Design and Performance Analysis of Opportunistic Routing Protocols for Delay Tolerant Networks

by

Tamer Abdel-kader

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Delay Tolerant Networks (DTNs) are characterized by the lack of continuous end-to-end connections because of node mobility, constrained power sources, and limited data storage space of some or all of its nodes. Applications of DTNs include vehicular networks and sensor networks in suburban and rural areas. The intermittent connection in DTNs creates a new and challenging environment that has not been tackled before in wireless and wired networks. Traditional routing protocols fail to deliver data packets because they assume the existence of continuous end-to-end connections. To overcome the frequent disconnections, a DTN node is required to store data packets for long periods of time until it becomes in the communication range of other nodes. In addition, to increase the delivery probability, a DTN node spreads multiple copies of the same packet on the network so that one of the copies reaches the destination. Given the limited storage and energy resources of DTN nodes, there is a trade off between maximizing delivery and minimizing storage and energy consumption.

DTN routing protocols can be classified as either blind routing, in which no information is provided to select the next node in the path, or guided routing, in which some network information is used to guide data packets to their destinations. In addition they differ in the amount of overhead they impose on the network and its nodes. The objective of DTN routing protocols is to deliver as many packets as possible. Acquiring network information helps in maximizing packet delivery probability and minimizing the network overhead resulting from replicating many packet copies. Network information could be node contact times and durations, node buffer capacities, packet lifetimes, and many others. The more information acquired, the higher performance could be achieved. However, the cost of acquiring the network information in terms of delay and storage could be high to the degree that render the protocol impractical. In designing a DTN routing protocol, the trade-off between the benefits of acquiring information and its costs should be considered.

In this thesis, we study the routing problem in DTN with limited resources. Our objective is to design and implement routing protocols that effectively handles the intermittent connection in DTNs to achieve high packet delivery ratios with lower delivery cost. Delivery cost is represented in terms of number of transmissions per delivered packet. Decreasing the delivery cost means less network overhead and less energy consumption per node. In order to achieve that objective, we first target the optimal results that could be achieved in an ideal scenario. We formulate a mathematical model for optimal routing, assuming the presence of a global observer that can collect information about all the nodes in the network. The optimal results provide us with bounds on the performance metrics, and

show the room for improvement that should be worked on. However, optimal routing with a global observer is just a theoretical model, and cannot be implemented practically.

In DTNs, there is a need for a distributed routing protocol which utilizes local and easily-collectable data. Therefore, We investigate the different types of heuristic (non-optimal) distributed routing protocols, showing their strengths and weaknesses. Out of the large collection of protocols, we select four protocols that represent different routing classes and are well-known and highly referred by others working in the same area. We implement the protocols using a DTN simulator, and compare their performance under different network and node conditions. We study the impact of changing the node buffer capacities, packet lifetimes, number of nodes, and traffic load on their performance metrics, which are the delivery ratio, delivery cost, and packet average delay. Based on these comparisons, we draw conclusions and guidelines to design an efficient DTN routing protocol.

Given the protocol design guidelines, we develop our first DTN routing protocol, *Eco-Friendly Routing for DTN (*EFR-DTN*)*, which combines the strengths of two of the previously proposed protocols to provide better delivery ratio with low network overhead (less power consumption). The protocol utilizes node encounters to estimate the route to destination, while minimizing the number of packet copies throughout the network.

All current DTN routing protocols strive to estimate the route from source to destination, which requires collecting information about node encounters. In addition to the overhead it imposes on the network to collect this information, the time to collect this information could render the data worthless to propagate through the network. Our next proposal is a routing protocol, *Social Groups Based Routing (*SGBR*)*, which uses social relations among network nodes to exclude the nodes that are not expected to significantly increase the probability of delivering the packet to its destination. Using social relations among nodes, detected from node encounters, every group of nodes can form a social group. Nodes belonging to the same social group are expected to meet each other frequently, and meet nodes from other groups less frequently. Spreading packet copies inside the same social group is found to be of low-added value to the carrying node in delivering a packet to its destination. Therefore, our proposed routing protocol spreads the packet copies to other social groups, which decreases the number of copies throughout the network. We compare the new protocol with the optimal results and the existing well-known routing protocols using real-life simulations. Results show that the proposed protocol achieves higher delivery ratio and less average delay compared to other protocols with significant reduction in network overhead.

Finally, we discuss the willingness of DTN nodes to cooperate in routing services. From a network perspective, all nodes are required to participate in delivering packets

of each other. From a node perspective, minimizing resource consumption is a critical requirement. We investigate the degree of fair cooperation where all nodes are satisfied with their participation in the network routing services. A new credit-based system is implemented to keep track of and reward node participation in packet routing. Results show that the proposed system improves the fairness among nodes and increases their satisfaction.

Acknowledgements

My utmost thanks to ALLAH for all his gifts and blessings in all my life.

I would like to express my sincere appreciation and thanks to my supervisor Prof. Kshirasagar Naik for his guidance and for helping me with his comments on every step of this work. I have learned from him the highly positive and self motivated attitude. His friendly meetings with me were very helpful in removing the stress of work, and having a better view of the problems and how to solve it. This thesis would have not been done without his guidance, advice, and valuable comments.

Thanks should go to all my committee members for their fruitful discussions and comments on this work.

Special thanks should go to my friends who helped and supported me in different ways throughout my work.

I am grateful to my sisters, friends and colleagues for their encouragement, help and prayer.

Great thanks to my wife, Walaa, who supported me and endured with me this hard part of the life. My wife, my daughter, Maryam, and my son, Ziad, without you, I wouldn't be able to continue this work.

Thanks are not enough to be given to my parents. Rather, there are no words that are valuable to equalize their love and support for me. They are behind any success in my life.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

DC	Delivery Cost
Del	Delay
DR	Delivery Ratio
NU	Network Utility
UF	Unfairness Index
ACK	Acknowledgment
AODV	Ad hoc On-demand Distance Vector
AP	Access Point
CDT	Credits and Debits Table
DSR	Dynamic Source Routing
DTN	Delay Tolerant Network
EBR	Encounter-based routing
FCFS	First Come First Serve
FIFO	First In First Out
GPS	Global Positioning System
ID	Identity
IPN	Interplanetary Network

MAC Medium Access Control

MANET Mobile Ad-Hoc Networks

MBR Model-Based Routing

OLSR Optimized Link State Routing Protocol

ONE Opportunistic Network Environment

PROPHET Probabilistic Routing Protocol using History of Encounters and Transitivity

SnW SPRAY and WAIT

VAN Village Area Networking

Chapter 1

Introduction

In today's world, there is a large number and a wide variety of information-storage devices: Cell phones, laptops, desktop computers, super-power servers, GPS devices, different types of sensors, satellites and many other. Sharing information between these devises is crucial to efficiently store, process and utilize the information. Therefore, communication and networking protocols were developed to facilitate the efficient sharing of information. These protocols were designed to serve the main applications that needed networking during the previous decades and that were cost effective when implemented, such as providing Internet services or cell phone connection in urban areas and crowded networks. Because of that, these protocols were based on several assumptions such as the presence of immediate or short-delayed end-to-end connections between sources and destinations. In addition, they assumed a reliable connection where data are highly expected to be transferred from one node to another after one or few retransmissions. After the spread of mobile and battery-powered devices and the need to cover sparsely-populated regions, such as rural and suburban areas, the assumptions of the traditional protocols were violated, and there becomes a need to develop new protocols that can handle the new challenging environment.

1.1 Delay Tolerant Networks (DTNs): Overview

In networks where nodes are sparsely distributed and mobile, power sources and data storage spaces are limited, and one-hop connections are unreliable, node communications become highly challenged. Such a challenging environment is referred to in literature as a Delay or Disruption Tolerant Network (DTN) [77]. The term *Delay Tolerant Network* has been coined by Kevin Fall in March 2003 as a terrestrial application of what was called at that time *Interplanetary Network* (IPN). At that time, the first draft of the RFC 4838 [26] was published. Following that, many publications have studied, analyzed the DTN architecture [27] and proposed new protocols to handle the new environment.

In DTNs, nodes do not have end-to-end connections for long periods of time. Therefore, in order to provide reliable communications in the intermittent connection environment, intermediate (relay) nodes are required to store data packets for long periods of time. This generates a challenging situation because of the limited storage space, the limited battery power and the uncertainty of whether the stored data is still of value to the destination or not.

We can summarize the main characteristic of DTN as follows:

- Long delay .
- Frequent disconnection.
- Opportunistic connection.
- Large packet sizes (called bundle in the DTN architecture [66]).
- Small buffer space dedicated for delay-tolerant data.
- Battery-powered devices: Many of the DTN nodes depend on mobile power sources, such as battery.

1.2 Applications of DTNs

Delay tolerant networking can be applied in situations where data is not delay sensitive and the main goal is to deliver as much of the generated data as possible. Applications vary between the commercial and non-commercial including scientific and environmental applications. In the following, we present some of the implemented DTN projects and possible DTN applications.

