JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

Message Drop Control with Priority Buffer Management policy for Delay Tolerant Networks

By

Kasech Tsegaye Turufi

February 2020

JIMMA, ETHIOPIA

JIMMA UNIVERSITY

JIMMA INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

Message Drop Control with Priority Buffer Management policy for Delay Tolerant Networks

By: Kasech Tsegaye Turufi

This Research Study Submitted To Faculty of Computing, Institute of Technology, Jimma University, In Partial Fulfillment For Degree of Master of Science In Computer Networking

Principal Advisor: Dr. Girum Ketema

Co-Advisor: Mr. Gemechu Birhanu

February 2020

JIMMA, ETHIOPIA

DECLARATION

This is to certify that the thesis prepared by Kasech Tsegaye titled: Message Drop Control with Priority Buffer Management Policy for Delay-Tolerant Networks has been read and approved as meeting the requirements of Department of Computing in partial fulfillment for the degree of Master of Science in Computer Networking complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Advisor Name: Dr. Girum Ketema

Signature: ______________, Date: 28/02/2020

Co-Advisor Name: Mr. Gemechu Birhanu

Signature: ______________, Date: ______________

External Examiner Name: Dr. Melkamu Dheresa

Signature: ______________, Date: ______________

Internal Examiner Name: Mr. Kebebew Ababu

Signature: ______________, Date: ______________

Abstract

The term delay-tolerant network (DTN) is coined to describe and include all types of disconnected,long delay, disrupted or intermittently-connected networks. So, in such network, communication is created with the help of encounter opportunities between nodes and those nodes follow store-carry and forward approach to achieve data delivery. In this approach, each node stores the messages in their buffers until they encounter another node and also to achieve a high delivery ratio most of DTN routing transmits several copies of messages. But the combination of long term storage and several copies of message transmission in this process impose greater transmission overhead and buffer occupancy. Most of the previous works on improving the performance of routing do not consider the issue of resource limitations like buffer and bandwidth. However, hand held devices which are having limited storage and bandwidth are used in many of DTN applications Therefore, buffer management policies have a direct impact on routing performance of DTNs routing. Basing on this point, in this paper Message Drop control with priority(MDCP) buffer management policy has been proposed in the situations where there is short contact duration, limited bandwidth and buffer.

The proposed policy prioritize messages in parallel with controlling unnecessary message drops by using three local information's namely hop count, time to live and size of messages. The performance of proposed policy is validated through simulation by using opportunistic Network Environment (ONE) simulator. The proposed Message Drop Control with priority buffer management policy (PMDCP) and existing buffer management policies namely Drop oldest (DO), Drop Largest (DL), Drop Maximum Hop Count (DMaxHop), Efficient buffer management policy (EBMP) was adapted on Epidemic Routing(ER) protocol for evaluation. They were analyzed on two different metrics namely delivery probability and overhead ratio. The simulation results obtained in this thesis shows that for the ER with the proposed policy, the message delivery probability is high and overhead is low. Therefore, the proposed policy has better performances than existing policies by increasing delivery probability and decreasing overhead ratio.

Keywords: *Delay Tolerant Networks,Routing, Buffer management*

Dedication

I Dedicate this thesis for My Beloved Family.

Acknowledgments

First of all I would like to thank my GOD,for his blessing and steering throughout my life. Next, I would like to express my sincere gratitude to my advisors Dr. Girum Ketema and Mr. Gemechu Birhanu for their critical and constructive guidance starting from the beginning till end. Last but not least, my thanks extends to my families and friends for their important help and support.

Contents

List of Figures

List of Tables

List of Algorithms

List of Equations

List of Acronyms

DTN: Delay Tolerant Networks

DL: Drop Largest

DO: Drop Oldest

DMaxHop: Drop Maximum hop-count

EBMP: Efficient Buffer Management policy

E-Drop: Equal Drop

ER: Epidemic Routing

GUI: Graphical User Interface

MANET: Mobile Ad-hoc Networks

ONE: Opportunistic Network Environment

PMDCP: Proposed Message Drop Control with Priority

POI: Point of Interests

RAPID: Resource Allocation Protocol for Intentional DTN

SAD: Size Aware Drop

TCP/IP: Transmission Control Protocol/Internetworking protocol

TTL: Time to Live

WKT: Well Known Text

WLANs: Wireless Local Area Networks

Chapter 1

INTRODUCTION

1.1 Background of the Study

In the current fast and rapidly growing world of technologies, more and more businesses understand the advantages of usage of computer networking. A computer network is a system for communication between computers [1] i.e. it's a system that enables computers to communicate and share available resources. Initially it was used for a sharing of files and printers, but when applications are developed it has moved from a sharing of files and printers to application mutuality. In today's world wired and wireless network types are available.

Wired network is a category of network where connections are made through wired connection [2]. It uses networking cables to connect computers. This network has high bandwidth, low interference and better security than wireless networks since the transmission occurs through physical medium. But it's more expensive and gives low speed when compared to wireless networks.

Wireless network is a category of network where connections are established without the physical wired path [2, 3]. It uses radio frequencies to connect computers. Nowadays, the wireless network is chosen over wired due to low cost and mobility. But this network needs higher security than wired network because of the data is transmitted in the air. So, there is more chance of interception which can be improved by encryption technique in this network [2, 3].

They are two categories of wireless networks: Infrastructure and Infrastructure less networks (multi-hope) [1, 4].The infrastructure networks are the classification of wireless networks that contains wireless router/access point and enables other computers connect to it wirelessly. Typical applications of this type of network contain wireless local area networks (WLANs). The second type of wireless network i.e. infrastructure less mobile network, are commonly known as an ad-hoc network. They doesn't need fixed routers, all nodes have capablity of movement and can be connected dynamically in an random manner [1].

Delay tolerant networking (DTN) signify a class of infrastructure-less wireless systems that support the functionality of networks experiencing frequent and long-lasting partitions due to sparse distribution of nodes in the topology[5]. It, delivers an alternative approach to a variety of emerging wireless applications and architectures that challenge the limitations facing the transport and routing layers in the TCP/IP model. In the traditional Internet model which uses TCP/IP model there is an assumption of low error rates, low propagation delays and most importantly, a steady end-to-end connection between any source and destination pair of nodes. So, a class of challenged networks like terrestrial and military networks which violates one or more of those assumptions are being undeserved by the TCP/IP paradigm [5]. Those challenged networks are considered as a potential application for the DTN architecture.

DTN architecture signifies an attempt to extend the reach of networks [5]. It guarantees to act as an important platform between instances that originally adopt inconsistent or heterogeneous , even if they exist in territories lacking a proper communication infrastructure and to enable communication between instances of such challenged networks. The main purpose of the DTN approach is to deliver a means for message delivery in challenged settings [5] i.e. which violates assumption in TCP/IP paradigm.

DTNs can be characterized by lack of establishment of a path between any pair of nodes at a time, long delays, tolerates the intermittent connectivity, limited buffer space and bandwidth [5, 6]. Intermittent connectivity can be defined as an interruption occurred during transmission i.e. unexpected change of state (up/down) of any communication link between the nodes. This intermittent connectivity is occurred because of power management, mobility, node density, and limited radio range. DTN can also be defined as intermittently connected wireless ad-hoc network that makes the communication between the wireless nodes in the scenario where end to end connectivity can never be assumed or when the delay associated in relaying data could be very high[7]. It prevents data from being lost by using store-carry-forward approach. In this approach, messages are stored to their own storage until appropriate forwarding opportunity arises i.e., until the node encounters another node. When the node makes a communication link with another node, the node replicates the message to this node. This operation makes DTN as a solution for Physical distance and temporal discontinuity problems. But the combination of long-term storage and transmitting multiple copies of message in this process imposes greater transmission overhead and buffer occupancy. The routing packet of DTN is done hop by hop, since there is no continuous, two ways path between source and destination and the next hop is selected dynamically based on application scenario as well as the algorithm used.

DTN routing protocol can be broadly classified into flooding and forwarding Based [7].

Flooding based routing: This type of routing does not need any information about the network. However, sometimes knowledge about network is added as additional metric for improvement. In this routing strategy the number of copies of each message is transmitted to a group of nodes (sometimes called relays).The relays maintain and store the messages in their buffer until they connect to the next nodes.

Forwarding based routing: In this type of routing, the node does not need replication. Instead, information about network topology is used to select the best suitable relay nodes, and they hold a single-copy message and forward it to them. Forwarding based routing protocols are less resource consuming, but they don't achieve high delivery ratios. On the other hand, flooding based routing achieve a high delivery ratio but incur greater overhead.

Most conventional routing protocols proposed for DTN try to minimize overhead through strategic selection of relays [8] and limit the number of replicas of message [9] but it minimizes the delivery ratio. Many of those routing protocols do not consider the issue of resource limitations like buffer and bandwidth. But since many of DTN applications uses handheld devices which are having limited storage, performance degradation was occurred i.e., delivery ratio is minimized. Example Epidemic Routing [10] achieves high performance with infinite buffer but with limited buffer size the performance of this routing protocol is degraded since buffer over flow is occurred. To solve this problem Buffer management strategy is needed.

As stated above DTN store the message for long period until time-to-live (TTL) of the message is equal to zero. In this case, when buffer over flows occur, it is important to identify which message will be dropped to accept the new incoming message. And also in cases,finite bandwidth and short duration of contact between the nodes may not allow to transmit all messages, it is important to identify which message is transmitted first. Those are the main task of buffer management policy. Existing buffer management policies are categorized into three [6].Global buffer management policy: which utilizes network-wide information regarding all messages, local buffer management policy: which uses partial network knowledge and traditional buffer management policy: which considers only the order in which the message is stored in the buffer or don't consider any information.

Traditional buffer management policies perform poorly in DTN environments since they are very simple. Although, since global buffer management schemes utilizing network-wide information it's difficult to apply on the real world. All outperform the ones based on local information in aspects of optimizing specific network performance [6]. So, Since Policies based on partial network information out perform than the other we are motivated to propose local based Message Drop Control with Priority buffer management policy that considers three attributes (Number of hop count, Size and TTL) by compensating limitations of recently proposed local based policies. This Buffer Management policy is aimed to maximize delivery ratio and minimize overhead compared to other existing buffer management policies.

1.2 Statement of the Problem

In DTN, many routing protocols are flooding families i.e., they replicate the message to maximize the delivery ratio. However, this replication and storage of message for long period of time causes frequent buffer overflow due to limitation of buffer size. This implies there is a need of buffer management policy for DTN routing protocols which prioritize a messages and controls unnecessary message drops.

Over time, many buffer management policies are proposed by different authors but most of them only consider single information about the message without considering the effect of discarding the message on the performance of the network. and most of them don't consider unnecessary message drops. Considering only single information affect the performance of the network by minimizing delivery ratio. And also doesn't considering message drops increase the overhead ratio.Starting from this problem some authors proposed global based policies which considers global information but it's difficult to apply in the real world.

By putting those problems into consideration we decide that it needs a further improvement, and we propose Message Drop control with priority buffer management policy which considers local information's like hop count, TTL and size. Its aim is to maximize delivery ratio and minimize overhead by controlling unnecessary message drops with prioritization. This work is differentiated from the other by prioritizing a messages in parallel with controlling unnecessary message drops by combining different attributes.

To this end, this study attempts to explore and answer the following research questions:

1. How to combine different attributes to manage buffer effectively in DTN?
2. How to prioritize a message in parallel with controlling unnecessary message drops in DTN?
3. How to maximize delivery probability and minimize overhead occurred due to unnecessary message drops in DTN?

1.3 Objectives of the Study

1.3.1 General Objective

The Overall objective of this study is to propose a local based Message Drop Control with priority buffer management policy for DTN.

1.3.2 Specific Objectives

The specific objectives of this research are:

- To review different resources to understand the limitation of recently proposed buffer management policies for DTN.
- To design algorithm that satisfies better performance for routing protocols by considering Hop count, size and TTL.
- To implement an algorithm for the proposed model.
- Performing a simulation to check performance of the proposed model.
- Analyze and interpret the results of the simulation.

1.4 Methodology

To achieve the above stated objectives, the research work uses the following three concepts.

