed: Sep. 2, 2017].

[12] K. Scott and S. Burleigh. "Bundle Protocol Specification." Available: https://tools.ietf.org/html/rfc5050, November 2007.

[13] K. Fall and S. Farrell, "DTN: an architectural retrospective," in IEEE Journal on Selected Areas in Communications, vol. 26, no. 5, pp. 828-836, June 2008, doi: 10.1109/JSAC.2008.080609.

[14] D. Michael, J. Ott, and S. Perreault. Delay-tolerant networking tcp convergence-layer protocol. No. rfc7242. 2014.

[15] X. Wang, C. Yuen, X. Chen, N. Hassan, Y. Ouyang, Cost-aware demand scheduling for delay tolerant applications, In Journal of Network and Computer Applications, Volume 53, 2015, Pages 173-182, ISSN 1084-8045, https: //doi.org/10.1016/j.jnca.2015.04.002.

[16] T. Neau, M. Amorim, and V. Conan. Uncovering vicinity properties in disruptiontolerant networks. Comput. Netw. 73, C (November 2014), 210-223. DOI=http: //dx.doi.org/10.1016/j.comnet.2014.08.004

[17] V. Juyal, A. V. Singh and R. Saggar, "Message Multicasting in Near-Real Time Routing for Delay/Disruption Tolerant Network," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, 2015, pp. 385-390, doi: 10.1109/CICT.2015.79.

[18] Y. Cao and Z. Sun. “Routing in Delay/ Disruption Tolerant Networks: A Taxonomy, Survey and Challenges.” Communications Surveys & Tutorials, pp. 654-677, 2013

[19] A. Galati, T. Bourchas, S. Siby, S. Frey, M. Olivares and S. Mangold, "Mobile-enabled delay tolerant networking in rural developing regions," IEEE Global Humanitarian Technology Conference (GHTC 2014), 2014, pp. 699-705, doi: 10.1109/GHTC.2014.6970359.

[20] V. Cerf, S. Burleigh, and A. Hookeetal. “Delay-tolerant networking architecture.” Available: https://tools.ietf.org/html/rfc4838, [Accessed: Sep. 2, 2017].

[21] M. Shahzamal, F. Pervez, U. Zaman, and D. Hossain. "Mobility models for delay tolerant network: A survey." *International Journal of Wireless & Mobile Networks (IJWMN) Vol* 6 (2014).

[22] V. Ankita, and S. Kumar. "Routing protocols in delay tolerant networks: Comparative and empirical analysis." *Wireless Personal Communications* 118, no. 1 (2021): 551-574.

[23] V. Amin, and D. Becker. "Epidemic routing for partially connected ad hoc networks." (2000): 2019.

[24] B. John, B. Gallagher, D. Jensen, and N. Levine. "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks." In *Infocom*, vol. 6. 2006.

[25] E. Vijay, and M. Crovella. "Forwarding in opportunistic networks with resource constraints." In *Proceedings of the third ACM workshop on Challenged networks*, pp. 41-48. 2008.

[26] Z. Li and H. Shen, "SEDUM: Exploiting Social Networks in Utility--Based Distributed Routing for DTNs," in IEEE Transactions on Computers, vol. 62, no. 1, pp. 83-97, Jan. 2013, doi: 10.1109/TC.2011.232.

[27] Z. Li and H. Shen, "SEDUM: Exploiting Social Networks in Utility--Based Distributed Routing for DTNs," in IEEE Transactions on Computers, vol. 62, no. 1, pp. 83-97, Jan. 2013, doi: 10.1109/TC.2011.232.

[28] M. Shashidhar, H. Ammar, and W. Zegura. *Routing in space and time in networks with predictable mobility*. Georgia Institute of Technology, 2004.

[29] J. LeBrun, Chen-Nee Chuah, D. Ghosal and M. Zhang, "Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks," 2005 IEEE 61st Vehicular Technology Conference, 2005, pp. 2289-2293 Vol. 4, doi: 10.1109/VETECS.2005.1543743.