1.2.1 Providing Residential Internet Access

Imagine the case of a suburban or semi-rural area where it is required to connect the people to the Internet for only delay-insensitive applications such as emailing. The cost of extending Internet cables or building a complete wireless Internet infrastructure will be very high for such a purpose. Delay tolerant networking can provide a solution by collecting the data from this area into one or several spots on the roads coming out of that area so that vehicles can transfer the data to the nearest Internet gateway which may be in a neighbor city (within tens of kilometers). The same procedure can be done for the incoming data. Data can be collected wirelessly using access points mounted on the vehicles, or they can be collected on any digital media such as CDs and then carried by vehicles. This idea has been commercialized by First Mile Solutions with a system called DakNet [61]. Figure 1 shows how delay tolerant networking can help in extending Internet services in suburban areas. Another project by Wizzy Digital Courier [64] implements a one-hop delay tolerant network to provide Internet to rural South African schools.

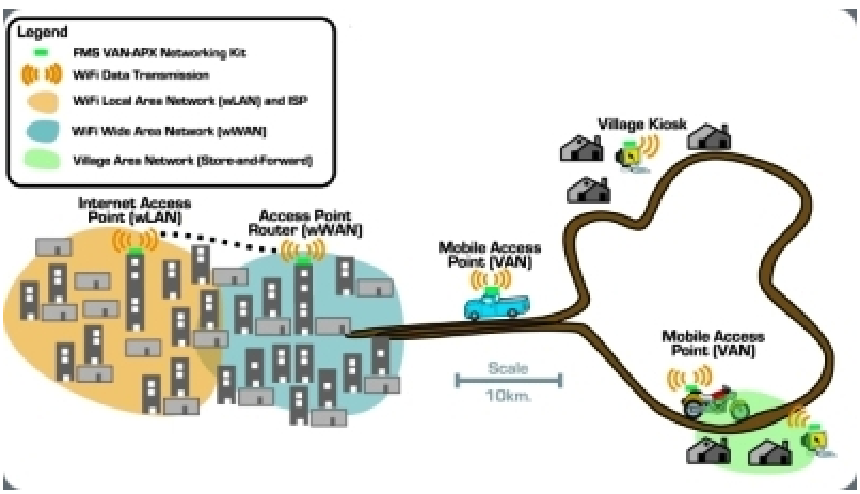


Figure 1.1: Village Area Networking (VAN) by First Mile Solutions

Another application is when there is a limited bandwidth connection which is shared by both delay-sensitive and delay-tolerant applications. The delay tolerant data can be delayed to periods where the bandwidth is not fully utilized by the delay-sensitive data. Delay-tolerant data can then be sent on opportunistic basis. In addition, they can be used as a high bandwidth alternative for the low bandwidth Internet connections [79] in developing regions by using DVDs as communication media.

1.2.2 Vehicular Networking

Vehicular networking is a wide and growing field of DTNs, where many applications are being explored. One of these applications is the virtual warning signs [54] which brings the hidden or unseen warning signs to the vehicle driver to be able to take the required precautions as early as possible. Another application is to provide Internet access to vehicles, by connecting to roadside wireless base stations [59].

1.2.3 Sensor Networks and Scientific Applications

There are a plenty of non commercial DTN applications. These include monitoring and tracking wildlife animals [42] and whales in oceans [69], and environmental monitoring such as lake water quality monitoring and road-side noise monitoring [58]. Collecting data from sparsely distributed sensors [67] is one of the common applications of DTNs. DTNs can be applied in a variety of other fields ranging from healthcare to education to economic efficiency [10]. Moreover, they can be used in space networking such as the interplanetary network [13] which was historically the first application of a DTN.

1.3 Routing Challenges in DTNs

Many routing protocols have been developed for the Mobile Ad-Hoc Networks (MANET) to handle its dynamic behavior, such as DSR [40], AODV [62], and OLSR [19] and many others. However, these protocols assumed the existence of continuous end-to-end connections between sources and destinations. Therefore, if applied to the intermittent environment, packets will be dropped once the carrying node fail to find their destinations within a short time.

Routing protocols developed for DTNs are adapted to this challenging environment by sending multiple copies of each data packet to increase the probability that one of the copies reaches the destination. Nodes receiving the packet copies store and carry them until they meet other nodes or meet their destinations. In simple DTN routing protocols, nodes blindly send data packets to other nodes in their communication range, without having node selection criteria. These blind-routing protocols range from the full network flooding to the limited flooding. This approach might achieve high delivery ratio of data packets provided that there are sufficient storage and energy resources to handle all the propagating packets. On the other hand, it has its drawbacks, such as burdening the buffers and the inefficient use of the contact duration. Other routing protocols tend to restrict forwarding of data packets to selected nodes. Using information collected from the network, nodes estimate and predict geographical or social relations among themselves. The collected information is used to guide the packets to their destinations. This approach fails when the network topology is changing faster than the rate of information gathering. If provided full knowledge about the network, such as nodes mobility, buffer, and energy, then optimal routes could be found and only a single packet copy is required to propagate to its destination. However, in DTN networks, collecting full knowledge is impractical. Practical routing protocols are those with little or no knowledge about the network. Finding optimal routes could be done in theoretical studies to use its performance results as a comparison benchmark.

From the analysis of current DTN routing protocols, it is found that there are several trade-offs to be considered in a protocol design:

- A trade off between maximizing packet delivery ratio and minimizing the delivery cost. Maximizing delivery ratio requires increasing the number of packet copies spread throughout the network to increase the probability of reaching the destination, while minimizing delivery cost, in terms of network overhead, requires decreasing the number of copies.
- Another trade off is the compromise regarding the amount of information collected to guide the packets to their destinations. Collecting information from the network helps in selecting the relaying nodes to the destination, but requires time to collect the information which increases the packet delays. On the other hand, collecting little or no information leads to spreading the packet copies blindly, and decreases the probability of reaching the destination unless a large number of copies were spread.

An efficient DTN routing protocol should spread a small number of packet copies to reduce network overhead, while guiding the packet copies using only local information to reach the destination.

1.4 Motivations & Objectives

Routing in DTNs is a key component in providing and maintaining high performance networking. The main performance metrics are delivery ratio, network overhead, and the average delay. Although many routing protocols have been proposed to provide high-performance routing, they were captivated to reducing the delay at the expense of the other metrics.

1.4.1 Exploiting the Social-Grouping Characteristic of DTNs

Imagine a network of student smartphones and laptops. Students are distributed among several buildings in school, while some are in their homes or in streets. Students in the same building are in the communication range of each other most of the time, through single or multiple hops. Therefore, a building network is behaving as a traditional Ad-Hoc network. Networks in the different buildings are connected to each other by students moving in and out the buildings going from one class to another. Students in the streets are connected to those in campus by intermittent connections via other transceiver devices around them. The whole network can be viewed as a DTN which has many inter-groups connected to each other intermittently. The rate of connection among the nodes inside the same group is much higher than to those outside the group. In addition, the possibility of a node in a group to connect to another node outside the group is around the same possibility of another node in the same group to connect to the same node outside the group. i.e., the behavior of the group nodes is expected to be the same.

Another example of such grouping would be the taxi companies in a big city. Taxis that belong to the same company contact each other more frequently, because they either meet in the city roads or in the company garage during their break times. Taxis that belong to different companies only meet each other by coincidence in the city roads. Therefore, taxis that belong to the same company are considered to have strong social connection among each other and weak social connections with the taxis of other companies.

Our objective is to develop a routing protocol that spreads a small number of packet copies to reduce network overhead, while guiding the packet copies using only local information to reach the destination. To achieve that goal, we exploit the social grouping characteristic of DTN nodes. We consider two nodes to belong to the same social group if they contact each other frequently compared to their contacts with other nodes.

1.4.2 Supporting Fair Cooperation in DTNs

When designing a DTN routing protocol, a question arises: What are the node incentives to receive and forward packets of other nodes, while it is energy constrained ? This question could be answered using some of the game theory techniques, such as reputation systems [65, 38, 46, 53] and credit-based systems [17, 85]. Simply, if a node does not cooperate in serving other nodes, it will not be served. However, these trust management systems do not answer other questions that are unique to the DTN environment, such as:

- What guarantees that a node, receiving packets of other nodes, will store, replicate and select the best next-hop node, and not just forward to a random node?
- How does a node reward other nodes for their cooperation, especially that there will be multiple copies of the same packet?
- How much should a node cooperate? In other words, what is the fair point of cooperation at which all the nodes are satisfied ?

These questions are unique to the DTN environment because of the hop-by-hop routing used, the multiplicity of packet copies spread throughout the network, and the fully decentralized nature of DTN. To answer these questions, we propose a new utility function that is calculated by each node to capture the degree of cooperation of the node and the network. Using this utility function, nodes can determine the amount of cooperation they can offer without being accused of selfishness.

1.5 Summary of Contributions

Our contributions in this research are summarized as follows:

- We formulate an optimization problem of single-copy centralized routing in DTNs, assuming the availability of present and future node contacts and buffer information. We implement the problem with three different objectives: minimum delay, minimum number of hops, and maximum number of delivered messages. We solve the problem using parameters from simulated networks. Output results are used as a performance benchmark to compare with the heuristic (non-optimal) protocols.
- We study the distributed non-optimal routing protocols developed for DTNs, and conduct a performance comparison among selected well-known protocols (Epidemic, SPRAY-AND-WAIT (SnW), PROPHET and MAXPROP) representing the different types of routing protocols (blind and guided, limited and full flooding). Based on these comparisons, we draw conclusions and guidelines to design an efficient DTN routing protocol.
- Given the protocol design guidelines, we develop our first DTN routing protocol, Eco-Friendly Routing for DTN (EFR-DTN), which combines the strengths of two of the previously proposed protocols to provide better delivery ratio with lower network overhead (less power consumption). The protocol utilizes node encounters to estimate

the route to destination, while minimizing the number of packet copies throughout the network.