1. **Literature Review:**Reviewing different resources like books, research papers and other documents can be used to investigate existing system, identify the problem, examine the existing approaches and understand the mechanism how to achieve the proposed objectives related to buffer management for DTN. Techniques and approaches appropriate for the development of buffer management policy for DTN will also be reviewed.
2. **Analysis and Model:**After the initial literature review, in this phase, the model for the proposed solution is designed and its respective algorithm is developed for the activities involved in the proposed solution.
3. **Evaluation of the proposed work:**After designing the model, we test the effectiveness of our proposed buffer management policy and evaluate its performance related to its objective and contributions with Opportunistic Network Environment (ONE) simulator. We select ONE simulator because it combines mobility modeling, DTN routing and visualization in one package that is easily extensible and provides a rich set of reporting and analyzing modules[11]. Lastly the simulation result of the new policy is compared with existing policies and Evaluation is conducted by considering different parameters.

1.5 Scope and Limitation of the Study

The scope of this study is limited to designing and implementing Buffer management policy for DTN.The research is aimed to apply different routing principle or techniques to discover new Message drop Control with Priority buffer management policy to address the routing challenges of DTN. This policy will improve the performance of the existing routing protocols within DTN by satisfying better dropping and forwarding strategy for them. The study will address issues like delivery ratio and overhead.

The major limitation of the proposed model is it only considers management of buffer for DTN, it doesn't consider relay selection and security issues due to complexity nature of the implementation.

1.6 Application of Results

DTN provides improvement over other wireless networks in that it can tolerate intermittent connectivity and guarantee the data delivery even with very minimum knowledge of network. In other traditional networks if the communication link broken at any point of time of communication then the data is lost but in DTN, if the communication is in process and the linkis broken then the data get stored in each node's buffer rather than being lost. This is because, in DTN the message forwarding is through replicating the message to other nodes, i.e. the sender, relay and receiver nodes can have the copy of the same message stored in their buffer.

Considering the above advantage into account there are various real–life application areas that make use of the DTN concepts where, mobile or stationary wireless nodes, are focused to undergo extreme operational condition and wait for long interval of time. Some of these applications are Inter-planetary Communication, Wildlife Monitoring, Military Application and Deep Space Exploration. Since DTN follow store-carry and forward strategy, there is a high probability in which buffer over flow is occurred. The proposed policy manages a buffer of DTN routing protocols, mostly for flooding based to minimize overhead and maximize delivery ratio.

1.7 Organization of the Thesis

The thesis was organized as follows: In Chapter two we present a literature about research works on DTN regarding its architecture, the area of its application, routing protocols and related works pertaining to the research under consideration. Chapter three introduces the methodology employed, the proposed message drop control with priority buffer management policy framework and describe each component with the algorithm executed by each of them. Chapter four describes the implementation and the experimentation setup in detail. Chapter five presents the result of the simulation run and discussion. Results of the simulation are also analyzed and interpreted in this Chapter. The conclusion drawn from the result is briefly explained and future work in the area is presented in the last Chapter, Chapter six.

Chapter 2

LITERATURE REVIEW

2.1 History and Definition

Transmission has been very easy thing at current through the network using uniform sets of protocols (TCP/IP protocol). It is based on the existence of steady end to end path to allow a communication between source and destination .But it is still very problematic to transmit data in some networks that are often in interruption and delay . Meanwhile, network delay and disruption are very common due to various reasons such as the harsh environment or change in network topology . To address this concern, in 2003 Delay or Disruption Tolerant Networking (DTN) networks were introduced. Next, to research the DTN, Internet Research Task Force (IRTF) establish DTN Research Group (DTNRG) based on the Interplanetary Network Research Group (IPNRG)[12].

DTN is essentially deals with the occasionally-connected networks that may experience from frequent disconnection or partitioning. The networks which are suffered with disruption and disconnection, high or moderate delay and periodic connectivity because of environmental factors comes under DTN[13]. This arbitrarily long delay and frequent connectivity disruptions makes DTNs separate from traditional networks. It denotes there is no warranty that an end-to-end path between a given pair of nodes exists at a given point in time. Instead, connection or disconnection may occurred in the network due to a diversity of factors such as mobility, wireless channel impairments and nodes being turned off or running out of power. Therefore, in DTNs, the set of links connecting DTN nodes i.e. “contacts”, varies over time. This major difference between conventional networks and DTNs results in a paradigm change in the design of core networking functions such as forwarding, routing, congestion and flow control.

The DTN network architecture was proposed in 2007 [12]. It's purpose is to provide implementations for reliable message delivery in intermittently-connected networks. It presents the store-carry and forward paradigm, which is opposed to the Internet's store and forward, to deliver messages from source to destination. In store-carry and forward paradigm a node can store and carry data in its own buffer and forward these data to other connected nodes whenever they are available.But in traditional networks, i.e. store and forward approach, messages are only stored in the nodes before forwarding them; however, the time scales at which data is stored locally while waiting to be forwarded are very small when compared to DTNs. So, persistent storage is used on storage in store-carry and forward typically to imply that DTN nodes need to be organized accordingly.

This architecture was also designed to work as an intermediate layer, called the bundle layer which is above the transport layer and below application layer.It delivers services such as in-network data storage and re-transmission, inter-operable naming, authenticated forwarding and coarse-grained classes of service. The basic unit of data in this layer is called bundles or messages. DTN bundle layers communicate among themselves using simple sessions. This architecture can be applied to many types of challenged environments which have constraint like lack of infrastructure, disconnection, disruption and lack of resources[13]. The difference between TCP/IP layer and DTN layer is shown on figure 2.1

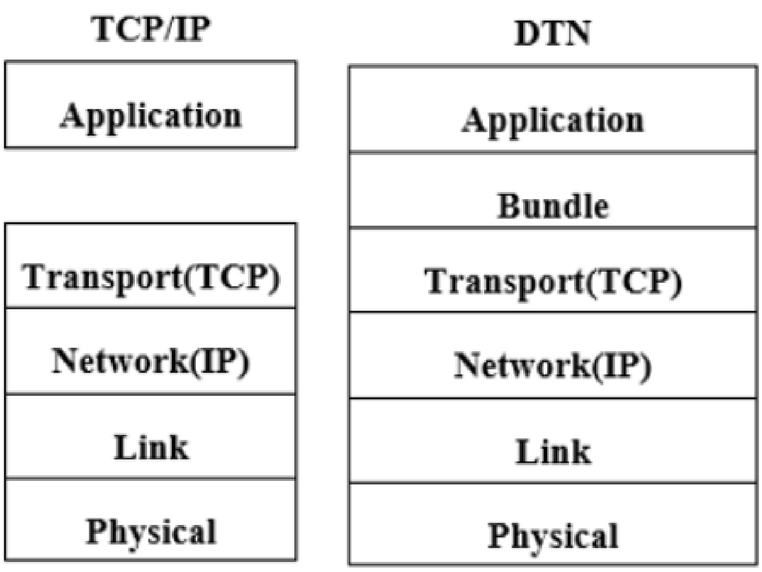

Figure 2.1: The difference between TCP/IP layer and DTN architecture [13]

The DTN architecture also specifies the bundle protocol[14]. This protocol controls the exchange of bundles, i.e., application-layer messages. The Bundle Protocol can work either above transport protocols like TCP, UDP, etc, or lower layer protocols like Bluetooth, Ethernet, etc. The term "bundle" was chosen to connote the self-sufficiency of the messages: application-layer messages are expected to contain sufficient meta data to enable processing by the recipient without negotiation. When working above the transport layer, the bundle protocol accepts messages from the application layer, compress them into bundles, and then transfers them to the next-hop DTN node. End points i.e the final destinations of bundles are known by EIDs (End Point Identifiers) which is associated with each nodes.

In DTN route , communication links are created whenever nodes come in range of one another, since end to end paths cannot be assumed. Contacts may be decided by their duration, capacity, latency, end points, and direction. And reliability is fulfilled hop-by-hop using custody transfers. A node accepting custody of a message obligates to send that message to another node that accepts the message's custody or its destination . Once the node reliably transfers the message to the message's next custodian, its duty as custodian for that message stops. Custody transfers makes nodes to free up storage used to carry messages that get transfer to other nodes [14].

According to the store-carry and forward approach, when a DTN node comes in communication range with another DTN node, it decides whether to forward messages it is carrying to the other node. So, the concept of links in traditional networks (wired or wireless) is substituted with the notion of contacts. Some times there is a case in which a node may need to choose which messages to forward based on some priority and may also decide whether the new neighbor is a "good" candidate to carry its messages since, contact times are finite and may be arbitrarily short.

In DTNs, the Internet model of end-to-end reliability is replaced by custody transfer, due to the failure to guarantee end-to-end connectivity. And also hop-by-hop acknowledgments are used by custody transfer to identify the correct receiver of messages between two directly connected nodes. Functions based on the TCP/IP model, like congestion and flow control will not always work in DTNs.Instead, hop-by-hop control can be created.

2.2 Characteristics of DTN

Following are the most important characteristics of DTN [15]

- **Tolerates intermittent connectivity:** It can be defined as inconsistent connectivity occurred due to nodes movement or the unexpected change of state (Ups/Downs) of any communication link. DTN tolerates the intermittent connectivity since it supports the buffer and custody transfer concept.

- **Tolerates High latency:**The communication is supported in DTN even when the delay in occurrence of next contact could be very high. so, the messages are kept in buffer memory of the sending node until the next node will come in contact.

- **Tolerates asymmetric data rates:** The data can be transferred between the nodes in DTN , even when, the rates of incoming and outgoing transfer of messages are different.

- **Long queuing delay:** since disconnection problem is high in DTN when compared to the conventional network, the queuing delay i.e. The time it takes to drain the queue messages ahead of the tagged on, may be extremely large in the worst cases e.g. Minutes, hours, and days.

- **Prevent data loss:** In DTN, since every node has provided with some amount of buffer memory that is used to make persistent storage data can be prevented from loss. The node can remove the messages, only when the custody of messages have been transferred to other node.

- **Store-Carry and Forward approach:** DTN works on the Store-Carry and Forward approach, which makes the use of the Bundle Protocol. Store-Carry and Forward approach makes the DTN nodes to receive the message, store it in buffer, carry it to the relay or destination node and forward it. If the relay node is not exist , then they save it in buffer.

2.3 Applications and Issues of DTN

There are many real-life applications where wireless mobile or stationary nodes, are forced to undergo extreme operational condition or wait for extended intervals of time that exceed traditional IP forwarding times (that are usually measured in milliseconds) before being able to transfer their data to next hops[16]. Some of those applications are described as follows:

2.3.1 Applications of DTN

Some applications used in DTN are described below [13]

- **Interplanetary Internet:** Interplanetary communication is the extreme cases in which DTN can be applied. The DTN application of interplanetary network beats the traditional border of TCP. The huge space separating global artificial purposes restrict the existing method to swap data among them or with base-stations on earth.

- **WSN with DTN architecture:** Wireless Sensor-based Networks in which nodes perform their function with low power or weak or intermittent radio communication is follow DTN architecture. In these types of networks bundle layer are implemented on some node which act as DTN nodes.

- **Terrestrial Mobile Networks:** These are the networks which cannot ordinarily maintain end-to-end connectivity and connect mobile devices. These networks may become unexpectedly disconnected due to variable signal strength or node mobility , while others may be partitioned in a periodic, predictable manner like satellite.

- **Military Ad-Hoc Networks:** These networks work in environments where factors like mobility environmental, intentional jamming may be cause for disconnection. Reconnaissance operation using (Unmanned Aerial Vehicle) where the goal is to monitor a specific area and to report all such activities happens at that particular area can be taken as an example.

2.3.2 Issues in DTN

The above mentioned characteristics of DTN impact the design of routing rule and create the following routing issues [16].

- **Buffer Space:** In DTN network suffer from long disconnection due to packets are stored for long period in each node buffer. So that, they need enough buffer space to store all message that are waiting for communication opportunities. So, if buffer space of node is limited, the node buffer will be overflowed.

- **Energy:** The energy is a key problem in delay tolerant network since nodes in network may have limited energy supplies due to either mobility or dis-connectivity. Routing in DTN wastes large amount of energy by sending, receiving, storing and computation process than conventional routing technique.

- **Encounter Schedule:** When nodes in DTN transfers the data from source to destination , it can wait till it encounter the destination node and after that transfer the packet by direct delivery to the destination. This process may take long time or not happen as DTN suffer from dis-connectivity problem, each node in the network try to communicate when opportunistic contact is available. The encounter schedule is very key issue in Delay Tolerant Network. Since the delivery of messages is straightly depends upon the schedule of the encounter.