[30] H. Huang et al., "Performance Evaluation of SUVnet With Real-Time Traffic Data," in IEEE Transactions on Vehicular Technology, vol. 56, no. 6, pp. 3381-3396, Nov. 2007, doi: 10.1109/TVT.2007.907273.

[31] J. Zhu, J. Cao, M. Liu, Y. Zheng, H. Gong and G. Chen, "A Mobility Prediction-Based Adaptive Data Gathering Protocol for Delay Tolerant Mobile Sensor Network," IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference, 2008, pp. 1-5, doi: 10.1109/GLOCOM.2008.ECP.146.

[32] B. Christian, and G. Schiele. "New mechanisms for routing in ad hoc networks through world models." In *Proceedings of the 4th CaberNet Plenary Workshop, Pisa, Italy*, pp. 1-4. 2001.

[33] Z. Wenrui, M. Ammar, and E. Zegura. "A message ferrying approach for data delivery in sparse mobile ad hoc networks." In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pp. 187-198. 2004.

[34] Y. Quan, I. Cardei, and J. Wu. "Predict and relay: an efficient routing in disruption-tolerant networks." In *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, pp. 95-104. 2009.

[35] A. Mei and J. Stefa, "Give2Get: Forwarding in Social Mobile Wireless Networks of Selfish Individuals," in IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 4, pp. 569-582, July-Aug. 2012, doi: 10.1109/TDSC.2012.37.

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

[37] M. Khalil, A. Vitaletti, A. Vernata, and I. Chatzigiannakis. "Routing protocols for delay tolerant networks: a reference architecture and a thorough quantitative evaluation." *Journal of Sensor and Actuator Networks* 5, no. 2 (2016): 6.

[38] B. John, B. Gallagher, D. Jensen, and N. Levine. "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks." In *Infocom*, vol. 6. 2006.

[39] C.Sobin. "An efficient buffer management policy for DTN." *Procedia Computer Science* 93 (2016): 309-314.

[40] W. Hezhe, H. Wang, G. Feng, and H. Lv. "NWBBMP: a novel weight-based buffer management policy for DTN routing protocols." *Peer-to-Peer Networking and Applications* 11, no. 5 (2018): 917-923.

[41] S. K. Dhurandher, J. Singh, I. Woungang and J. J. P. C. Rodrigues, "Priority Based Buffer Management Technique for Opportunistic Networks," 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1-6, doi: 10.1109/GLOBECOM38437.2019.9013339.

[42] P. Matzakos, T. Spyropoulos and C. Bonnet, "Joint Scheduling and Buffer Management Policies for DTN Applications of Different Traffic Classes," in IEEE Transactions on Mobile Computing, vol. 17, no. 12, pp. 2818-2834, 1 Dec. 2018, doi: 10.1109/TMC.2018.2816025.

[43] C. -Ș. Nănău, "MaxDelivery: A New Approach to a DTN Buffer Management," 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), 2020, pp. 60-61, doi: 10.1109/WoWMoM49955.2020.00023.

[44] D. Zhaoyang, W. Celimuge, C. Xianfu, X. Wang, T. Yoshinaga, and Y. Ji. "A VDTN scheme with enhanced buffer management." *Wireless Networks* 26, no. 3 (2020): 1537-1548.

[45] G. Goudar and S. Batabyal, "Optimizing Bulk Transfer Size and Scheduling for Efficient Buffer Management in Mobile Opportunistic Networks," in IEEE Transactions on Mobile Computing, doi: 10.1109/TMC.2021.3075993.

[46] K. Ari, J. Ott, and K. Teemu. "The ONE simulator for DTN protocol evaluation." In *Proceedings of the 2nd international conference on simulation tools and techniques*, pp. 1-10. 2009.

[47] S.Martha. *Routing in communications networks*. Prentice Hall International (UK) Ltd., 1995.