- We propose our second heuristic distributed protocol, *Social Groups Based Routing*(SGBR), which uses social relations among network nodes to exclude nodes that are not expected to significantly increase the probability of delivering the packet to its destination. Simulation results show that SGBR protocol achieves lower network overhead while achieving the same or better delivery ratio compared to the other routing protocols.

- Finally, we discuss the incentives of DTN nodes to cooperate in the routing process. We investigate the degree of fair cooperation at which the network nodes are satisfied. A new credit-based system is implemented to keep track of and reward node participation in packet routing.

1.6 Thesis Organization

The thesis is organized as follows:

- In Chapter 2, we review the routing protocols developed for DTNs.

- In Chapter 3, we present our system model, including the network model, the performance metrics and the methods used to verify and validate the system used.

- In Chapter 4, we introduce a novel optimal routing formulation. We simulate a DTN network and conduct performance comparison among the different objective function.

- In Chapter 5, we conduct performance comparison among selected well-known DTN routing protocols, and analyze the results.

- The first proposed distributed routing protocol EFR-DTN is presented in chapter 6 with an extensive set of experiments to compare with the other protocols and the optimal results.

- The second proposed distributed routing protocol SGBR is presented in chapter 7 with an extensive set of experiments to compare with the other protocols and the optimal results.

- A game theoretic approach to support fair cooperation in DTN is proposed in chapter 8 with an extensive set of experiments to show its usefulness .
- Finally, we state our conclusions and future work in chapter 9.

Chapter 2

Background and Literature Review

The routing decision depends on the available information about the network, such as the time and the frequency with which nodes meet each other and the duration of these meetings, the storage capacity of each node and the number of nodes in the network. This information may be collected in networks with infrastructure using fixed or moving data collectors, such as rovers and basestations, or in networks without infrastructure by exchanging data between meeting nodes. The more the amount and the accuracy of the information, the better the routing decision. However, in practical scenarios, the information collected is often less than required and not accurate enough to take an optimal decision. Therefore, routing protocols implement heuristics to estimate a good decision with the available information. In addition, the protocols depend on spreading several copies of the same data packet through the network to increase the chance of reaching the destination. Therefore, routing protocols can be classified according to the amount and type of information used to take the routing decision [82], the method of replication of data packets [41], or the type of infrastructure used to collect data [60]. We combined the three classifications in one diagram as shown in figure 2.1.

The amount of information is used in the first dimension, in which the routing protocols are categorized into three levels of knowledge: No-Knowledge, Partial-Knowledge and Full-Knowledge. In the No-Knowledge level, a routing protocol does not have a node selection criteria, i.e., it choose the next node on a route blindly. Therefore, it is required to spread multiple copies of the same data packet to increase the probability of reaching the destination. Hence, the replication method is used to further classify protocols within this level of knowledge. Protocols range from those with tight limits on the number of hops and number of packet copies to those with no limits (full flooding). The second level of knowledge is the Partial-Knowledge, where most of the routing protocols belongs to. The

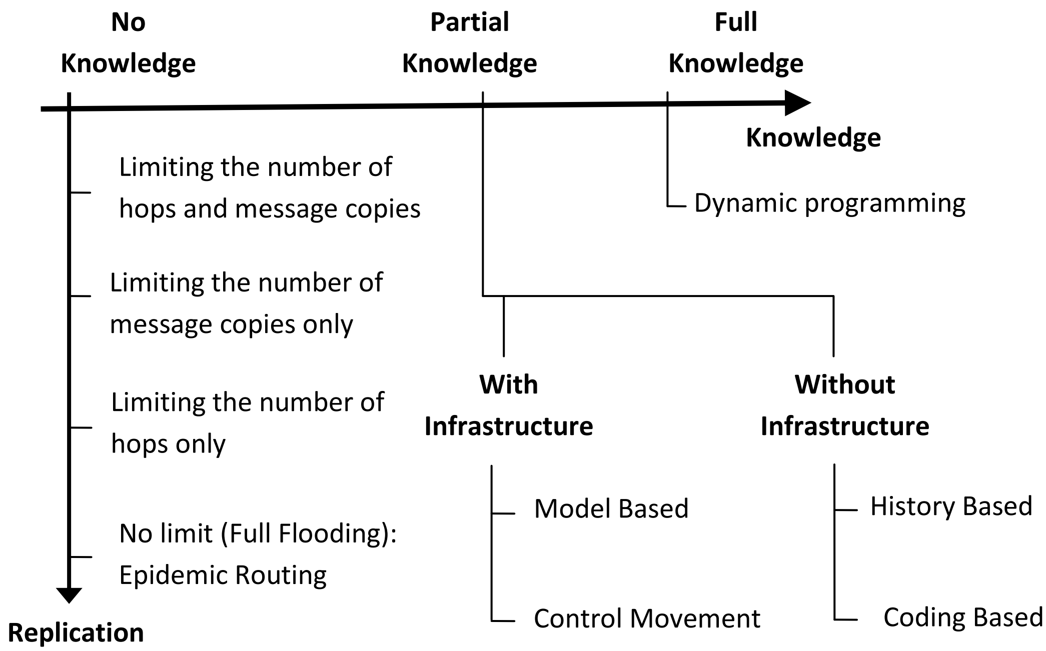


Figure 2.1: Classification of DTN Routing Protocols

information types can be historic or present information about node contacts, locations or mobility patterns. This information may be collected using additional infrastructure, such as rovers or basestations, or without infrastructure by exchanging information during node contacts. The third level of knowledge is the full knowledge, which is a theoretical case used as a reference to compare with. If we have full knowledge, then we can find optimal solutions to the routing problem.

In the following subsections, we present a brief overview on some of the routing protocols proposed under each knowledge level.

2.1 Routing Protocols with No-Knowledge about the Network (Blind Routing)

These are the simplest protocols in terms of communication overhead and processing power consumed to take a routing decision. There is a minimal control-data exchange between contacting nodes, no infrastructure used and little processing required for the routing decision. Routing protocols in this category range from the full flooding to the limited flooding. Limitation can be by number of hops or number of packet copies or both.

The simplest of these protocols is the full flooding technique, or so called Epidemic Routing [78]. In Epidemic routing, each node broadcasts its data packets to whatever nodes it meets over the time. Broadcasting of a particular data packet stops when the packet expires or is deleted by the node either because of buffer fullness or reception of destination delivery acknowledgment. This protocol proves to provide the highest delivery ratio and the lowest end-to-end delay, if the storage space did not overflow. However, since buffers are limited, the protocol performance drops significantly with the large traffic. More explanation of the Epidemic protocol is in Section 5.1.

To reduce the buffer overhead, limited flooding protocols were proposed. Limitation can be applied on the number of hops [28, 30, 56, 78] or the number of packet copies [70, 72, 74], or both [70]. In [28], number of hops is limited to only one, which means the source keeps the data packet until it meets the destination. This reduces the network overhead to its minimum because there is only one copy of any data packet in the network. However, the probability that a source node of a packet meet its destination node is expected to be low, especially in a sparse network. This renders the delivery ratio to be low. A variation of [28] is to allow for two hops from the source [30], which increases the delivery ratio. However, it is still prone to the low probability to find destination nodes after few number of hops. Increasing the number of hops [56] improves the delivery ratio, but increases the network overhead.

In [70], it is suggested to limit the number of packet copies for each node, in addition to the limitation on the number of hops. However, the best performance of all of these protocols were proved if the total number of packet copies in the network is limited [70, 72, 74]. The number of packet copies or number of hops to limit depend on the network parameters, such as the number of nodes and their contact rate.

2.2 Routing Protocols with Partial Knowledge about the Network (Guided Routing)

Guided routing protocols collect information about other nodes in the network to guide packets to their destinations. These protocols assign weights to nodes using information collected from the network. This information could be topological [50, 5, 12, 6, 24], environmental and energy-aware [43], or content based [21, 32]. The following are some of the information that can be collected and used as routing metrics:

- Contact times: The times at which two nodes meet each other. Future contact times may be estimated in cases such as the city bus schedule.
- Contact rate: The frequency of contacts between each pair of nodes.
- Contact duration: For how long the nodes will stay in the transmission range of each other.
- Buffer occupancy: How many bytes are available in the each node's buffer.
- Packet copies: How many copies of each data packet are there in the network.
- Location: The location of the nodes at different times can help predict their future locations.
- Mobility pattern: If there is a known mobility pattern for the nodes or they are moving according to predefined paths, then we can target their location at different times in the future.
- energy: How much energy is available at each node could help avoid unreliable paths.
- sleep/awake schedule: For a communication to occur between two nodes, they have to be awake and in the transmission range of each other. Therefore, knowing the sleep/awake schedule of each node affects the decision taken about next-hop nodes.