- **Resource Allocation:** Resource allocation is a key problem in DTN. As discussed in the above topics Delay tolerant Network work in stressful environment where there is an absence of end to end connection. The main objective of DTN is maximize message delivery ratio in parallel with minimizing resource consumption (overhead ratio) which are clash with each other. For example to maximize packet delivery ratio from source to destination the best way is to distribute the multiple copies of the message in the network but it needs more space.

- **Reliability:** Reliable delivery of packet can be achieved by ensuring the successful and steady delivery of packet by any routing protocol that has some acknowledgment. When a messages reach to the destination, some accepted message should be sent back from destination to source.

- **Security:** It is a major problem for any network either traditional or DTN. Before transferred to its final destination, a message may be passed randomly through the intermediate hosts. Based upon the application's security, users are needed a secure guarantee about the message authenticity. Cryptographic technique is beneficial for a secure intermediate routing.

2.4 Routing Protocols in DTN

Routing itself is a way of transferring data from source to destination. The characteristics of DTN require the adoption of techniques to counter and wisely utilize resources. To perform efficient routing in DTN, the nodes need to use any available information about a future link establishments, exploit mobility, and wisely allocate buffer resources to save power. Also, security and reliability of the data is an important challenging issue in DTN. Since those characteristics differentiate DTN from Mobile Ad-hoc Networks (MANET) the routing protocols for MANET doesn't used for DTN [17].

As shown on figure 2.2 routing protocols in DTN are classified into 2 categories based on the property they used to find the destination: **Flooding based and Forwarding based.** To find the estimation, two different approaches namely replication and knowledge are used. The flooding strategy uses replication and there are many algorithms to make multiple copies of message and manage those copies. Whereas forwarding strategy uses knowledge and some works have been devoted to develop more efficient methods to obtain some network state information and then use it to make routing decision [18].

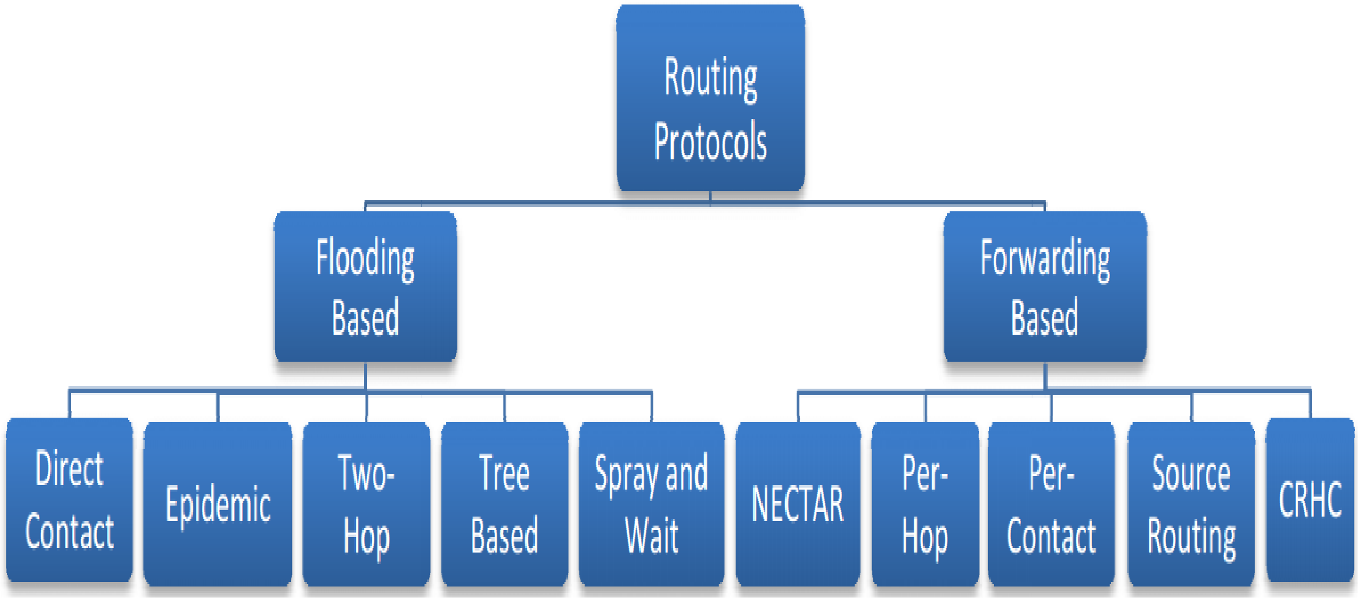

Figure 2.2: Categories of DTN routing approaches [7]

2.4.1 Flooding Based Routing

In flooding based routing, each node transfers a multiple copies of each message to a set of nodes (sometime called relays). All relays keep the copies and store them in their own buffer until they connect with the next hops. Many of existing works in the area of DTN routing are category of flooding families. Using the message replication can improve performance of the network by increasing the probability of message delivery, but it consumes more network resources. The Flooding based routing is further categorized into two types [7]:

- **Replication Based:** Replication based routing makes the nodes in the network to create the copies of the received message. The maximum number of copies generated within a network for a specific message could be n-1, where n signifies the number of nodes in the network.
- **Quota Based:**In Quota based routing the number of copies for each message is limited.

Some examples of flooding based routing are discussed below.

Direct contact

In Direct contact [19] routing algorithm data can be transmitted in one hop only, since exactly one message transmission is used when the source can directly contact with the destination. It doesn't require any network information but it requires direct path between source and destination. Hence, if no contact occurs, message is not delivered to it's destination.

Epidemic routing

In Epidemic [10] all nodes are made relay nodes, and they exchange a message with each other and message will be transmitted to destination. Messages can be delivered with a high probability in epidemic, but it consumes network resources like bandwidth, power and buffer space heavily.

Two-hop relay

In this approach[18], the source node replicates the message to many relay nodes. The transmission in this routing protocol is performed by using two hops between source and destination. If there are n number of nodes around and directly connect with the source, then equal number of (n) copies of the message should be generated from source and be transmitted to these nodes.

Tree-based flooding

In this routing protocol [7], the method of flooding is based on tree structure. It works upon the concept that the source node must be limit the number of replicas to Nc -1. When the number of copy of nodes are limited then they can go in depth up to a certain level. Each node can have maximum of two child nodes so the replicas are equally spread in between them. After this receiving phase, the nodes start offloading the message to collection stations to reach the destination. A strategy limits the number of hops that could be traveled hence limiting trees depth, but not width. Key issues in this routing protocol are making decision on how to make copies and ensuring the number of copies.

Spray and wait routing protocol

The Spray and Wait [9] algorithm is the higher version of the epidemic routing. In this algorithm the nodes are not distributing the replicas to each and every node that achieves resource efficiency but an optimal number of nodes (say m) are selected to which the source node will relay the message. There are two phases in this method:**Spray and Wait**.In Spray phase, the source node transmits the packets to the m nodes and these m nodes will further transfers them to m relay nodes. If the destination is not found in spray phase then the relay nodes will store the message and transmits the messages directly to the destination.

2.4.2 Forwarding Based Routing

Forwarding Based routing uses the knowledge about the node in the network. It's used to select the best path and then the message is forwarded from node to node through the selected path. This approach minimizes the extra resource consumption as the replication of messages are not allowable. It is used when the resources are limited. NECTAR, source routing, per-hop routing, per-contact routing and hierarchical routing protocols are categorized under forwarding based routing.

NECTAR

The NECTAR [7] algorithm uses the concept of neighborhood index table that keeps the information about the meeting frequency of the node with all other node in the network and maintained at each of the node. In this algorithm, a higher index value will be assigned to a node which has higher meeting frequency and when a node wants to forward the message to a specific destination, the node that have highest index value will be chosen.

Per-Hop Routing

In Per-Hop routing [7], when a message transferred to the intermediate node, they made forwarding decision. Each intermediate node determines the next hop to which the packet is to be forwarded for the destination and keeps it in a queue for that contact. It has better performance than source routing since it uses the more updated information.

Per-Contact Routing

In Per-contact routing [18], instead of computing the next hop for a message, each time a contact is available, the routing table is recomputed . so the nodes use the most updated information. the intermediate node checks the current active contacts and select the appropriate node for relaying the message and then forward to those active contacts, when any intermediate nodes receives the message. But, to guarantee the loop freedom is a big problem.

Source Routing

The Source routing[20] means the source node is in charge of the whole transmitting and determines the path based on the topology of the network before the message gets into the node. It contains two phases: route discovery phase and route maintenance phase. Primarily a route is discovered by sending control packets towards a destination node. Each of the intermediate nodes will add its address in the packet and also maintains a cache for the routes that the node has learnt over time. When the packet is delivered to the destination the entire route is appended in the packet only. In second (route maintenance phase) if a link failure is identified then a route error message is broadcasted by the source node. This protocol will show good performance only when the distance between source and destination is short.

Hierarchical Forwarding and Cluster Control Routing

This routing [7] presents the concept of grouping (i.e. clustering) of nodes based on link property and communication characteristics. After information of clusters, a cluster head is chosen depend upon some criteria. Then the routing decisions are then taken by this cluster head.

2.5 Types of Contacts in DTN

DTN architecture explains different types of contacts between network nodes.Five types of contacts explicitly, persistent, on-demand, intermittent-scheduled, intermittent-opportunistic and intermittent-predicted that may exist in DTN are described by Request for comments (RFC) of DTN [21].

2.5.1 Persistent Contacts

Persistent contacts are a type of contacts were no connection start up action is required to instantiate (i.e., always available). An 'always-on' Internet connection such as a DSL or Cable Modem connection would be a type of this class.

2.5.2 On-Demand Contacts

On-Demand contacts need some action to instantiate, but then they perform their function as persistent contacts until ended.one example of an On-Demand contact is a dial-up connection (at least, from the perespective of the dialer; it may be viewed as an Opportunistic Contact, below, from the viewpoint of the dial-up service provider).

2.5.3 Intermittent - Scheduled Contacts

A scheduled contact is an agreement to establish a contact at a specific time, for a specific duration. A link with a low-earth orbiting satellite is an example of a scheduled contact. The satellite's schedule of view times, capacities, and latencies builts a node's list of contacts. Note that, the notion of the particular time is delay dependent for networks with substantial delays.

2.5.4 Intermittent - Opportunistic Contacts

Opportunistic contacts present themselves unexpectedly as opposed to scheduled contacts. A good example for this type of contacts is an unscheduled aircraft flying overhead and beaconing, advertising its availability for communication. An infrared or Bluetooth communication link between a personal digital assistant (PDA) and a kiosk at an airport concourse can be taken as another type of opportunistic contact. The opportunistic contact begins as the PDA is brought near the kiosk, lasting an unknown amount of time (i.e., until the link is lost or terminated).

2.5.5 Intermittent - Predicted Contacts

Those contacts are based on no fixed schedule, but rather are predictions of likely contact times and duration's based on a history of previously observed contacts or some other information. When is assumed as a great enough confidence in a predicted contact, routes may be chosen based on this information. From the contact types discussed above, the researcher of this study considered the Intermittent Opportunistic Contacts for the proposed policy.

2.6 Buffer Management in DTN

Owing to the characteristics of DTN remarked in section 2.2, the design of Delay Tolerant Networks creates the issue of how to effectively deliver data based only on occasional encountering of nodes, where the conventional routing schemes do not work properly [22]. In the study of DTN the store-carry and forward approaches have two major problems: firstly forwarding of messages to any encountered node is not feasible and secondly nodes buffer space gets filled up quickly [6].

The first problem (forwarding of messages to any encountered node is not feasible problem) is occurred in DTN since forwarding to many nodes puts a burden on limited node resources like buffer, energy, network bandwidth. Also it creates to network contention because of the flooding of messages. So, nodes to which message has to be forwarded should be chosen carefully. This problem is addressed through design of routing protocols. The second problem arises because of limited buffer space available at DTN nodes. Since DTN nodes work on store-carry and forward mechanism i.e. a message is stored for long period of time until the time to live of the message is expired or encounter node comes into the network, their buffer space gets filled up quickly. In this situation where each node buffer is full with message from other nodes in the network and a new encounter happens, either the node has to drop a message from its buffer or it may have to deny the sending node of a new message transfer which minimizes the delivery ratio. This problem is solved by buffer management policies in DTN.