[48] N. Mehta and M. Shah. “Performance Evaluation of Efficient Routing Protocols in Delay Tolerant Network under Different Human Mobility Models.” International Journal of Grid Distribution Computing. Vol.8, pp. 169 -178, No.1 2015

Appendix

Sample of .txt settings file

```
Scenario.name = Contact_Duration_Aware
Scenario.simulateConnections = true
Scenario.updateInterval = 1
Scenario.endTime = 21600

WiFi80211.type = SimpleBroadcastInterface
WiFi80211.transmitSpeed = 1375k
WiFi80211.transmitRange = 100

Scenario.nrofHostGroups = 12

Group.movementModel = MapRouteMovement
Group.router = MaxPropRouter
Group.bufferSize = 20M
Group.waitTime = 0,24
Group.nrofInterfaces = 1
Group.interface1 = WiFi80211
Group.speed = 0.5, 1.5
Group.msgTtl = 300
Group.nrofHosts = 2

Group1.groupID = A
Group1.nrofHosts = 3

Group2.groupID = B
Group2.speed = 7, 10
Group2.movementModel = BusMovement
Group2.routeFile = data/HelsinkiMedium/A_bus.wkt
Group2.routeType = 2
Group2.busControlSystemNr = 1
        .
        .
Group12.groupID = K
Group12.speed = 7, 10
Group12.movementModel = BusMovement
Group12.routeFile = data/HelsinkiMedium/H_bus.wkt
Group12.routeType = 2
```

Group12.busControlSystemNr = 8

Events.nrof = 1
Events1.class = MessageEventGenerator
Events1.interval = 10, 20
Events1.size = 5M, 10M
Events1.hosts = 0,24
Events1.prefix = M

MovementModel.rngSeed = [2; 8372; 98092; 18293; 777]
MovementModel.worldSize = 10000, 8000
MovementModel.warmup = 43000
MapBasedMovement.nrofMapFiles = 1
MapBasedMovement.mapFile1 = data/HelsinkiMedium/roads.wkt

Report.nrofReports = 2
Report.warmup = 0
Report.reportDir = reports/ProposedMaxPropRouter
Report.report1 = MessageGraphvizReport
Report.report2 = MessageStatsReport