Topological information address the nodes contact information such as expected contact times, location and mobility patterns of nodes. This information can be collected using special infrastructure [69] or by exchanging data between mobile nodes [50, 12, 6]. If infrastructure is installed, it can be fixed as in the Infostation model [29]. In this model, nodes communicates only with basestations, and there is no node-to-node communication. Another version of the Infostation model is the SWIM [69], where nodes can communicate with each other in addition to communicating with the basestations. A mobile infrastructure helps in gathering data from fixed objects as in the Data-MULE system [67]. In the Data-MULE system, mobile agents called MULEs move around a sensor network to collect data gathered and sampled by the sensors and deliver it to a set of fixed access points (APsAP) distributed around the network. To achieve better and fast delivery of data, additional mobile nodes can be exploited to offer relaying services. This approach is implemented in [83], and the additional mobile nodes are called *Message Ferries*. Although adding hardware infrastructure to the network improves the performance, it adds to the cost of the network. This cost may be tolerated in some industrial applications, while in other applications it may be render the network impractical. Guided-Routing protocols outperform Blind-Routing protocols in the delivery ratio, but unless they have an efficient packet selection mechanism, they increase the average packet delay.

2.2.1 History-based (Social) Routing

Exchanging data between mobile nodes is the low-cost alternative for installing special infrastructure. In [14] and [50], nodes record their contacts with other nodes. This information is used to predict the probability to deliver data packets through each node. The protocol in [14] relies on the meeting between nodes and the visits of nodes to geographical locations. While the protocol in [50] relies only on the meeting, but it can predict the delivery probability by using the direct meeting between two nodes or the indirect meeting using intermediate nodes in between. When two nodes meet, they increase their link weight towards each other and towards the nodes met by the other node. More explanation of the PROPHET protocol is in Section 5.3.

Similar to PROPHET, MAXPROP [12] strengthens the link between two nodes using the number of meetings. The contribution in MAXPROP can be observed in its buffer management technique which encourages forwarding packets with lower number of hops over those traveled far in the network without reaching their destinations. More explanation of the MAXPROP protocol is in Section 5.4.

The Context-Aware Routing (CAR) protocol [55] is designed to support message delivery in both continuously and intermittently connected environments. If at the time a packet arrives, a path to its destination exists, and the packet can be forwarded using an existing routing protocol, the CAR protocol routes the packet using one of the known adhoc networking routing protocols, such as DSDV [63]. However, If at the time a packets arrives, a path to its destination cannot be found, the CAR protocol stores the packet waiting for an opportunity to be forwarded. Instead of replicating the message to all the neighbors, the message is sent to a host characterized by the highest probability of reaching the recipient. In other words, this host acts as a message carrier. This process is based on the evaluation and on the prediction of the context information using a time series analysis technique, Kalman Filter. Delivery probabilities are synthesized locally from context information such as the rate of change of connectivity of a host. The prediction process is used during temporary disconnections and it is carried out until it is possible to guarantee with certain accuracy.

Gathering information about user mobility to build up a mobility model is presented in [76]. The authors performed an experimental study to collect user mobility to form an opportunistic ad-hoc network in a campus environment. The approach is unique in that they do not have a predetermined model of user mobility, and they strive to provide a networking model based only on pair-wise contact. Using this trace data, they simulate a network using epidemic propagation. They observe that power management is one of the critical issues that should be considered in DTNs.

In [68], the authors presented a relay-based routing scheme for ad-hoc satellite networks where nodes are required to buffer data for a certain period of time until the node gets an opportunity to forward it. They proposed Interrogation-Based Relay Routing (IBRR), where the nodes interrogate each other to learn more about network topology and nodal capacity to make intelligent routing decisions. Optimistic forwarding [18] is used in IBRR. To select the next hop, a node needs to not only know the present and future connectivity relation with its current time, but also the same information of its current neighbors. This one-hop look-ahead is necessary for making routing decisions in the relay-based routing framework. Lookahead beyond one-hop can prove to be time consuming and counter productive.

The work in [25], adds the limited flooding approach to the guided routing by waiting for a better quality node. Encounter-based routing (EBREBR) [57] uses as a routing criteria the number of encounters for each node. The node with more encounters is selected as the best candidate. Several other protocols are examined for heterogeneous networks in [75].

The collected information can be used to detect social relations among the network nodes as in [50, 12, 34, 11, 22]. The work is [11] is one of the early studies about community-based routing. However, the study is restricted to two communities only, and the authors did not provide a routing protocol based on their study.

2.2.2 Model-based Routing

Users and vehicles carrying devices usually move following certain known patterns such as walking along a street or driving down the highway. Once users describe their motion pattern, the intermediate nodes have a more accurate estimation of which nodes move toward the destination with higher probability. Model-Based Routing (MBRMBR) [9] uses world models of the mobile nodes for a better selection of relaying nodes and the determination of a receiver location without flooding the network. World models contain location information (e.g. road maps or building charts) and user profiles indicating the motion pattern of users. The key idea of the approach is to take into account that mobile devices typically do not follow the random walk motion pattern but are carried by human beings. Once humans describe their motion pattern or some sort of monitoring deduces it, MBR can rely on this information in the form of user profiles to choose a relay that moves toward the target with higher probability. With the information of the receiver location, each intermediate node can determine the next relaying node based on the user profile. Each node offers an interface that emits the probability that the user will move toward a given location. Hence the routing algorithm can choose less relays if a small number of relays have been found that will move near or to the location with high probability. Their work relies on the known receiver location, which is provided by a central location service, an unrealistic assumption.

A model of nodes moving along on a highway is described in [18]. With ad hoc networks deployed on moving vehicles, network partitions due to limited radio range become inevitable when traffic density is low, such as at night, or when few vehicles carry a wireless device. A key question to ask is whether it is possible to deliver messages in spite of partitions, by taking advantage of the fact that predictable node movement creates opportunities to relay messages in a store-and-forward fashion. The authors in [18] test the hypothesis that the motion of vehicles on a highway can contribute to successful message delivery, provided that messages can be relayed and stored temporarily at moving nodes while waiting for opportunities to be forwarded further. Messages are propagated greedily each time step by hopping to the neighbor closest to the destination. Two kinds of transmission schemes are used, pessimistic forwarding and optimistic forwarding, which are distinguished by how long the messages are permitted to stay in intermediate nodes. In pessimistic forwarding, a message is dropped whenever no next hop exists for its destination. This is how forwarding works in most ad hoc network implementations. In optimistic forwarding, messages without next hops may remain on intermediate nodes for some time, hoping that physical movement of network nodes eventually creates a forwarding opportunity. Using vehicle movement traces from a traffic micro simulator, the authors measure average message delivery time and find that it is shorter than when the messages are not relayed.

2.2.3 Node Movement Control-based Routing

Instead of waiting for another node to pass by and connect, a node can move to other nodes to contact with them and exchange their packets. In [49], the host trajectories of nodes are controlled to facilitate communication in ad hoc networks.Each node knows the trajectories of all other nodes in the network. When it has a packet detained to another node, it computes the shortest path to the destination given the trajectories of all the nodes, then it moves to the next node in the shortest path.

Driven by the proved theorem that mobility increases the wireless network capacity [31], [14] suggests the addition of a limited number of autonomous agents to the network area and studies the problem of increasing the capacity of a DTN through autonomous agents that move in the network with the purpose of increasing network performance. The addition of these agents requires a control algorithm that can coordinate agent movements in order to optimize the performance of the network.

In [83, 84], special mobile nodes, called ferries, move around the deployment area and take responsibility of transferring data between nodes. Node movements are controlled to optimize some metric, such as the message delay.

2.2.4 Coding-based Routing

Coding techniques can be used to limit message flooding and to achieve higher delivery probability. Erasure coding techniques [37, 80] encodes a data packet into a large number of blocks. Having a certain number of these blocks reaching the destination, the original message can be decoded. Network coding techniques [81] combine some of the packet received and send them out as one packet, which decreases the transmissions over the network leading to better performance and less energy consumed at nodes.

2.3 Routing Protocols with Full Knowledge about the Network (Fully-Guided Routing)

In this category, the protocols assume they have the full knowledge about the network in advance. So, for example each node should know when it is going to meet the other nodes, and when the other nodes are going to meet each other, which is practically impossible. However, the results of applying these protocols can be used as a reference to compare with when implementing the practical protocols. Because they have the full knowledge, they do not need to send multiple copies of packets. Only a single copy is sent in the optimal route according to the objective of the routing protocol. In [36], several routing protocols are proposed including the one which uses the full knowledge about the network. They define four knowledge oracles that represent different information category. The *Contacts Summary Oracle* contains information about the aggregate statistics of the contacts. The *Contact Oracle* contains information about all the contacts that will occur. The *Queuing Oracle* contains information about the buffer capacities and remaining spaces at all nodes at any time. The *Traffic Demand Oracle* contains information about the present and future traffic load. If either the Contacts Summary Oracle or the Contacts Oracle is available alone, then a modified Dijkstra algorithm can be used. If all the Oracles are available, then a dynamic programming problem can be formulated to find the optimal route.