Buffer Management technology is a fundamental approach that manages the several resources among different situations as per the technique used [23]. It is very important in DTN since the combination of long-term storage of messages and the message replication places a high bandwidth and storage overhead on nodes. When the buffer is full, often in order to receive a new message a DTN node will have to drop an important message. But if efficient dropping policy that can prioritize message drop sequence is implemented, it will have a great impact on the delivery ratio of the network. Only dropping policies don't define buffer management in DTN, but also the scheduling policies. Since DTN suffers from partial transmissions, sorting the messages in the order of their relevance is very important for their successful delivery. Partial transmissions occur when two nodes are transmitting the messages and their transmission has to be aborted because of some link failure or power shut down by either of the nodes. Thus, both scheduling and dropping policies play a huge role in the delivery performance of a DTN network [6].

Basically buffer management policies can be classified into three based on information used by them [6]: **Global, Local and Traditional buffer management policies**. Buffer management policy which utilize network-wide information regarding all messages in the network are categorized under global buffer management policy. But buffer management policies which uses partial network knowledge, rather than all network-wide information related with messages and additional message properties like remaining TTL and size are categorized under local buffer management policies. And buffer management policies like drop head, drop tail and drop random which doesn't consider any information i.e drop randomly or only considers the order in which message was stored in the buffer are categorized under traditional buffer management policies. We see in detail in the related work section.

Generally the primary and necessary goal of buffer management policy must be to improve the delivery ratio of the network [6]. In addition to improving delivery ratio efficient buffer management policy try to minimize overhead ratio by minimizing the number of relays to which a message is forwarded and controlling redundancy i.e. controlling number of message dropped to accept a new incoming message.

2.7 Related Works

To solve the problems related to buffer over load researchers have proposed a number of buffer management schemes that can be adopted by various DTN applications. As we see in section 2.6 we can categorize DTN buffer management policies into three i.e. traditional, global, and local buffer management policies based on information used by them.

2.7.1 Traditional Buffer Management Policies

Algorithms that only consider the order in which the message arrive without considering any particular message attribute to transmit and drop the messages are categorized under traditional buffer management policy. Examples of Those dropping algorithms are (1) Drop Front:- the main idea of this algorithm is dropping the message from the head of queue i.e. the first message which enters into the queue will be removed first. (2) Drop last:- its main idea is dropping the message from the tail i.e. the last message which enters into the queue will be removed first. (3) Drop Random:- It simply discards the message randomly[23]. However, as stated on [6,24] those algorithms are not suitable i.e. perform poorly in DTN since they are very simple and maximize the drop. Most probably they were designed for fully connected networks.

On [25]diverse traditional buffer dropping policies with the following strategies: first in first out policy, evict the most forwarded first (MOFO) policy, evict the most favorably forwarded first policy, evict the shortest lifetime first (SHLI) policy and evict the least probable first policy are also evaluated. They concluded that concerning message deliveries the MOFO is the best buffering policy and in terms of the average delay the SHLI is the best. Those policies are easy to implement as long as the buffer size on all hosts is larger than the expected number of messages in transit at any given time and very reasonable. However, whenever the available buffer size is limited in relation to the number of messages, these policies perform poorly in a DTN environment [24].

Similarly, in addition to dropping policies a number of scheduling policies are proposed, First Come First Serve(FCFS) is the simple policy which is easy to implement. When the contact duration is long enough to transmit all messages a node has, FCFS is a very reasonable policy. However if the contact duration s limited, FCFS is sub-optimal as it does not provide any mechanism for preferential delivering of high priority messages[26].

2.7.2 Global Buffer Management Policies

Rather than traditional buffer management policy global Buffer management policies utilize network-wide information regarding all messages to improve performance of routing protocols in DTN.

Resource Allocation Protocol for Intentional DTN (RAPID) [27] is the first work that takes into account network wide information. RAPID “routes” a packet by opportunistically replicating it until a copy reaches the destination. It converts the routing metric to per-packet utilities that determine at every transfer opportunity if the resources used are justified by the marginal utility of replicating a packet. RAPID loosely tracks network resources through a control plane to adjust a local view of global network state. In RAPID an in-band control channel is used to exchange network state information between nodes using a fraction of the available bandwidth. This protocol handles DTN routing as a resource allocation problem that translates the routing metric into per-message utilities, which determine the order in which messages are replicated and dropped under resource constraints. But, Rapid’s utility formulation is sub optimal as it does not take into account nodes buffer state.

Krifa Amir, BarakatChadi [28] develop a theoretical framework based on Epidemic message dissemination that takes into account all information that are relevant for message delivery. Based on this theory they proposed a policy called the global knowledge-based scheduling drop, which is based on global knowledge about the network which is difficult to implement in practice. To amend this difficulty they also proposed a distributed algorithm called the history-based drop (HBD), which, uses statistical learning from network history to estimate the global knowledge needed by the global knowledge-based drop policy. This policy shows higher performance than traditional policies in terms of the message delivery rate and the average delay. But, the utility is computed under the assumption of homogeneous node mobility in which node pairs have the same meeting rate and also the performance of their scheme may degrade in environments where node mobility is heterogeneous. Furthermore, the author’s also assumed unlimited bandwidth and all messages have the same size which are uncommon in practice.

On [29] an adaptive optimal buffer management scheme to match node mobility characteristic for the realistic DTN was proposed. It uses the historical mobility information of all nodes in the network, to adjust the nodes mobility model adaptively. This policy expands the HBD algorithm and extends its applicability to situations where the bandwidth is limited and messages vary in size. But, they did not address the message scheduling issue nor provide any experimental results to validate their scheme.

Tuan Le, HaikKalantarian, Mario Gerla [30] proposed a novel utility function based on power-law distributed contacts to guide the scheduling and drop of messages. They develop utility function using network-wide information such as the number of existing copies of each message in the network and the distribution of pair-wise inter-contact times between nodes to compute per packet average delay utility. Then, messages are scheduled in decreasing order of their utilities and dropped in increasing order of their utilities. In this policy nodes should never discard its own source message to ensure that at least one copy of each message stays in the network and this optimization aimed to maximize its deliver ratio. Using ONE simulator the authors compare the performance this policy with traditional policies and show that this policy maximize the delivery ratio and minimize delivery delay. But, they assume homogeneous message size which isn't real.

2.7.3 Local Buffer Management Policies

Those policies uses partial network knowledge like number of copies of message in the network, instead of all network-wide information correlated with messages.

On [31] a local effective buffer management drop policy Equal-Drop (E-Drop) was proposed. In this policy when congestion is occurred the message which have equal or greater size with a new incoming message is dropped. The goal of E-DROP policy is to decrease the metrics of relayed, dropped, average latency, overhead ratio, and hop count and to increase the average delivery probability. Using simulations support, the authors show that E-Drop buffer management with random message sizes drop policy performs better as the existing MOFO. This policy controls unnecessary message drop in the case of there is a message which has equal or greater than the new incoming message is occurred in the buffer. However, this policy have some limitations like when all messages stored in the buffer have smaller size than the new incoming message there is no drop i.e. the new incoming message doesn't accepted. And also, they don't consider the buffer state and the situation when there is more than one message which have equal size is stored in the buffer and they consider only size of message which have direct impact on the delivery probability.

MominaMoetesum, FazleHadi [32] presents Size-Aware Drop (SAD), an adaptive and efficient buffer management scheme for highly congested resource-constrained DTNs. SAD improves some limitation of E-Drop by determining the exact requirement from a congested buffer and then picks an appropriate sized message to be discarded. This prevents excessive message drops by considering message size, but they don't consider a situation like when there is more than one equal size message in the buffer, and also considering only message size to drop a message by taking unrealistic assumption i.e. all messages have the same priority to be discarded may affect delivery probability. Example: when a message which has equal size with the coming message has long TTL.

Youssef Harrati, AbdelmounaimAbdali[33] have proposed a MaxHopCount dropping policy which consider hop count i.e. the number of nodes the message has crossed during its path from source to the current node. They assume that since a high hop count means that the message crossed lot of node, and then there may be lot of copies at the network, so dropping this message from the buffer may not affect its delivery. Otherwise, since a low hop count means low number of replicas at the network so dropping this message may lessen the delivery probability. Based on hop count this policy drops a message which has high hop count which doesn't affect the delivery probability. This Policy is optimal, but it doesn't consider the issue of unnecessary message drops. since dropping a message which have maximum hop count may not used to accept the new incoming message and this maximize the overhead. In addition this they didn't consider the situation's like equal number of hops.

CC Sobin[34] has also proposed Buffer Management Policy for DTN which is based on TTL and hop-count. In this policy Messages which have the lowest hop-count are scheduled to be transmitted first. Since, low hop-count depicts that the message hasn't traveled far from the source and is still far from the destination and also the message which has higher hop-counts and low TTL are the first to be dropped. Because, these messages have low probability to reach the destination from the current node because of low TTL and since their hop count is high, it is safe to assume that they have been sufficiently spread into the network that one of the copies will reach the destination if the current copy is dropped. But this policy also doesn't consider the size of message which is used to control unnecessary message drops and situation's like equal hop count in scheduling policy.

As stated on[6], from the above related work we can understand that traditional based buffer management policies perform poorly and global buffer management policies are difficult to apply practically since they need network wide information in DTN environment. However, using local information's are better than the others. As a result of this, we proposed Message Drop Control with Priority buffer management policy that combines advantages of the above local based algorithms by considering hop-count, TTL and size of the message. The following chapter discuss details of the proposed buffer management policy.

Chapter 3

PROPOSED APPROACH

3.1 Introduction

As we have described under chapter 2 most of the current buffer management focus on either prioritizing a message or controlling unnecessary message drops. There are no policies which prioritize a message in parallel with controlling unnecessary message drops. So, to achieve the objectives, the idea of this research is to enhance the buffer management policies for routing protocols in Delay-tolerant network because they follow store-carry and forward approach and most of them are flooding based to achieve a high delivery ratio. Therefore, buffer overflow is a known problem in DTN. Thus, we proposed a new Message Drop Control with Priority buffer management policy that prioritize a message in parallel with controlling unnecessary message drops by considering local information's which is compatible with any DTN routing protocol.

3.2 The Proposed Message Drop Control with Priority Buffer Management Policy for DTN

From the related work of literature review, it is understood that several research has been done in proposing efficient buffer management policy for DTN routing protocols to drop or transmit the message based on some mechanisms. But, in this study, the buffer management policy that control unnecessary message drops in parallel with prioritization was proposed. The proposed policy considers only local information's i.e hop count, size and time to live since using local information is better than the other as stated under section 2.7.

When there is a contact between two nodes first, the summary vector (packets IDs in a nodes buffer) are exchanged between them. And then the proposed approach logically splits buffer of each node into two by calculating the threshold value. To set the threshold value the adaptive approach on [35]was used since setting the threshold statically would be arbitrary, and would not work for all environments.After the buffer is separated into two based on the threshold value, the messages in the buffer are sorted according to their hop count and time-to-live for transmission. All data packets that are stored in one node and not in the other are ordered according to the proposed message scheduler to be transmitted to the other node.Packet transfer then starts from the sender to the receiver node until the contact duration ends. Then at the receiver node the buffer checker computes the size of incoming message with its available buffer space.

If the available buffer space is greater than the size of new incoming messages, the receiving node accepts the new incoming messages. However, if the size of available buffer space is less than the size of incoming messages, the buffer checker either drop the new incoming message or invoke message dropper to drop the messages from the buffer based on the priority. This process will continue until the computed available free buffer space of the receiving node is equal or greater than the size of new incoming message.This approach maximizes the delivery probability and minimizes the overhead.

3.2.1 The Threshold Calculation

The proposed buffer management policy uses adaptive approach on [35]to set the threshold. In environments where the average number of bytes transferred per transfer opportunity, x, is much smaller than buffer size, b, x is used as value of portion. As x grows, the threshold is reduced to the difference between the two values. When x is larger than the buffer size, then the threshold is completely removed. Specifically, after each transfer opportunity, were evaluating the threshold by first choosing a portion of the buffer p. After the portion is chosen, the threshold is set to be the minimum hop count that selects the packets in p.

3.2.2 Message Scheduler

The order in which the messages are transmitted is significant in cases short duration of contact between the nodes and finite bandwidth may not allow the node to transmit all the messages that are available in the buffer and also to make fair distribution of message in the network. Because of the above reasons the messages stored in the nodes buffer should have to be scheduled before transmission start when another node encounters. In the proposed and some existing policies a higher priority is given for lower hop count. This is used to give a higher transfer opportunities for the newer packets. So in the proposed buffer management policy, the messages in the buffer are sorted by their hop count and time-to-live. Assume if there are two messages Mi and Mj that are stored in one and not in the other node and if Mi have hop count less than threshold and Mj have hop count greater than threshold, in this case Mi is transmitted first and vice versa. But if both messages have hop count less than threshold or greater than threshold the messages which have higher time-to-live are transmitted first. The algorithmic design of the message scheduler is shown in algorithm 1.