Sample of ONE program

```
public class ProposedMaxPropRouter extends ActiveRouter {
        public static final String MAXPROP_NS = "ProposedMaxPropRouter";
        public static final String PROB_SET_MAX_SIZE_S = "probSetMaxSize";
        public static final int DEFAULT_PROB_SET_MAX_SIZE = 50;
        private static int probSetMaxSize;
        private MeetingProbabilitySet probs;
        private Map<Integer, MeetingProbabilitySet> allProbs;
        private MaxPropDijkstra dijkstra;
        private Set<String> ackedMessageIds;
        private Map<Integer, Double> costsForMessages;
        private DTNHost lastCostFrom;
        private Map<DTNHost, Set<String>> sentMessages;
        public static int BYTES_TRANSFERRED_AVG_SAMPLES = 10;
        private int[] avgSamples;
        private int nextSampleIndex = 0;
        private int avgTransferredBytes = 0;
        public static final String ALPHA_S = "alpha";
        private double alpha;
        public static final double DEFAULT_ALPHA = 1.0;
        private double ContactDuration;
        private double ConnectionUpTime;
        private double ConnectionDownTime;
        private double DAM;
        private double DSM;
       private List<Message> Messages;
       private Map<DTNHost,Double> ContactInfo;
```

```
public ProposedMaxPropRouter(Settings settings) {
            super(settings);
                 Settings maxPropSettings = new Settings(MAXPROP_NS);
                         if (maxPropSettings.contains(ALPHA_S)) {
                                 alpha = maxPropSettings.getDouble(ALPHA_S);
                         } else {
                                 alpha = DEFAULT_ALPHA;
                         }
                   Settings mpSettings = new Settings(MAXPROP_NS);
                   if (mpSettings.contains(PROB_SET_MAX_SIZE_S)) {
                       probSetMaxSize = mpSettings.getInt(PROB_SET_MAX_SIZE_S);
                   } else {
                              probSetMaxSize = DEFAULT_PROB_SET_MAX_SIZE;
                    }
           }
  protected ProposedMaxPropRouter(ProposedMaxPropRouter r) {
            super(r);
            this.alpha = r.alpha;
            this.probs = new MeetingProbabilitySet(probSetMaxSize, this.alpha);
            this.allProbs = new HashMap<Integer, MeetingProbabilitySet>();
            this.dijkstra = new MaxPropDijkstra(this.allProbs);
            this.ackedMessageIds = new HashSet<String>();
            this.avgSamples = new int[BYTES_TRANSFERRED_AVG_SAMPLES];
             this.sentMessages = new HashMap<DTNHost, Set<String>>();
             this.ContactDuration = 0;
             this.ConnectionUpTime = 0;
             this.ConnectionDownTime = 0;
```

```
                this.DAM = 0;
                this.DSM = 0;
                this.Messages = new ArrayList<Message>();
                this.ContactInfo = new HashMap<DTNHost,Double>();
        }
public void IntendedMessage(Connection con){
Collection<Message> msgCollection = getMessageCollection();
    if (con.isUp()) {
        double TotalMessageSize = 0;
        double MinSizeMessage = Integer.MAX_VALUE;
        DTNHost other = con.getOtherNode(getHost());
        ProposedMaxPropRouter othRouter = (ProposedMaxPropRouter)other.getRouter();
        Set<String> sentMsgIds = this.sentMessages.get(other);
    for (Message m : msgCollection) {
            if (othRouter.hasMessage(m.getId()) || m.getHops().contains(other)) {
                continue;
            }
            if (sentMsgIds != null && sentMsgIds.contains(m.getId())) {
                continue;
            }
        Messages.add(m);
        Collections.sort(Messages);
  }
        for (Message m : Messages) {
            if(m.getSize()<MinSizeMessage){
                MinSizeMessage=m.getSize();
            }
```

```
                TotalMessageSize += m.getSize();
            }
            DAM = TotalMessageSize/con.getSpeed();
            DSM = MinSizeMessage/con.getSpeed();
        }
    }
public Tuple<Message, Connection> tryOtherMessages() {
List<Tuple<Message, Connection>> messages = new ArrayList<Tuple<Message,
Connection>>();
        for (Connection con : getConnections()) {
            DTNHost other = con.getOtherNode(getHost());
            ProposedMaxPropRouter othRouter = (ProposedMaxPropRouter)other.getRouter();
            Set<String> sentMsgIds = this.sentMessages.get(other);
            if (con.isUp()) {
                    ConnectionUpTime=getSimTime();
            }
             else {
                ConnectionDownTime=getSimTime();
                ContactDuration = this.ConnectionDownTime - this.ConnectionUpTime;
                 ContactInfo.put(other,ContactDuration);
             }
             if (othRouter.isTransferring()) {
                 continue;
             }
    for (Message m : Messages) {
            if(ContactInfo.get(other)<DAM && ContactInfo.get(other)>=DSM){
                double getContactDuration=ContactInfo.get(other);
              if (getContactDuration<m.getSize()/con.getSpeed()){
```

```
                        continue;
                    }
                   else{
                            messages.add(new Tuple<Message, Connection>(m,con));
                            getContactDuration=getContactDuration - m.getSize()/con.getSpeed();
                    }
          }
             else {
                      messages.add(new Tuple<Message, Connection>(m,con));
                  }
          }
    }
          if (messages.size() == 0) {
                return null;
          }

          Collections.sort(messages, new MaxPropTupleComparator(calcThreshold()));
          return tryMessagesForConnected(messages);
}
```