2.4 Summary & Conclusion

Routing protocols are classified according to the amount and type of information used to take the routing decision. Blind routing protocols aim at fast spreading of packets in the network. They do not use node selection criteria. They vary according to their spreading mechanism and amount. Epidemic was historically the first routing protocol that belongs to this class. In Epidemic, each node spreads copies of the packets it carries to whatever node it meets, until the packet lifetime expires or a destination acknowledgment is received. This protocol proves to provide the highest delivery ratio and the lowest end-to-end delay, if the node buffers do not overflow and the contact durations are long enough to transfer all the uncommon packets. However, since data storage space is limited and contact durations may not be that long, the protocol performance drops significantly with the high traffic rates. To overcome this problem, other routing protocols limit the flooding of packets to a certain number of copies or hops. Spray-and-Wait (SnW) protocol limits the number of copies by associating with each copy the number it is allowed to spread. Nodes spread copies of the packets they receive according to the associated number of copies. When the allowed number of transfers reaches one, the carrying node stops transferring the packet until it either meet the destination or the packet is dropped due to buffer overflow or lifetime expiry. A binary version of SnW permits each node to use half the number of transfers allowed for the packet and the other half is left for the receiving node.

Guided routing protocols use the available network information to guide packets to their destinations. These protocols assign weights to nodes using information such as expected contact times, location and mobility patterns of nodes, number of packet copies. This information can be collected using special infrastructure or by exchanging data between mobile nodes. The PROPHET protocol estimates a node metric by tracing the number of meetings between nodes. When two nodes meet, they increase their link weight towards each other and towards the nodes met by the other node. Guided routing protocols outperform blind protocols in the delivery ratio, but increases the average packet delay.

If provided full knowledge about the network, such as nodes mobility, buffer, and energy, then optimal routes could be found and only a single packet copy is required to propagate to its destination. However, in DTN networks, collecting full knowledge is impractical.

Practical routing protocols are those with little or no knowledge about the network. Finding optimal routes could be done in theoretical studies to use its performance results as a comparison benchmark.

Chapter 3

System Model

In this chapter, we present our network model, the performance metrics and the methods used to verify and validate the system used.

3.1 Network Model

We consider a city-wide network where nodes are pedestrians and vehicles roaming along predefined paths representing city roads. N is the set of nodes. These nodes are connected to each other via wireless links when they come in the communication range of each other. In such an event, they are said to be in contact.

A contact c has four attributes: Sender (S_c), Receiver (R_c), Time (t_c), and Duration (D_c), as explained in the following:

- Sender: The node whose buffer contains the messages that are to be transmitted to the other node.
- Receiver: The node which is targeted to receive the messages transmitted from the other node.
- Time: The time at which the two nodes appear in the communication range of each other and start exchanging the control packets.
- Duration: The time period during which the two nodes are in the communication range of each other and are able to transfer messages.

Each node $n \in N$ has its own buffer to store messages. A node's buffer has two attributes: Capacity and occupancy, as follows:

- Capacity B_n: The maximum number of messages the buffer of node n can carry.
- Occupancy b_n: The number of messages in the buffer of node n at the beginning of the contact.

Each message m has four attributes: Source, Destination, Transmission time, and TTL, as explained in the following:

- Source: The node that generated the message.
- Destination: The node to which the message should be delivered.
- Transmission time d_m: The time it takes to transmit m from one node to another during a contact.
- Lifetime L_m: The Time-To-Live TTL, or the lifetime of message m, after which the message has no value and should be dropped.

Throughout the thesis, we will use the two words 'path' and 'route' interchangeably, and the same for the words 'packet' and 'message'.

3.2 Performance Metrics

We consider three metrics to measure the performance of the different protocols, which are:

- Delivery ratio, DR:

$$DR = \frac{\sum_{n \in N} (P_{dv})_n}{\sum_{n \in N} (P_g)_n} \tag{3.1}$$

  where $(P_{dv})_n$ is the number of packets delivered to their destination node n, and $(P_g)_n$ is the number of packets generated at their source node n. The delivery ratio is, simply, the ratio of the packets delivered to those generated in the network during the simulation time.

- Delivery cost, DC:

$$DC = \frac{\sum\limits_{n \in N} (P_r)_n - \sum\limits_{n \in N} (P_{dv})_n}{\sum\limits_{n \in N} (P_{dv})_n} \tag{3.2}$$

where $(P_r)_n$ is the number of packets received by node n. DC represents the cost the routing protocol should pay, in terms of redundant packets, to deliver one packet.

- Average packet delay, Del:

$$Del = \frac{\sum\limits_{n \in N} \sum\limits_{p \in DV_n} d_p}{\sum\limits_{n \in N} (P_{dv})_n} \tag{3.3}$$

where d_p is the delay encountered by packet p delivered to its destination node n, and DV_n is the set of packets delivered to their destination n. The metric is simply the ratio of the sum of all delivered packets delays to the number of delivered packets.

3.3 Confidence of Performance Results

As stated in [35], when any timing result is measured, the average time

$$\bar{T} = (t_1 + t_2 + ... + t_p)/P$$

is obtained from P repetitive experiments. P is chosen large enough such that with a degree of confidence 0.95 we guarantee that the expected value of T is within

$$\bar{T} \pm 0.1T$$

That is:

$$P_r\{\bar{T} - 0.1T < T < \bar{T} + 0.1T\} > 0.95$$

To test how the mean of the timing results satisfies the specified confidence interval, let u be the estimated value of the mean $\bar{T}$, and

$$S^2 = \frac{1}{p-1} \sum_{i=1}^{p} (T_i - \bar{T})^2$$

be the sample variance. Assume that the random variable T is normally distributed, then we can obtain the $1 - \alpha$confidence interval for using the following formula, with t as the $t - distribution$:-

$$\bar{T} - t_{\alpha/2,p-1} \frac{S}{\sqrt{p}} < u < \bar{T} + t_{\alpha/2,p-1} \frac{S}{\sqrt{p}}$$

It is required that the interval between the maximum and minimum values to be less than 10% of the mean:-

$$2 * t_{\alpha/2,p-1}\frac{S}{\sqrt{p}} < 0.1 * \bar{T}$$

Taking $\alpha = 0.05$and sample size $P = 10$, then

$$2 * t_{0.025,9}\frac{S}{\sqrt{10}} = 2 * 2.262 * \frac{S}{\sqrt{10}}$$

That is

$$2 * 2.262 * \frac{S}{\sqrt{10}} < 0.1 * \bar{T}$$

, i.e.

$$(45.24)^2 S^2 < 10\bar{T}^2$$

The previous equation is used to test all timing results obtained in the experiments. Each experiment is repeated $P = 10$times and a set of $T_i$are obtained. $\bar{T}$and $S$are computed and the result is accepted if the above equation is satisfied.

3.4 System Verification and Validation

The goodness of a simulation model is measured by the closeness of the model output to that of the real systems. Since a number of assumptions about the behavior of real systems are made in developing the model, there are two steps in measuring the goodness. The first step is whether the assumptions are reasonable, and the second step is whether the model implements those assumptions correctly. These two steps are called validation and verification, respectively.

3.4.1 System Verification

Verification, also called debugging, is related to the correctness of the implementation. A number of techniques have been used for debugging as presented below:

- **Top-Down Modular Design**: The model is structured in modules that communicate with each other via well-defined interfaces. These modules are objects of classes and the functions within these objects. The interface consists of a number of input variables and data structures. Once the interface and the function of the module have been specified, it can be independently developed, debugged, and maintained. Modularity thus allows the verification of the simulation to be broken down into smaller problems of verifying the modules and their interfaces. Top-down design consists of developing a hierarchical structure for the model such that the problem is recursively divided into a set of smaller problems. First, the model is divided into a number of modules with different functions. Each of these modules is then further subdivided into modules. The process is repeated until the modules are small enough to be easily debugged and maintained.

- **Antibugging**: Antibugging consists of including additional checks and outputs in the program that will point out the bugs. For example, the model counts the number of packets sent by a number of source nodes as well as the number of packets received by the destination nodes. The number of packets lost on the route and the packets received should equal the number of packets sent. A nonzero difference would indicate a programming error.

- **Deterministic Models**: The key problem in debugging simulation models is the randomness of variables. It is obvious that a deterministic program is easier to debug than a program with random variables. A common verification technique, therefore, is to specify constant (deterministic) distributions, then we can easily determine the output variables and thus debug the modules.

- **Run Simplified Cases**: The model is run with simplified cases, for example, only one packet, or only one source. These cases can be easily analyzed and the simulation results are compared with the analysis.

- **Trace**: A trace consists of a time-ordered list of events and their associated variables. They are present at several levels of detail: events trace, procedure trace, and variables trace. The trace outputs are useful in debugging the model. Tracing causes additional processing overhead, and therefore, the model should have switches that allow the traces to be turned on and off. After we finished our work, there were about 30 trace files used in our simulator.

- **Graphic Displays**: Simulations take a long time to run. Graphic displays for traces help debug the simulation; they can present the same information as in the trace but in a more comprehensive form. It is difficult to look at a long trace, while it is easy to inspect the display for the same period. We used MSEXCEL and MATLAB as a tool for displaying traces graphically.

- **Continuity Test**: Continuity tests consist of running the simulation several times for slightly different values of input parameters. For any one parameter, a slight change in input should generally produce only a slight change in the output. Any sudden changes in the output should be investigated.

- **Degeneracy Tests**: Degeneracy tests consist of checking that the model works for extreme (lowest or highest allowed) values of system, configuration, or workload parameters. For example the network simulation model works for a system with no sources, or channels with errors for the whole simulation time.

- **Consistency Tests**: These tests consist of checking that the model produces similar results for input parameter values that have similar effects. For example, two sources with an arrival rate of 100 packets per second should load the network to approximately the same level as four sources with an arrival rate of 50 packets per second each.