Algorithm 1 Algorithm for Message scheduler

Input: hop count of messages in the buffer (Mhc) ,time to live of messages stored in the buffer(Mttl) , threshold value(tr)

Output:transmit buffered messages based on their order

Process:

```
for each message M(i,j) in B(n) do
  if (Mihc < tr And Mjhc > tr) then
    transmit message Mi
  else if (Mihc > tr And Mjhc < tr) then
    transmit message Mj
  else if (Mihc < tr And Mjhc <tr || Mihc > tr And Mjhc > tr) then
    if (Mittl > Mjttl) then
      transmit message Mi
    else
      transmit message Mj
    end if
  end if
end for
```

3.2.3 Buffer Checker

By the benefit of the periodical broadcast of beacon, when two nodes are in contact the connection will be established. Based on the established connection, packets IDs in a node's buffer are exchanged between them. And if there is a message that are stored in one and not in the other node those messages are transmitted according to the message scheduler. In this case, when a transmission is started first, the buffer checker at the receiver side computes the size of incoming message with the total buffer size. If the size of new incoming message is greater than total buffer size of a node, the new incoming message is dropped because the message is too large and if its size is less than total buffer size, the available free buffer space of that node was calculated. The available free buffer space can be calculated as Equation 3.1.

Equation 3.1: The amount of available free buffer space

$$FBs = Bs - \sum_{i=1}^{n} MBi \tag{3.1}$$

Where: FBs: is available free buffer space

BS: is the size of the buffer

MBi: is Messages stored in the buffer

If the available free buffer space is greater than the new incoming message size that node accepts the message and stores it in its own buffer. Otherwise that node compares the hop count of incoming message with the hop count of messages stored in the buffer. If the hop count of new incoming message is higher than hop count of all messages stored in the buffer and the node is not the destination of the new incoming message, the new incoming message is dropped since it is assumed that the new incoming message have been sufficiently spread into the network and one of the copies will reach the destination if the current copy will be dropped. If not, this node invoke message dropper to make a room for new incoming message. The algorithmic design of the buffer checker is shown in algorithm 2.

Algorithm 2 Algorithm for buffer checker

Input : Total buffer size(Tbs), the available free buffer space(FBs) , the size of new incoming message(NMs), the hopcount of new incoming message(NMhc), hop count of total messages in the buffer(TMhc)

Output : Drop the new message or Accept the new message or invoke message dropper procedure

process

```
if ( NMs > Tbs) then
   Node drop NewMsg
else if ( FBs >= NMs ) then
   Node Insert(NewMsg)
else if (FBs < NMs) then
   if (NMhc > TMhc And NewMsgdest is not this node ) then
      Node drop NewMsg
   else
      Invoke message dropper
   end if
end if
```

3.2.4 Message Dropper

The existing work on buffer management drops the message in case buffer overflow occurred to accommodate a new message based on some priority. However, simply drop the message on buffer status is not efficient because when the messages are dropped in DTN it has huge impact on delivery ratio of the network. So the messages should be dropped if and only if there is insufficient buffer space to accept the new incoming messages and the new incoming message is important for that node. Because of the above reason the message selection for dropping must be decided carefully to improve delivery ratio and minimize overhead. To do this, as stated on section 2.7 the nodes make a decision based on local information or network wide information.

As stated on [6], since using local information is better than the other, in proposed approach the message dropper at each node use the local information to drop a message from buffer and make a room for new incoming message when it's necessary. Three metrics namely messages hop count, time to live and size are considered to drop a message from the buffer. The message with high hop count (the message which has hop count greater than threshold) have a high priority to be dropped since their hop count is high, it is safe to assume that they have been sufficiently spread into the network that one of the copies will reach the destination if the current copy is dropped. In addition to hop count, to avoid unnecessary message drops and minimize volume of drops this policy determines the exact requirement needed to accept the new incoming message by considering the size of the new incoming message. Exact requirement is calculated as follows. Equation 3.2: The amount of exact requirement

$$Er = NMs - FBs \tag{3.2}$$

Where: Er: is exact buffer space required to accept new incoming message.

NMs: is size of new message

FBs:is available free buffer space

After we get the value of exact buffer space required (Er), from the messages stored in the buffer which have hop count greater than threshold, first this policy finds the messages which have equal size with the exact buffer space required (Er) and drop that one. In case exact match doesn't exist, a node drops the message which has the least great size than the exact buffer space needed and replaces it with the new. Otherwise, it finds the minimum number of message discarded to accept the new incoming message. And also if there is a case in which there are more than one messages which have equal size with the exact requirement value or with each other the proposed policy prioritize them by comparing their time-to-live value and the message which has less time-to-live value is dropped first. This mechanism is used to control unnecessary message drops to minimize overhead. In addition to this if there is a case in which the size of all messages which have hop count greater threshold is less than the exact requirement needed to accept the new incoming message. The policy drops the message with low time to live.The algorithmic design of the message dropper is shown in algorithm 3.

Algorithm 3 Algorithm for Message Dropper

Triggered : When new message arrives and buffer overflow occurs

Input : Hop count(hc) , threshold value(tr) , size of the messages(Msize(i,j. . .)), Exact requirement(Er), time to live of the messages (Mttl(i,j. . .))

Output : Drop buffered message

Process

```
Calculate the exact requirement
Select the message M(i,j) in node n which have hc > tr
for each message M in node N which have hc > tr do
  if (Misize == Mjsize == Er) then
    if ( Mittl < Mjtt1) then
      drop Mi and insert new message
    else
      drop Mj and insert new message
    end if
  else if (Misize == Er And Mjsize > Er || Mjsize < Er) then
    drop Mi and insert new message
  else if (Misize > Er And Mjsize < Er) then
    drop Mi and insert new message
  else if (Misize > Er And Mjsize > Er) then
    if (Misize > Mjsize) then
      drop Mj and insert new message
    else
      Drop Mi
    end if
  else
    Find minimum number of messages to be discarded and insert new message
  end if
end for
if (total size of messages which have hc > tr are < Er) then
  drop message which have minimum TTL and insert new message
end if
```

3.3 General Operation of the Proposed Buffer Management Policy

In this section the above algorithms are combined and the overall pseudo code of the proposed buffer management policy was written as follows and the flow chart is shown on figure 3.1.

3.1 Pseudo code of the proposed buffer management policy

Triggered when node i contacts node j

1. Begin
2. Calculate threshold and split the buffer into two based on threshold value
3. Sort the messages in the buffer for transmission according to the message scheduler
4. For each message M in Buffer B
5. Replicate M in sender node according to their order
6. At the receiver node the buffer checker makes decision on the new incoming message either it will be accepted or dropped.
7. If the buffer checker decides to accept the new message and the buffer at receiver node is full
8. Delete the messages in the receiver node buffer according to the message dropper
9. End

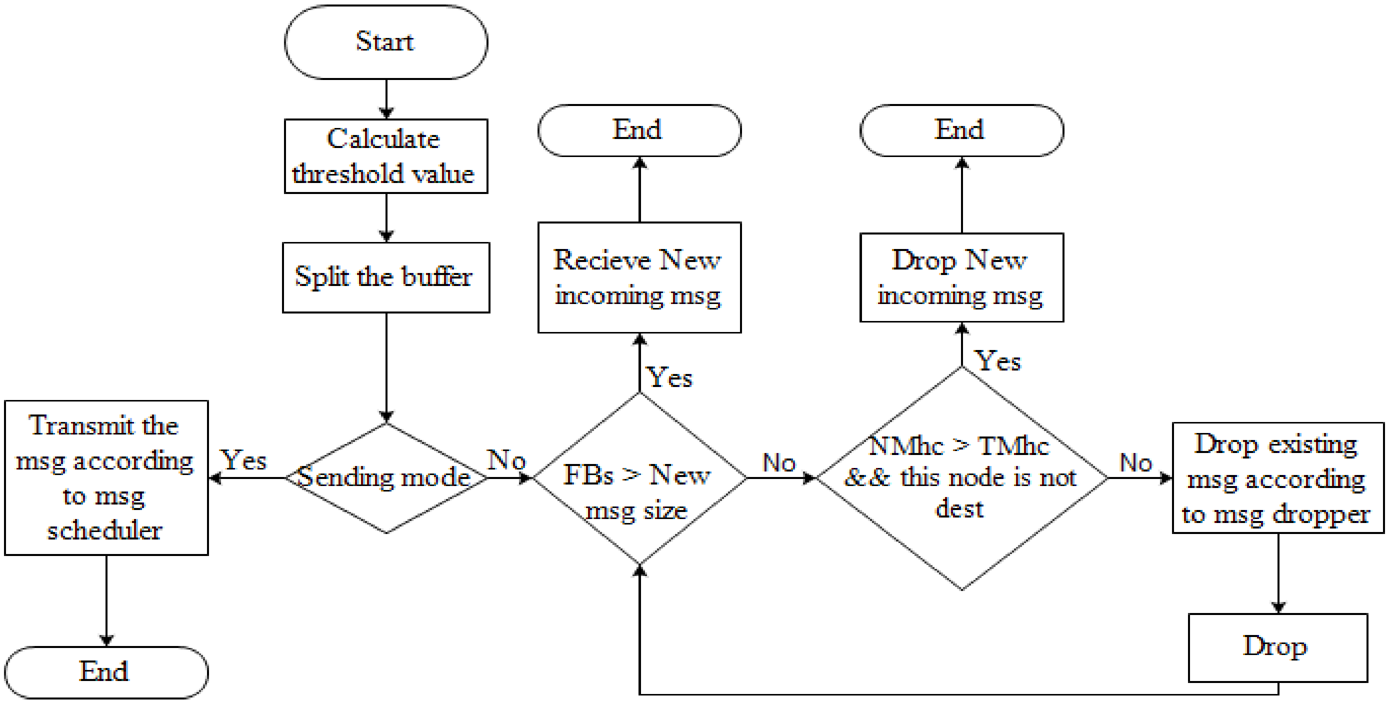

Figure 3.1: Flow chart of the proposed buffer management policy

Chapter 4

IMPLEMENTATION

4.1 Experimental Environment Setup

To demonstrate the studies, the ONE (Opportunistic Network Environment) simulator implemented in Java and available as open source has been used. It provides a powerful tool for generating mobility traces, running DTN messaging simulation with different routing protocols and visualizing both mobility and message passing in real time in its graphical user interface [36]. Many of the routing algorithms applicable to DTN environment are pre-implemented in the simulator.In this study, we did analysis on the performance of Epidemic routing protocol by applying varies buffer management policies.

To employ heterogeneous DTN environments we employ two map based movement models.The shortest path map-based movement model (the sophisticated version of the map based movement model) and the map route movement model. Both are provided with the ONE simulator and across all the simulation experiments the Well-Known Text (WKT) file of the default map of ONE simulator which represents a part of the city of Helsinki, Finland was used. Accordingly, the movement models gets its configuration data using files arranged 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.

In the simulation, we employee opportunistic network as dynamic set of mobile nodes. In our scenario, we have three types of groups which are consisted of pedestrians, cars, and trams. Pedestrians and cars nodes uses shortest map based movement model and trams use map route movement model that follows a pre-defined routes and follow tram lines. And, pedestrian and cars nodes have four types of point of interests like parks, shop, museums and hotels according to the interest of people as they visit many places regularly and occasionally. Also, it's assumed that all nodes in the network used a data rate of 2 Mbits/s in the simulation since DTNs includes various link technologies like Bluetooth and the standard data rate for Bluetooth is also similar with the data rate of the nodes configured for the nodes.

4.2 The Simulation Scenario

In the analysis phase we have used different scenarios to simulate the proposed buffer management policy using ONE simulator modeling on node movement; inter-node contacts, routing and message handling. The result of the simulation was analyzed based on the metrics described in section 5.1. In this study, the following points were considered for the simulation scenario.

Since varying the number of nodes and buffer size have an effect on the performance of DTN routing protocols with buffer management policies, in this scenario we varied number of total nodes between 55 to 155 and sizes of buffer between 2 to 10M for pedestrians and cars and fixed 10M in trams for validity of our result. Generally, the parameters like storage capacity, transmit range, bit-rates, movement and routing models and simulation duration are included in the defined nodes and their capabilities. It is assumed that each node have limited buffer space. So that they are prone to buffer overflows at a certain point of time due to the flooding of the messages when the nodes are encountered each other. The detail of various simulation parameters for the simulation scenario is shown in table 4.1

Table 4.1: Simulation environment parameters

Simulation Parameters	Values of Simulation Parameters
Simulation time	43200s=12h
Number of groups	3
Mobility model	Shortest map based movement,Map route movement
Routing protocol	Epidemic
Number of Pedestrian and car nodes	55-155
Number of tram	5
Time to live	300minutes(5h)
Buffer of pedestrian and car	2-10M
Buffer of tram	10M
car node speed	2.7-13.9 m/s
pedestrian node speed	0.5-1.5 m/s
Node wait time	5,15 minutes
Message size	500kB, 1MB
Message generation interval	25,35 seconds

4.3 Message Generation

The ONE simulator delivers two ways to generate application messages inside the simulation. These are message generators and external event files. The first way of message generation (i.e. message generators) is built-in in 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. Message event generator set in such a way that it generates the messages in every 25 to 35 seconds and every time message size can also be varied from 500 KB to 1MB.

4.4 The Mobility Model

This section provides the movement models (the way nodes move in simulation) of the node opted in ONE simulator for node mobility. As DTN are contingent greatly on 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 pedestrian, cars and trams carrying DTN enabled devices. In this study, the mobile devices with pedestrians, cars and trams are considered to buffer and exchange message with other DTN enabled devices from one point of interest to another node in the network.

Currently, ONE supports the basic Random Way point mobility model, arbitrary mobility models by using externally generated movement data, and different map-based movement models. All map-based movement models get their input data using files formatted with a subset of the Well Known Text (WKT) format. With map-based movement models, the nodes move using walkways and roads from the map data. In addition, different node groups can be set to use only certain parts of the map, thus allowing to differentiate between cars and pedestrians, so that, the former do not drive on pedestrian paths or inside buildings[37].

In this work, as stated in the above section two types of map based movement models i.e the shortest path map-based movement model and map route model which are pre-implemented in the ONE simulator are chosen for simulating the node mobility because, as stated on [37] the map based movement models are the only option to create scalable and flexible simulation. The shortest path map-based movement model is a derivative of the Random Way point model, where at decision point's node chooses a random destination and then follows the map-based shortest path to that destination. In our simulation it was used for pedestrians and cars group because nodes use Dijkstra's shortest path algorithm to calculate shortest paths rather than random from the current location to a randomly selected destination on map data and also, map data can contain Points of Interest (POIs).

POIs are places on the map area and, for each node group; separate probabilities can be defined for selecting POI from a certain group for nodes next destination. These POIs can be used to model e.g. tourist attractions, shops, restaurants etc. The map route model is route-based movement model which uses the same map data with other map based movement models but, instead of choosing destination map nodes in a random manner, nodes always choose the next destination on the route they are currently traveling. In this simulation it was used for trams group since it shows better performance on bus and tram route [37].

4.5 Reporting and Visualization

This section describes the way how the result of the simulation was visualized. These are through an interactive Graphical User Interface (GUI) and with generated images from the information gathered during the simulation. Figure 4.2 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 transfers. While the GUI is good for getting an intuitive over-all 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.

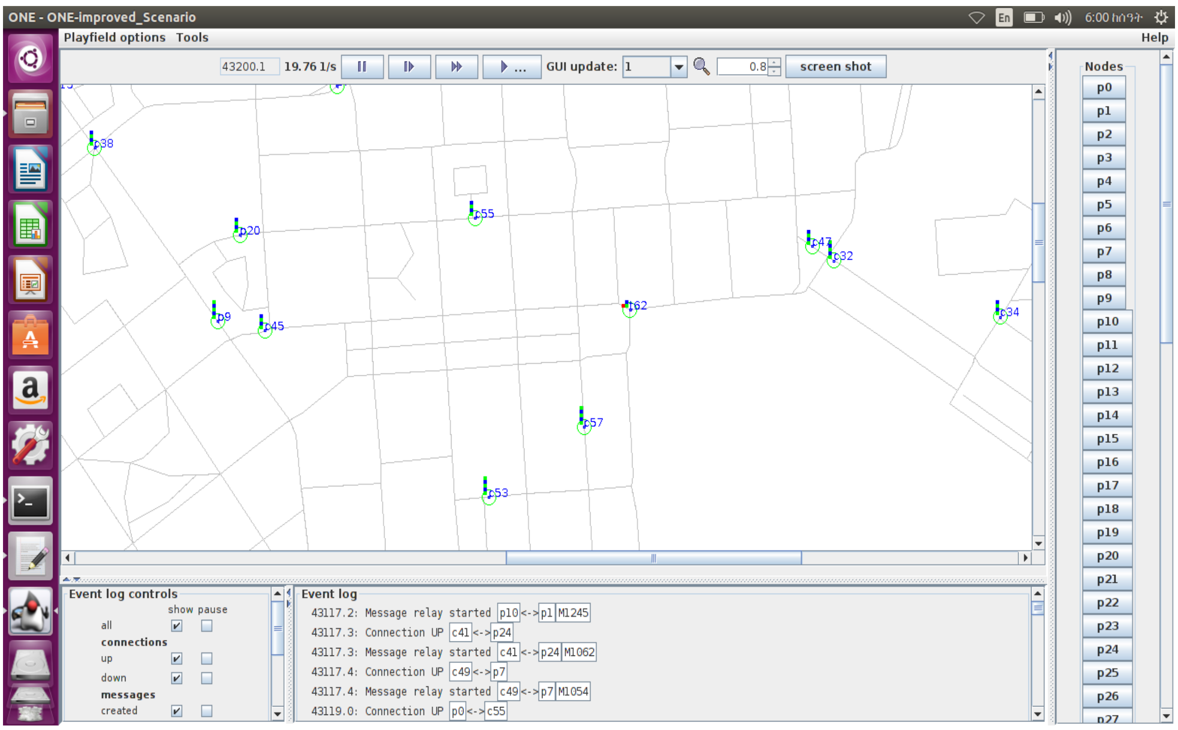

Figure 4.1: Screenshot of the ONE simulator GUI

Beside 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 message or connectivity events from the simulation engine and generate results based on them [37]. During the simulation run, the simulator generates the message statistics file in the report module that gathers statistics of overall performance (i.e. the number of created messages, message delivery ratio, message overhead ratio etc.). Accordingly, the report module's output was used to plot graph for the various message statistics report generated by the ONE simulator as clearly illustrated and discussed in chapter five.

Chapter 5

RESULT AND DISCUSSION

In this chapter the results, analysis and discussion of the scenarios targeted based on the simulation result is described. Additionally, the performance metrics for the designed simulation scenario was briefly described and simulation run with various configuration files based on the Predefined parameters were analyzed and discussed.

5.1 Performance Metrics

This section of the chapter describes the metrics adopted for evaluating the considered routing algorithms. Routing in the communication networks depend on the definition of performance indicator called routing metrics. As stated in [38], the popular routing metrics in traditional networks are number of hops, end-to-end delay, delay jitter, route life time 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[39,40,41]. The buffer management policy considered in this thesis was analyzed based on two routing metrics namely delivery probability and overhead ratio. Detail description of each metrics is explained as follows:

1. **Delivery Probability:** It is the fraction of generated messages that are correctly delivered to the final destination within given time period. It is computed as equation 5.1.

   Equation 5.1: Delivery probability

   $$Dp = \frac{\sum Md}{\sum Mc} \tag{5.1}$$

   Notation: Dp: is the message delivery probability

   Md:is the total number of message delivered

   Mc: is the total number of message created

2. **Overhead ratio:** is the percentage of resources consumed to process the transmission and storage of the message. In other word it's a metric used to estimate the extra number of messages needed by the routing protocol for actual delivery of the messages. It's computed as equation 5.2.

Equation 5.2 : Overhead ratio

$$OVR = \frac{\sum Rm - \sum Dm}{\sum Dm} \tag{5.2}$$

Notation:OVR:is Overhead ratio

Rm: is the total number of relayed messages

Dm: is the total number of delivered messages

5.2 Performance Evaluation

In this study, the network performance is studied from the application performance point of view. To evaluate performance of proposed buffer management policy we use Epidemic routing. It should be noticed that the proposed policy does not have any special assumption about routing protocol. Since epidemic routing uses lots of resources, the limitation of resources will dramatically decrease its performance. In this case, a good buffer management policy would increase routing performance. Because of this reason Epidemic routing is usually used for researching on buffer management policies and will be used in our study.

We did analysis on the performance of Epidemic routing by applying various existing buffer management policies like Drop Oldest(DO)[23], Drop Largest(DL)[23], Drop maximum hop-count(DMaxhop)[33], and Efficient buffer management policy(EBMP)[34] and compare them with the proposed buffer management policy. In addition to buffer management policies, 55-155 number of nodes and 2-10M buffer size of nodes have been used for evaluation. Based on buffer size and number of nodes, each simulation run generates the configuration file containing all the data packets that are sent between the nodes during the simulated scenario. Accordingly, the performance of the considered routing protocols was determined through analysis of the generated configuration file of the simulation. In the following section we are going to present the analysis of simulation result.

5.3 Result Analysis and Discussion

This section presents a detailed analysis of the simulation result using chart based on the generated configuration file of the simulation by comparing the performance of Epidemic routing under different buffer management policies in terms of the metrics discussed in section 5.1. The dropping policies used in the simulation i.e DO, DL, DMaxHop uses random scheduling policy which randomly transmits the message when there is a contact between two nodes. To measure each metrics we use two scenarios for each and individual of them.

In the first scenario, by fixing the number of nodes to 155 and changing the buffer size from 2M to 10M, the delivery probability and overhead ratio of all policies has been measured. And in the second scenario, the number of network nodes will be increased from 55 to 155, while buffer size of nodes is fixed and equal to 4M. Generally the results of changing in the buffer size and number of nodes on the metrics discussed above have been illustrated on the following figures.

5.3.1 Delivery Probability

Figure 5.1 illustrates the changes in message delivery ratio of Epidemic routing(ER) with DO, DL, DMaxHop, EBMP and proposed Message Drop control with priority buffer management policy(PMDCP) when the buffer size is between 2M and 10M. From the figure it can be noticed that when buffer size increases, the delivery probability of ER with all buffer management policies increase. This is because as the buffer size increases, buffer of the nodes becomes available to store and carry more messages, thereby enhancing the performance.

The DL policy shows less performance than the others in all cases due to its incapability to prioritize the messages which is stored in nodes buffer to be dropped when it is necessary in a selective manner and only considering the size of the message which have a direct impact on the delivery probability of the messages. In reverse the delivery probability of proposed policy shows higher performance than the others. This is because of priority mechanism for the messages stored in the buffer to be dropped when the available buffer space is less than the size of incoming messages and transmitted when the new nodes encounter. Therefore, from the graph we can conclude that the buffer space has substantial impact on delivery probability

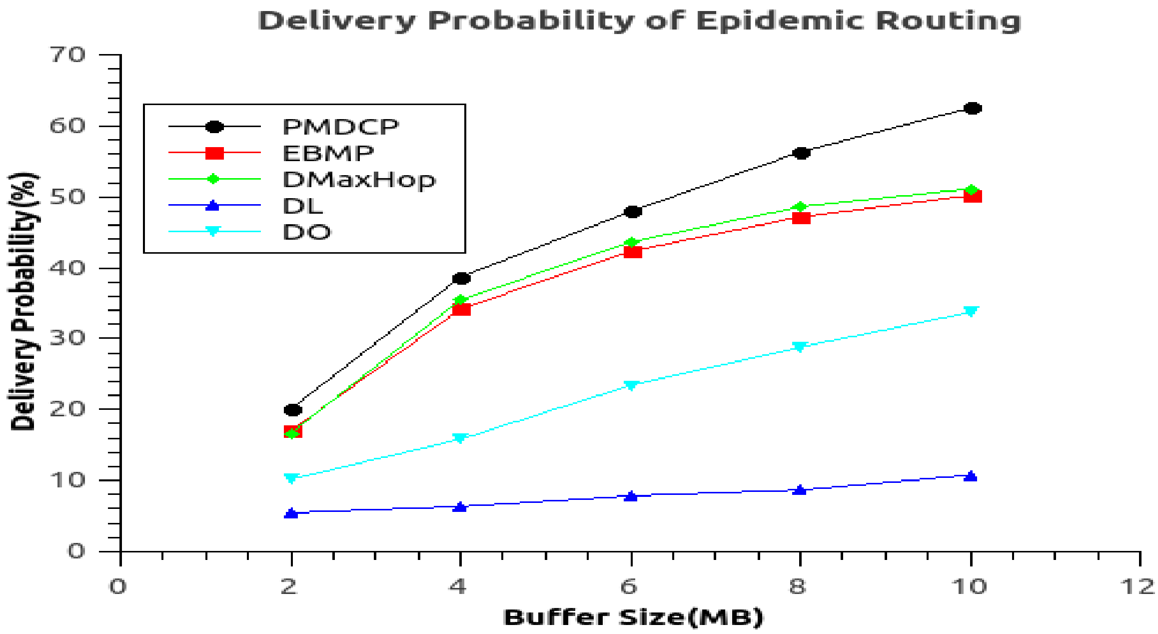

Figure 5.1: Percentage of delivery probability in terms of buffer size

Figure 5.2 shows the changes in message delivery ratio of Epidemic routing(ER) with DO, DL, DMaxHop, EBMP and proposed Message Drop Control with Priority buffer management policy (PMDCP) when the number of node is increased from 55 to 155. As understood from the figure the delivery probability of all policies increased when the number of node is between 55 and 85. But, when the number of node is above 100, the delivery probability of DO and DL policies are a little bit decreased and it is increased for the other policies.