- **Seed Independence**: The seeds used in random-number generation should not affect the final conclusion. Thus the model should produce similar results for different seed values. This is verified by running the simulation with different seed values.

3.4.2 System Validation

Validation refers to ensuring that the assumptions used in developing the model are reasonable in that, if correctly implemented, the model would produce results close to that observed in real systems [39, 52]. The validation techniques depend upon the assumptions and, hence, on the systems being modeled. Model validation consists of validating the three key aspects of the model:

1. Assumptions
2. Input parameter values and distributions
3. Output values and conclusions

Each of these three aspects may be subjected to a validity test by comparing it with that obtained from three possible sources: Expert intuition, Real system measurements, and theoretical results. The expert intuition is not available in our case. Theoretical results were obtained for the optimal case which acts as a performance benchmark. Comparison with real systems is the most reliable and preferred way to validate a simulation model. In our work, we use a well-known DTN simulator, the ONE simulator [45], to simulate the proposed protocols. Using the simulator, we create real scenarios, and apply each of the protocols to compare their performance in the different scenarios. In addition, we used traces of real systems [48, 16, 20, 15] to run the simulator and the results were almost the same as those provided in the references.

3.5 Model Based Simulations

In model-based experiments, we use a model for node contacts driven from real trace data [16]. According to [16], the inter-contact time, that is the interval between two successive contacts of the same pair of nodes is modeled using the power law distribution. The cumulative distribution function (CDF) of the power law distribution is computed as

$$\mathrm{P(X \geq x)} = \left(\frac{\mathrm{x}}{\mathrm{x_{min}}}\right)^{-\theta} \tag{3.4}$$

where X is the random variable, x is the inter-contact time, x_{min} is the minimum inter-contact time and θ is the parameter that characterizes the power law, $\theta > 0$. In our experiments, we use $\theta = 0.9$ as mentioned in [16]. To avoid partitioning, each node is made to contact with at least one node. The maximum number of nodes to contact with is drawn from a uniform distribution of up to one fifth of the total number of nodes in the network. Packets are created at every node using Poisson distribution, and assigned a random destination. When generated, each packet is associated with a unique identifier, time of creation and a time to live TTL.

3.6 Real Movement Simulations

We used the ONE simulator [44] to generate the movement scenario. The network consists of vehicles and pedestrians moving around a city. We set the mobility model as map-based movement, where vehicles and pedestrians are restricted to move in predefined paths and routes derived from real map data. We used the map data of the Helsinki downtown area (roads and pedestrian walkways) provided with the simulator. Figure 3.3 shows the Helsinki map with ten nodes (5 pedestrians and 5 vehicles) roaming in the city. Nodes choose a random point on the map and then follow the shortest route to that point from their current location.

In the ONE simulator, we also set the nodes types and capabilities. Nodes can be pedestrians, vehicles, or trams. Their capabilities include radio interface, persistent storage, movement, energy consumption and message routing. Nodes are grouped into groups, where each group is configured with different capabilities. Packets are generated randomly with a predefined lower and upper interarrival times, and assigned a random source and destination. Using the simulator script input, we can specify the values of the different parameters. An example of a ONE script is shown in Figures 3.1 and 3.2.

```
## Scenario settings
Scenario.name = default_scenario
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
# Define 2 different node groups
Scenario.nrofHostGroups = 2
# Common settings for all groups
Group.router = EpidemicRouter
Group.bufferSize = 5M
Group.waitTime = 0, 120
# 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
# group1 (pedestrians) specific settings
Group1.groupID = p
Group1.movementModel = ShortestPathMapBasedMovement
Group1.nrofHosts = 30
# group2 specific settings
Group2.groupID = v
# cars can drive only on roads
Group2.okMaps = 1
# 10-50 km/h
Group2.speed = 2.7, 13.9
Group2.movementModel = ShortestPathMapBasedMovement
Group2.nrofHosts = 20
```

Figure 3.1: A sample script for the ONE simulator (part 1)

```
## Message creation parameters
# How many event generators
Events.nrof = 1
# Class of the first event generator
Events1.class = MessageEventGenerator
# (following settings are specific for the MessageEventGenerator class)
# Creation interval in seconds (one new message every 25 to 35 seconds)
#Events1.interval = 500,700
Events1.interval = 550,650
# Message sizes (500kB - 1MB)
Events1.size = 500
# range of message source/destination addresses
Events1.hosts = 0,49
# Message ID prefix
Events1.prefix = M
## Movement model settings
# seed for movement models' pseudo random number generator (default = 0)
MovementModel.rngSeed = 1 #
World's size for Movement Models without implicit size (width, height; meters)
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
## Reports - all report names have to be valid report classes
# how many reports to load Report.nrofReports = 4
# length of the warm up period (simulated seconds)
Report.warmup = 0
# default directory of reports (can be overridden per Report with output setting)
Report.reportDir = reports/tamer # Report classes to load
Report.report1 = MessageStatsReport
Report.report2 = EventLog
Report Report.report3 = ContactTimesReport
Report.report4 = CreatedMessagesReport
```


Figure 3.2: A sample script for the ONE simulator (part 2)

Figure 3.3: A map of Helsinki city with 5 pedestrians and 5 vehicles roaming.

Chapter 4

Optimal Routing in DTN

In this chapter, we formulate an optimization problem of single-copy centralized routing in DTNs, assuming the availability of present and future node contacts and buffer and message information. We implement the problem with three different objectives: minimum delay, minimum number of hops, and maximum number of delivered messages. We solve the problem using parameters from simulated networks. The optimal results provide us with bounds on the performance metrics, and show the room for improvement that should be worked on. Optimal routing with a global observer is just a theoretical model used for performance comparison only, and cannot be implemented practically. This work has been published in [2].

4.1 Problem Setup

We consider a route as an ordered set of contacts from source to destination. Our objective is to find the optimal route for each message that satisfies the objective function. In this formulation, the inputs to the problem are the contact, buffer and message information. The network model including the description of node contacts, node buffers and message attributes is detailed in Section 3.1.

We compare between three objective functions:

- Min-H: Minimizing the number of hops from source to destination,
- Max-M: Maximizing the number of delivered messages, or

- Min-D: Minimizing the end-to-end delay.

The constraints to be satisfied are:

- Buffer: The messages transferred during each contact should not exceed the receiver buffer capacity.
- Flow Conservation: The number of contacts in which each message is sent should be equal to the number of contacts in which that message received, except for its source and destination of that message.
- Source and Destination: There should be only one contact in which each message is sent by its source, and one contact in which each message is received by its destination.
- Contacts order: Contacts in the same route should have increasing order of their starting times.
- Contact duration: The sum of all messages transfer times during a contact should be less than the duration of that contact.
- Message Lifetime: The starting time of the last contact in the route of a message should be less than the message lifetime (Time-To-Live or TTL).

Table 4.1 shows a summary of the symbols used to represent the system parameters with a short description.

4.2 Problem Formulation

To reduce the complexity of the problem, we divide it into two sub-problems:

- Find all the possible paths for message m. The problem is solved for each message individually. The output is a set of paths, P_m, for each message m.
- Among all the possible paths, find the optimal path for each message. This step requires considering all the paths for all the messages in the optimization problem.

Table 4.1: Symbols used to represent system parameters

Symbol	Description
N	Set of all nodes in the network
n	A node in the set of nodes N in the network
B_n	The buffer capacity of node n
b_n	The initial buffer occupancy of node n
C	Set of all contacts
c	A contact between two nodes
S_c	Sender node of contact c
S_c^p	Sender node of contact c that belongs to path p
R_c	Receiver node of contact c
R_c^p	Receiver node of contact c that belongs to path p
t_c	The starting time of contact c
t_c^p	The starting time of contact c that belongs to path p
D_c	The time duration of contact c
M	Set of all messages in the network nodes
m	A message in the set of messages M
L_m	The lifetime (TTL) of message m
d_m	The transmission time of message m
P_m	The set of valid paths of message m
T	Simulation Time

4.2.1 Finding all the possible paths for a message m

The problem is to find all the paths for each message m, by tracing the tree of node contacts starting from the source of message m to its destination. A valid path is the one that has timely ordered contacts from source to destination and does not violate the message lifetime (TTL) constraint.

- Input: Contacts senders, receivers, and starting times, and messages lifetimes.
- Output: A set of paths P_m for each message $m \in M$.

The binary variable x_c is used to represent if contact c is included in the path or not. It has the values $x_c = 1$ if contact c is included in the message m path, or $x_c = 0$ if not. The same variable with a superscript n, x_c^n, indicates that n is one of the two nodes of contact c. Other symbols are described in Table 4.1. The objective is to satisfy a set of constraints, as shown in Figure 4.1.