The performance of DMaxHop policy is almost equal to EBMP when the number of node is between 55 and 100. But when the number of node is above 100 DMaxHop has better performance. Among existing policies, DMaxHop and EBMP are more efficient than others since they use some priority mechanism while DL has the worst performance. As displayed, PMDCP has a better performance compared to other policies in all cases.

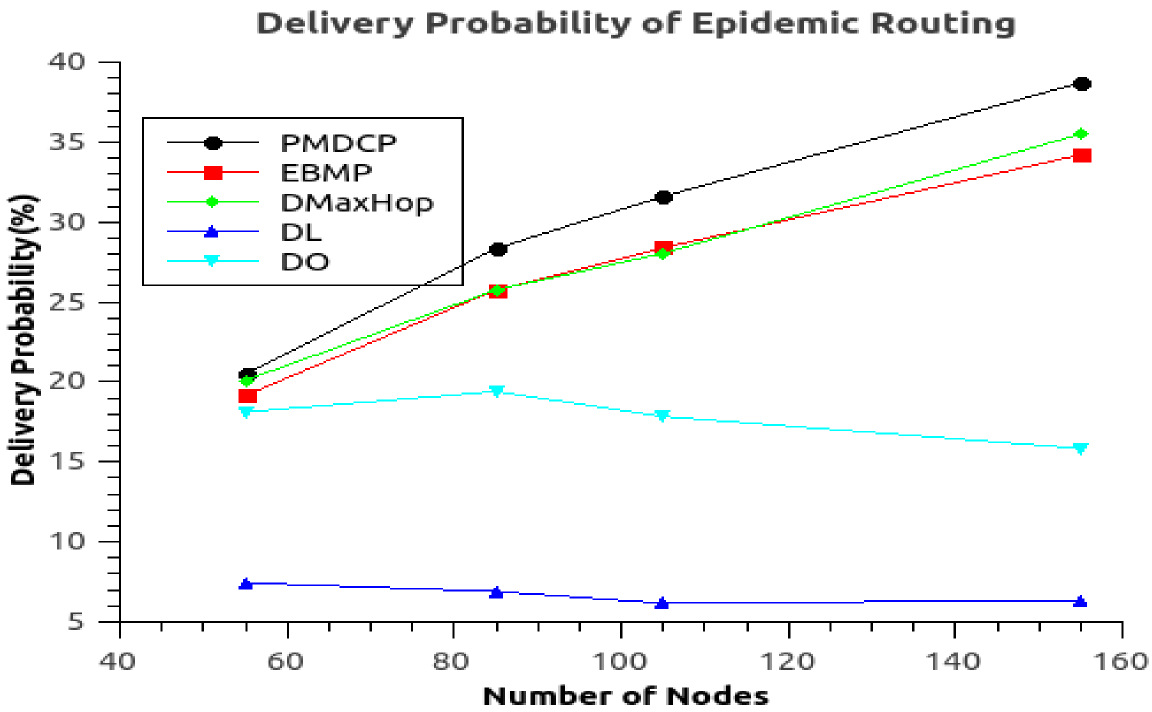

Figure 5.2: Percentage of delivery probability in terms of number of nodes

5.3.2 Overhead Ratio

Figure 5.3 shows the overhead ratio obtained for different buffer sizes for existing and proposed buffer management policies. As shown on figure 5.3 the PMDCP cause less overhead ratio value as compared with the other existing policies especially when the nodes buffer is highly constrained. This is because, In order to improve message delivery probability and reduce overhead, the proposed buffer management policy use the priority mechanism for the messages stored in the nodes buffer to be dropped or transmit. In addition to this, it computes the exact requirement and prepares room to accommodate the incoming message by dropping the message which has high hop count and equal size with the exact requirement. This approach avoids unnecessary message drop so that it reduce more overhead for the nodes.

In reverse DO shows the worst performance in highly congested networks because discarding oldest message may not help in accommodating newer and redundantly discarding a message to accept the new message cause a high overhead. Overall, overhead ratio decreases with the increased buffer size. This is because when the buffer size increase, buffer of the node have sufficient space to accommodate the new incoming message without dropping existing stored messages.

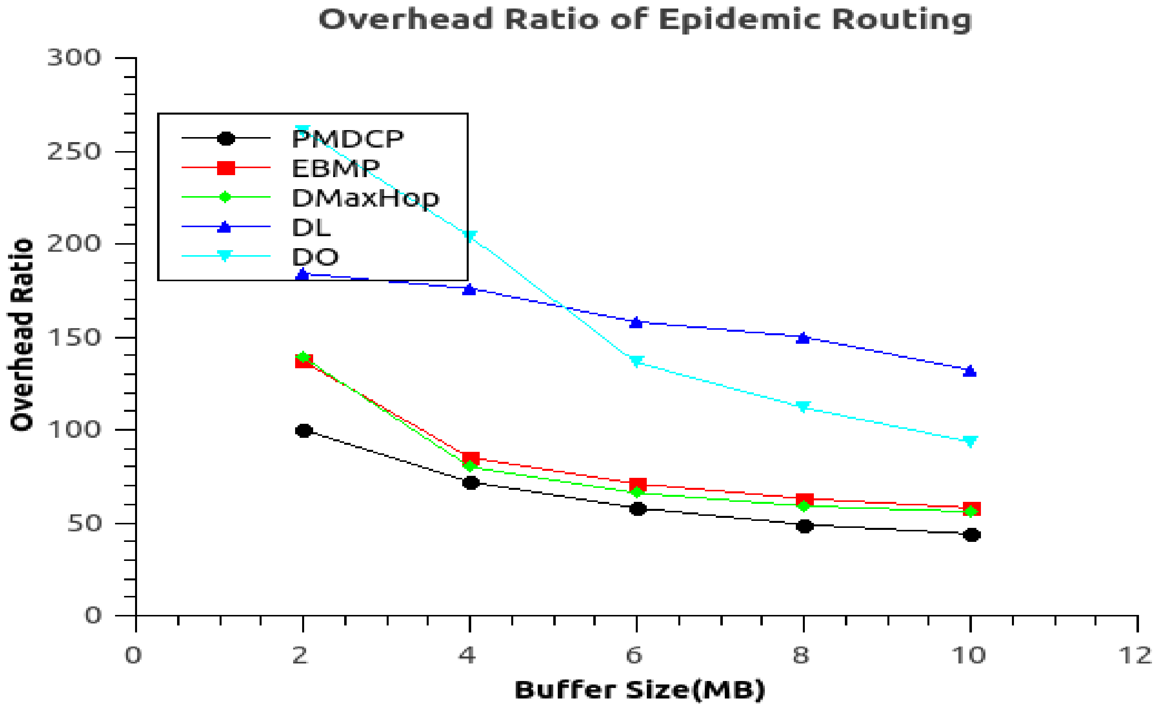

Figure 5.3: overhead ratio in terms of buffer size

Figure 5.4 shows the effect of varying number of node on the overhead ratio of ER with existing and proposed buffer management policies. From the figure it can be seen that the ER with DL shows less performance when the number of node is between 55 and 120. But, when number of node is above 120 DO show less performance than DL. PMDCP cause less overhead value because it minimizes number and volume of drops in addition to prioritization as discussed earlier. DMaxHop and EBMP show comparable performance because they mainly consider the same information for dropping. Generally from the figure we can conclude that as the number of node increase, the overhead ratio increases because as the number of node increase the amount of message generated in the network which increases the resource needed.

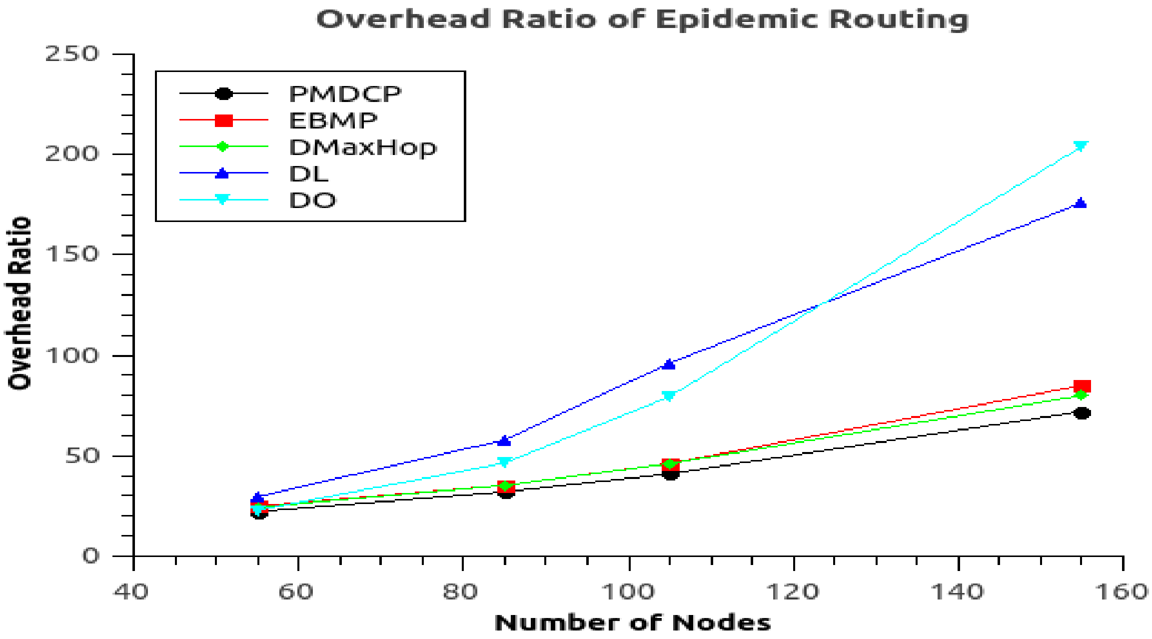

Figure 5.4: Overhead ratio in terms of number of nodes

5.4 Summary

The flooding based Epidemic routing has been chosen to evaluate the performance of various existing and proposed buffer management policies with respect to number of nodes and the nodes buffer size. This is because epidemic routing uses lots of resources, the limitation of resources will dramatically decrease its performance and usually used on buffer management researches. From the above sections we can conclude that combining different information's to effectively manage the buffer of nodes is used to improve the performance of routing protocols. And the performance of proposed policy outperforms the other in terms of delivery ratio and overhead. The following tables summarize the obtained results conducted through simulation studies with respect to the number of nodes and buffer size of nodes which is presented in the above section.