Flow Conservation	$\sum_{c1 \in C} x_{c1}^n - \sum_{c2 \in C} x_{c2}^n = 0$	$\forall n \in N,\ if\ n = R_{c1} = S_{c2}$
Source and Destination	$\sum_{c \in C} x_c = 1,$	if S_c is the source
	$\sum_{c \in C} x_c = 1,$	if R_c is the destination
Contacts Order	$t_{c1}x_{c1} + (T - t_{c2})x_{c2} < T,$	$if\ R_{c1} = S_{c2}, \forall c1, c2 \in C,$ $T > t_{c1},\ T > t_{c2},$
Message Lifetime	$t_c x_c < L_m,$	$\forall c \in C$
Boundary Limits	$x_c, x_{c1}, x_{c2} \in \{0, 1\},$ $t_{c1}, t_{c2}, L_m \geq 0$	

Figure 4.1: An optimization model to find all the possible paths of message m

The model is explained as follows:

- Flow Conservation: Every node $n \in N$, except for the source and destination, included in the selected path of a message m should be associated with two contacts:
  - A contact in which the node receives the message from the previous node on the route.
  - A contact in which the node sends the message to the next node on the route.

$$\begin{aligned} &\sum_{c1 \in C} x_{c1} - \sum_{c2 \in C} x_{c2} = 0, \\ &\forall n \in N, if\ n = R_{c1} = S_{c2} \end{aligned} \tag{4.1}$$

- Source and Destination: The source node is associated with only one contact in which message m is sent by that node. The destination node is associated with only one contact in which message m is received by that node.

$$\begin{aligned} &\sum_{c \in C} x_c = 1, \text{if } S_c \text{is the source of message } m, \\ &\sum_{c \in C} x_c = 1, \text{if } R_c \text{is the destination of message } m \end{aligned} \tag{4.2}$$

- Contacts order: Contacts along the message route should have increasing order of their starting times. If there are two contacts $c1$ and $c2$ that are candidates to be in message m route in the order $c1$-$c2$, then it should be verified that $t_{c1} < t_{c2}$. Both t_{c1} and t_{c2} are to be less than the simulation end time T. Given these requirements, the constraint can be formulated as follows:

$$\begin{aligned} &t_{c1} x_{c1} + (T - t_{c2}) x_{c2} < T, \\ &if\ R_{c1} = S_{c2}, \forall c1, c2 \in C,\ T > t_{c1}, T > t_{c2} \end{aligned} \tag{4.3}$$

  This formulation ensures that having one or both contacts in the route does not violate the constraint, as shown in Figure 4.2.

x_{c1}	x_{c2}	$t_{c1}x_{c1} + (T - t_{c2})x_{c2}$	$< T$
0	0	0	Yes
0	1	$T - t_{c2}$	Yes
1	0	t_{c1}	Yes
1	1	$t_{c1} + T - t_{c2}$	Yes

Figure 4.2: Verifying the “Contacts Order” constraint

- Message Lifetime: The starting time of the any contact on the message route should be less than the message lifetime (TTL).

$$t_c x_c < L_m, \ \forall c \in C \tag{4.4}$$

The output, which is the set of paths P_m for each $m \in M$, is fed as an input to the second part of the problem, which aims to find the optimal path for each message m, as explained in Section 4.2.2.

4.2.2 Finding the optimal path for each message

- Input: Set of paths, P_m, for each message m.
- Output: One path (the optimal route) for each message m.

In this formulation, we use a binary variable x_p to indicate if path p is optimal or not. If path p is found to be optimal, then $x_p = 1$, otherwise $x_p = 0$. We use the symbol $\sqsubset$ to denote a contact is included in a specific path, and the symbol $\sqsupset$ to denote a path includes a specific contact. For example, $c \sqsubset p$ denotes that contact c is one of the contacts constituting path p, and $p \sqsupset c$ denotes that path p includes contact c among its contacts. The formulation is modeled as an optimization problem, as shown in Figure 4.3. The objective is to minimize the total number of hops (contacts) for all the routes. The solution should satisfy the following set of constraints:

- Buffer: The total amount of message bytes transferred during each contact should not exceed the receiver buffer capacity.

$$\begin{aligned}&\textstyle\sum_{m\in M}\sum_{p1\in P_m}\sum_{c\in C, c\sqsubset p1} x_{p1}\\&-\textstyle\sum_{m\in M}\sum_{p2\in P_m}\sum_{c\in C, c\sqsubset p2} x_{p2} \leq B_{R_c} - b_{R_c}\\&\forall c \in C, if\ R_c = R_c^{p1} = S_c^{p2},\\&and\ t_c^{p1}, t_c^{p2} < t_c\end{aligned} \tag{4.5}$$

where $p1$ represents all the incoming past and the expected current traffic to the receiver buffer, while $p2$ represents all the outgoing traffic. The constraint ensures that, during any contact, the total amount of transmitted bytes does not exceed the buffer capacity.

Min-H	Minimize$\sum_{m \in M} \sum_{p1 \in P_m} \sum_{c \in C, c \sqsubset p1} x_{p1}$	
OR		
Min-D	Minimize $max(t_c x_p)$	
OR		
Max-M	Maximize $\sum_{m \in M} \sum_{p1 \in P_m} x_{p1}$	
subject to		
Buffer Capacity	$\sum_{m \in M} \sum_{p1 \in P_m} \sum_{c \in C, c \sqsubset p1} x_{p1}$ $- \sum_{m \in M} \sum_{p2 \in P_m} \sum_{c \in C, c \sqsubset p2} x_{p2}$ $\leq B_{R_c} - b_{R_c}$	$\forall c \in C,$ $if\ R_c = R_c^{p1} = S_c^{p2},$ $t_{p1.c}, t_{p2.c} < t_c$
Contacts Duration	$\sum_{m \in M} \sum_{p \in P_m, p \sqsupset c} d_m x_p \leq D_c,$	$\forall c \in C$
Paths per Message	$\sum_{p \in P_m} x_p$ =1,	$\forall m \in M$
Boundary Limits	$x_p, x_{p1}, x_{p2} \in \{0, 1\}$, $d_m, D_c \geq 0$	

Figure 4.3: An optimization model to find the optimal path of each message

- Contact duration: The sum of all message transfer times during a contact should be less than the duration of that contact.

$$\textstyle\sum_{m \in M} \sum\limits_{p \in P_m, p \sqsupset c} d_m x_p \leq D_c, \forall c \in C \tag{4.6}$$

- Paths per Message: For each message, there should be only one path selected.

$$\textstyle\sum_{p \in P_m} x_p = 1, \forall m \in M \tag{4.7}$$

The objective is one of the three considered objectives

- Min-H: Minimize the number of hops of all the routes, by minimizing the number of contacts.
- Min-D: Minimize the end-to-end delay, by selecting the route with the minimum time of the last contact.
- Max-M: Maximize the number of delivered messages, by maximizing the number of selected routes.

4.3 Performance Comparison and Simulation Results

We consider a sparse mobile network where nodes are connected to each other at discrete time intervals via wireless links. Nodes communicate when they get into the transmission range of each other. In such event, they are said to be in contact. The inter-contact time, that is the interval between two contacts of the same pair of nodes is modeled using the power law distribution explained in Section 3.5. To avoid partitioning, each node is made to contact with at least one node. The maximum number of nodes to contact with is drawn from a uniform distribution of up to one fifth of the total number of nodes in the network. Packets are created at every node using Poisson distribution, and assigned a random destination. When generated, each packet is associated with a unique identifier, time of creation and a time to live (TTL). Table 4.2 shows the values used for the network parameters.

We study the impact of varying the buffer capacity, the traffic load, and the packet time to live (TTL) on the performance of the routing protocol with the objectives:

- Min-H: Minimize the number of hops.

Table 4.2: Network Parameters

Parameter	Value
Simulation time	12 *hours*
Number of nodes	50
Minimum inter-contact time	4 *hours*

- Max-M: Maximize the number of delivered messages.
- Min-D: Minimize the end-to-end delay.

The performance measures considered are:

- Delivery ratio: The number of delivered packets to the number of packets generated.
- Average number of hops: Total number of hops divided by the number of delivered packets. This metric is used an indicator for the energy consumption. The less the number of hops, the less is the number of transmissions, and therefore, the less is the energy consumption.
- Average delay: Total delay of all delivered packets divided by the number of delivered packets.

The mathematical notation of the performance metrics is provided in Section 3.2. The 'average number of hops' metric for optimal routing has the same formula as the packet delivery cost. In optimal routing, there is only a single copy and no packet is transmitted from the source unless there is a route to destination.

4.3.1 Impact of Varying the Buffer Capacity

As shown in Figure 4.4, the delivery ratio for the three protocols increases with increasing the buffer. This is because of the ability to store more packets in the larger buffers, and therefore, avoid their early dropping. It is also noticed that the delivery ratio for the three objectives is almost the same for this scenario, except for little outperformance of the Max-M and Min-H over the Min-D in the smaller buffer capacities.

Figure 4.5 shows the impact of varying the buffer capacity on the average number of hops per one route. It can be noticed that Min-H significantly reduces the number of hops

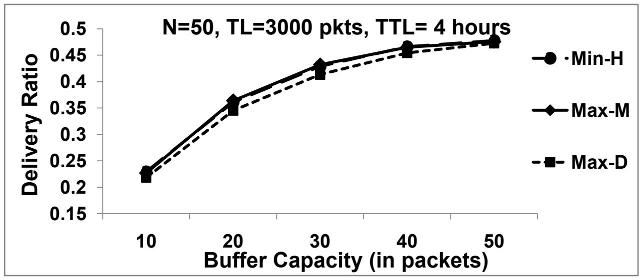


Figure 4.4: Impact of varying buffer capacities on the delivery ratio

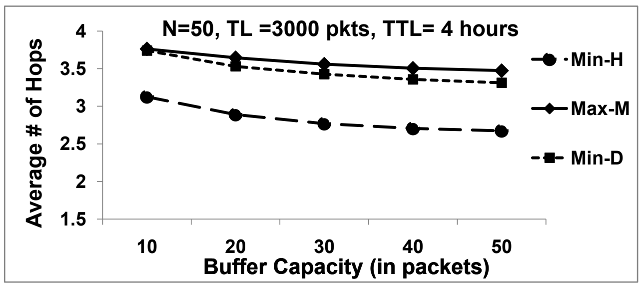


Figure 4.5: Impact of varying buffer capacities on the number of hops

with a ratio of approximately 0.75 to that of the other objectives. Increasing the buffer capacity slightly reduces the number of hops, especially at small capacities.