Table 5.1: Percentage of delivery probability in terms of buffer size

Buffer Management policies	Buffer size				
	2M	4M	6M	8M	10M
DO	10.17	15.84	23.41	28.81	33.72
DL	5.46	6.35	7.85	8.67	10.78
DMaxHop	16.66	35.5	43.62	48.6	51.13
EBMP	17	34.2	42.32	47.17	50.17
PMDCP	20.07	38.7	47.92	56.31	62.53

Table 5.2: Percentage of delivery probability in terms of number of nodes

Buffer Management policies	Number of nodes			
	55	85	105	155
DO	18.13	19.4	17.85	15.84
DL	7.43	6.9	6.16	6.35
DMaxHop	20.1	25.75	28.04	35.5
EBMP	19.15	25.75	28.39	34.2
PMDCP	20.5	28.35	31.6	38.7

Table 5.3: Overhead ratio in terms of buffer size

Buffer Management policies	Buffer size				
	2M	4M	6M	8M	10M
DO	261	204	136.1	111.9	93.4
DL	184	176	158	154	132
DMaxHop	139	80	66	59	56
EBMP	137	85	71	63	58
PMDCP	100	72	58	49	44

Table 5.4: Overhead ratio in terms of number of nodes

Buffer Management policies	Number of nodes			
	55	85	105	155
DO	22.86	46.4	79.5	204
DL	29.57	57.9	96.12	176
DMaxHop	24.37	35.25	46.1	80
EBMP	25	35.25	46.2	85
PMDCP	22.19	32.02	41	72

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

As already described in the above chapters, the DTNs attempts to provide a means for message delivery in challenged settings in case there is no end to end path between source and destination by following store-carry and forward approach. So, each node store the message for long period and most of DTN routing protocols are flooding based (replicate a message to other nodes without any knowledge) to achieve a high delivery ratio. Because of this buffer of nodes filled up quickly. In this case buffer management plays a great role on the performance of DTN routing mostly for flooding based. In this study, we proposed Message Drop Control with Priority buffer management policy for DTN routing protocols. It contains an intelligent decision for both message transfer and message drop for buffer management by combining three local information's (hop count, Time to live and size). Unlike most of the existing policies, the proposed policy combines different local information's and use adaptive threshold method to carefully prioritizing the messages. In addition to this, it decides whether to accept or discard the incoming message based on it's hop count value when buffer overflow occur and if it's must to accept the new incoming message, it tries to avoid unnecessary message drops and improve buffer utilization by exactly determining exact buffer requirement and discarding existing messages based on the determined value.

Finally, we have implemented our proposed policy on one of the flooding based routing category called Epidemic routing by using ONE simulator. We have evaluated our proposed policy by comparing the performance ER with existing policies and the proposed one regarding delivery ratio and overhead ratio. From the conducted simulation as result shows, the ER with the PMDCP shows better performance than the others in terms of packet delivery ratio and overhead.

6.2 Contribution

This study mainly focuses on buffer management policies which is the key issue in DTN. In this study first, we review the existing state of art in buffer management policies in DTN and after study bring the concept of combining different local information's and controlling unnecessary message drop in parallel with prioritization by using those information's. Finally, Message Drop Control with Priority buffer management policy is implemented to enhance the performance of routing protocols in DTN in terms of minimized overhead ratio and improved average delivery probability.

6.3 Future Work

As can be seen in results and discussion of this thesis, the proposed routing policy shows higher performance than the existing schemes in terms of delivery probability and overhead ratio.From this we can understand that effectively combining different attributes improve the performance of routing protocols in DTN. But in addition to attributes considered in this thesis there are another additional attributes that have an impact on the performance of routing protocols in DTN. Therefore, the future research work will come up with any new mechanism that combines other attributes in addition to attributes considered in this thesis in order to use accurate information in dropping or forwarding policies.

Bibliography

[1] B. D. Sumyla, "Mobile Ad-hoc Networks," 2006.

[2] S. Shukla, "Comparison of wireless network over wired network," vol. 5, pp. 14–20, 2017.

[3] Hussaini Adamu, "Comparative analysis between wired and wireless technologies in communications: a review," no. March, pp. 45–48, 2017.

[4] M. R. A. Kale, "An overview of MANET AD HOC network," vol. 6, no. 2, pp. 223–227, 2013.

[5] A. E. Al Fagih and H. S. Hassanein, "Routing Schemes for Delay-Tolerant Networks:An Applications Perspective Technical Report 2012-588," no. December 2010, 2012.

[6] S. Jain and M. Chawla, "Survey of buffer management policies for delay tolerant networks," J. Eng., no. March, pp. 1–7, 2014.

[7] S. Shukla, A. Munjal, and Y. N. Singh, "Routing Protocol Approaches in Delay Tolerant Networks," 2000.

[8] A. Lindgren, "MobiHoc Poster: Probabilistic Routing in Intermittently Connected Networks," vol. 7, no. 3, pp. 19–20.

[9] C. S. Raghavendra, "Spray and Wait: An Efficient Routing Scheme for," pp. 252–259.

[10] A. Vahdat and D. Becker, "Epidemic Routing for Partially-Connected Ad Hoc Networks," 2000.

[11] A. Ker, "Opportunistic Network Environment simulator," 2008.

[12] W. Sun, C. Liu, and D. Wang, "On Delay-Tolerant Networking and Its Application," vol. 51, no. Iccsit 2011, pp. 238–244, 2012.

[13] P. Puri, "A Survey Paper on Routing in Delay-tolerant Networks," pp. 215–220, 2013.

[14] A. P. Silva, S. Burleigh, C. M. Hirata, and K. Obraczka, "Ad Hoc Network: A survey on congestion control for delay and disruption tolerant networks," 2014.

[15] A. Upadhyay, "Routing Issues and Performance Of Different Opportunistic Routing Protocols In Delay Tolerant Network," vol. 7, no. 5, pp. 2147–2151, 2016.

[16] S. Almelu, A. J. Deen, and S. Silakari, "Delay Tolerant Network Routing Protocol: A Comprehensive Survey With Hybrid Technique," pp. 481–487, 2015.

[17] S. Ali, J. Qadir, and A. Baig, "Routing Protocols in Delay Tolerant Networks:A Survey," pp. 70–75, 2010.

[18] I. Engineering, "Comparative Study of Routing Protocols in Delay Tolerant Networks," pp. 7242–7246, 2014.

[19] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-Copy Routing in Intermittently Connected Mobile Networks," vol. 0, no. c, 2004.

[20] P. Pham and S. Perreau, "Multi-Path Routing Protocol with Load Balancing Policy in Mobile Ad Hoc Network," pp. 48–52.

[21] V. Cerf, S. Burleigh, and A. Hookeetal. "Delay-tolerant networking architecture." Internet: https://tools.ietf.org/html/rfc4838, April 2007 [Sep. 2, 2015].

[22]D. Pan, W. Cao, H. Zhang, and M. Lin, "Buffer Management and Hybrid Probability Choice Routing for Packet Delivery in Opportunistic Networks," vol. 2012, 2012.

[23] K. K. Ahmed, M. H. Omar, and S. Hassan, "Routing Strategies and Buffer Management in Delay Tolerant Networks," vol. 8, no. 10, pp. 139–143, 2003.

[24] K. S. S. Kim, "Enhanced buffer management policy that utilizes message properties for delay-tolerant networks," no. May 2010, pp. 753–759, 2011.

[25] A. Lindgren and K. S. Phanse, "Evaluation of Queuing Policies and Forwarding Strategies for Routing in Intermittently Connected Networks," 2006.

[26]G. Fathima, R. S. D. Wahidabanu, A. College, H. Tamilnadu, G. College, and S. Tamilnadu, "Integrating Buffer Management with Epidemic Routing in Delay Tolerant Networks," vol. 7, no. 7, pp. 1038–1045, 2011.

[27]A. Balasubramanian, B. N. Levine, and A. Venkataramani, "DTN Routing as a Resource Allocation Problem Categories and Subject Descriptors," 2007.

[28] A. Krifa and C. Barakat, "An Optimal Joint Scheduling and Drop Policy for Delay Tolerant Networks," 2008.

[29] Y. Li, M. Qian, D. Jin, L. Su, and L. Zeng, "Adaptive Optimal Buffer Management Policies for Realistic DTN," pp. 0–4, 2009.

[30] T. Le, H. Kalantarian, and M. Gerla, "A Buffer Management Strategy Based on Power-Law Distributed Contacts in Delay Tolerant Networks," 2016.

[31] S. Rashid, Q. Ayub, M. S. M. Zahid, and A. H. Abdullah, "E-DROP: An Effective Drop Buffer Management Policy for DTN Routing Protocols," vol. 13, no. 7, pp. 8–13, 2011.

[32] M. Moetesum, F. Hadi, M. Imran, and A. Ali, "An adaptive and efficient buffer management scheme for resource-constrained delay tolerant networks," 2015.

[33] Y. Harrati and A. Abdali, "MaxHopCount: A New Drop Policy to Optimize Messages Delivery Rate in Delay Tolerant Networks," pp. 37–41, 2016.

[34] C. C. Sobin, "An Efficient Buffer Management Policy for DTN," Procedia - ProcediaComput. Sci., vol. 93, no. September, pp. 309–314, 2016.

[35] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine, "MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks," vol. 00, no. c, 2006.

[36] S. Ramesh and P. Ganesh Kumar, "Opportunistic Network Environment simulator," 2008.

[37] A. Keränen and J. Ott, "Increasing reality for DTN protocol simulations," p. 9, 2007.

[38]M.E. Steenstrup, "Routing in Communications networks," Hertfordshire, UK: Prentice Hall International (UK) Ltd., 1995.

[39] M. C. Anjula Mehto, "Comparing Delay Tolerant Network Routing Protocols for Optimizing L-Copies in Spray and Wait Routing for Minimum Delay," April 2013.

[40] N. Mehta and M. Shah, "Performance Evaluation of Efficient Routing Protocols in Delay Tolerant Network under Different Human Mobility Models," Int. J. Grid Distrib. Comput., vol. 8, no. 1, pp. 169–178, 2015.

[41]E. P. C. Jones, "Practical Routing in Delay-Tolerant Networks," 2006.

Appendix

Simulation Scenario Configuration File

```
## Scenario settings
Scenario.name = Scenario_%%Group.router%%
Scenario.simulateConnections = true
Scenario.updateInterval = 0.1
# 43200s == 12h
Scenario.endTime = 43200

# "Bluetooth" interface for all nodes
btInterface.type = SimpleBroadcastInterface
# Transmit speed of 2 Mbps = 250kBps
btInterface.transmitSpeed = 250k
btInterface.transmitRange = 10

# High speed, long range, interface for group 3
highspeedInterface.type = SimpleBroadcastInterface
highspeedInterface.transmitSpeed = 10M
highspeedInterface.transmitRange = 10

# Define 3 different node groups
Scenario.nrofHostGroups = 3

# Common settings for all groups
Group.movementModel = ShortestPathMapBasedMovement
Group.router = EpidemicRouterPMDCP
Group.bufferSize = 2M
Group.waitTime = 0, 20

# All nodes have the bluetooth interface
Group.nrofInterfaces = 1
Group.interface1 = btInterface
# Walking speeds
Group.speed = 0.5, 1.5
# Message TTL of 300 minutes (5 hours)
Group.msgTtl = 300
Group.nrofHosts = 75
```

```
# group1 (pedestrians) specific settings
Group1.groupID = p

# group2 (cars) specific settings
Group2.groupID = c
# cars can drive only on roads
Group2.okMaps = 1
Group2.speed = 2.7, 13.9

# groups(Trams) specific settings
Group3.groupID = t
Group3.bufferSize = 10M
Group3.movementModel = MapRouteMovement
Group3.routeFile = data/tram3.wkt
Group3.routeType = 1
Group3.waitTime = 10, 30
Group3.speed = 7, 10
Group3.nrofHosts = 5
Group3.nrofInterfaces = 2
Group3.interface1 = btInterface
Group3.interface2 = highspeedInterface

# Message creation parameters
# How many event generators
Events.nrof = 1
# Class of the first event generator
Events1.class = MessageEventGenerator

#settings  specific for the MessageEventGenerator class
Events1.interval = 25,35
# Message sizes (500kB - 1MB)
Events1.size = 500k,1M
# range of message source/destination addresses
Events1.hosts = 0,154
# Message ID prefix
Events1.prefix = M
```

```
## Movement model settings
# seed for movement models' pseudo random number generator
MovementModel.rngSeed = 1
# World's size for Movement Models without implicit size
MovementModel.worldSize = 4500, 3400
# How long time to move hosts in the world before real simulation
MovementModel.warmup = 1000

## Map based movement -movement model specific settings
MapBasedMovement.nrofMapFiles = 4
MapBasedMovement.mapFile1 = data/roads.wkt
MapBasedMovement.mapFile2 = data/main_roads.wkt
MapBasedMovement.mapFile3 = data/pedestrian_paths.wkt
MapBasedMovement.mapFile4 = data/shops.wkt

# how many reports to load
Report.nrofReports = 1
# length of the warm up period (simulated seconds)
Report.warmup = 0
# default directory of reports
Report.reportDir = reports/testPMDCP
# Report classes to load
Report.report1 = MessageStatsReport
```