The outperformance of Min-H over Min-D in the delivery ratio and the number of hops is paid for by the end-to-end delay. Min-D outperforms the other objectives in minimizing the delay by approximately 10 minutes per packet (around 7% of the delay). However, in a DTN, where data is delay tolerant, it is acceptable to increase the packet delay by a small percentage to gain higher delivery ratio and lower energy consumption.

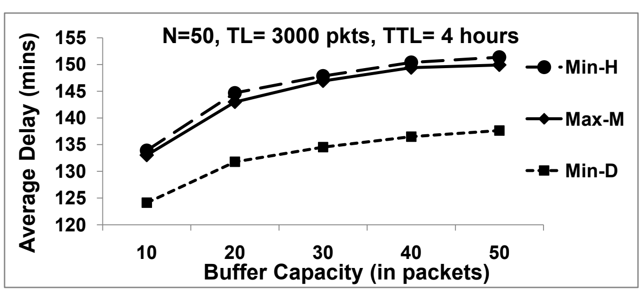


Figure 4.6: Impact of varying buffer capacities on the average delay

4.3.2 Impact of Varying the Traffic Load (TL)

As shown in Figure 4.7, injecting more packets into the network, causes the dropping of many of the stored packets, and therefore, decreases the delivery ratio of all the protocols. Min-H and Max-M shows better delivery ratio than the Min-D at large traffic loads.

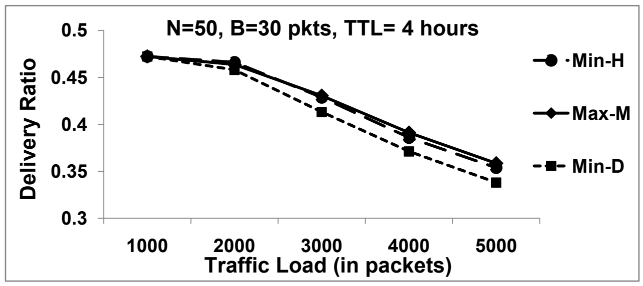


Figure 4.7: Impact of varying traffic loads on the delivery ratio

In Figure 4.8, number of hops slightly increases with increasing the traffic load. It is also noticed that Min-H outperforms the other objectives with approximately the same ratio for all the traffic loads.

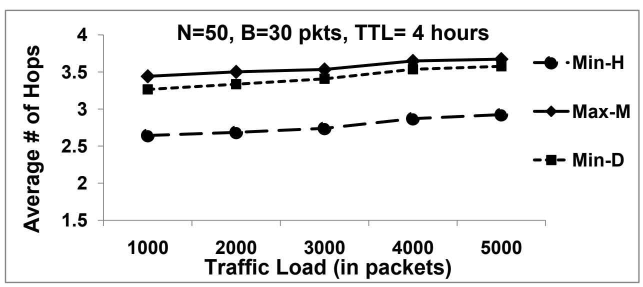


Figure 4.8: Impact of varying traffic loads on the number of hops

Min-D always keeps its significant reduction in the end-to-end delay over the other two objectives, as shown in Figure 4.9.

4.3.3 Impact of Varying the Packets TTL

It is observed from Figure 4.10 that the delivery ratio increases with increasing the packet TTL. This is because increasing the packet lifetime increases the probability to find a route to the destination. The rate of increase is higher at small TTL values, and begins to slow

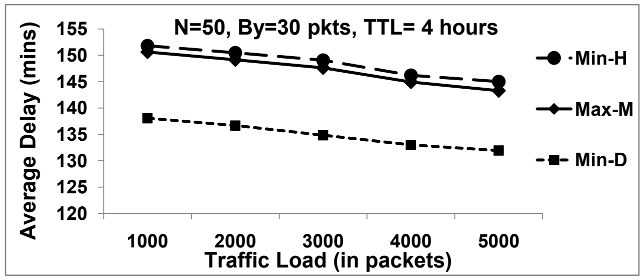


Figure 4.9: Impact of varying traffic loads on the average delay

down at higher values. This improved delivery ratio may also be explained by finding longer routes (consisting of more hops to the destination), which can been seen obviously in Figure 4.11. In addition, it can be explained by having the ability to wait for delayed contacts resulting in higher-delay routes as shown in Figure 4.12.

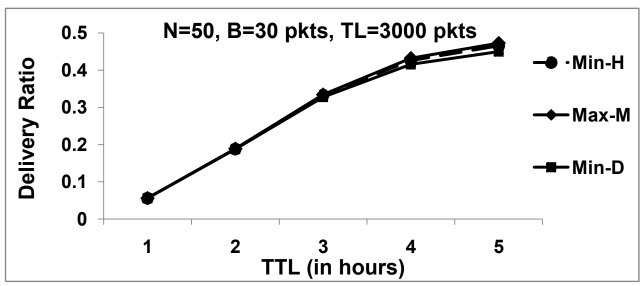


Figure 4.10: Impact of varying packets TTL on the delivery ratio

4.4 Summary & Conclusion

Optimal routing in DTN is a theoretical perspective assuming the availability of present and future knowledge of contacts, nodes and packets information. In rare cases, it could be approximately feasible, such as in a city bus network. In this chapter, we formulated an optimization problem that finds the optimal routes in a DTN using three objective functions: minimizing the end-to-end delay (Min-D), minimizing the number of hops(Min-H), and maximizing the number of delivered messages (Max-M). The problem is subject to the node constraints (storage capacity), contact constraints (order and duration), and message constraints (lifetime and transmission times). We implemented the model with the three objectives in a sparse mobile network. The impact of buffer capacities, traffic loads and

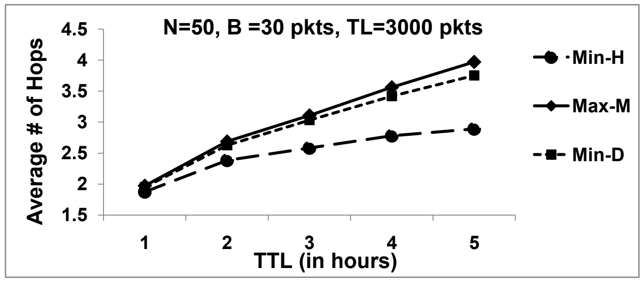


Figure 4.11: Impact of varying packets TTL on the number of hops

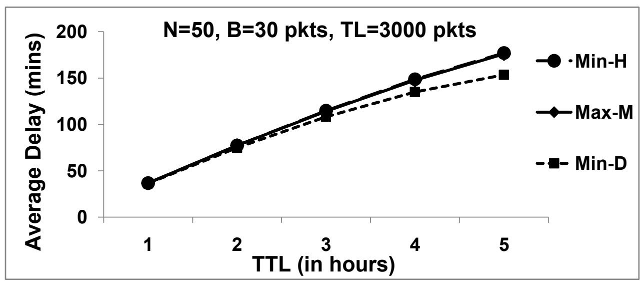


Figure 4.12: Impact of varying TTL on the average delay

packet TTL values are considered on the performance metrics, which are: delivery ratio, number of hops and end-to-end delay. Simulation results show that minimizing the number of hops achieves higher delivery ratio than minimizing the delay and almost the same as maximizing the number of delivered messages for most of the buffer capacities, traffic loads and TTL values. In addition, minimizing the number of hops proves to significantly reduce the number of transmissions in the network which results in considerable energy saving. In a delay tolerant network, it is acceptable to tolerate a small increase in the delay, to gain higher delivery ratio and lower energy consumption. In the following chapters, we will use the Min-H results to compare with the other distributed protocols, and it will be named OPT as an abbreviation of optimal routing.

Chapter 5

Performance Comparison of DTN Routing Protocols

Routing protocols developed for DTNs adapt themselves to their challenging environment by propagating multiple copies of data packets to increase the probability that one of the copies reaches the destination. Nodes receiving the packet copies store them until they meet other nodes or meet their destinations. Simple DTN routing protocols blindly forward packet copies to any node they become in contact without using a selection criteria. They range from the full network flooding to the partial flooding. The Blind-Routing approach may achieve high delivery ratio of data if provided enough storage and energy resources. On the other hand, it has its drawbacks, such as burdening the buffer and the inefficient use of the contact duration. Other routing protocols tend to restrict forwarding of data packets to selected nodes. Using some information collected from the network, they guide the packet copies to their destinations by selecting the relay nodes. The Guided-Routing approach fails if the network topology is changing faster than the rate of information gathering.

In this chapter, we study four well-known distributed DTN routing protocols: EPIDEMIC, Spray-and-Wait (SnW), PROPHET and MAXPROP. We introduce a procedural presentation of the routing protocols together with a summary of similarities and differences. EPIDEMIC is an example of a Blind-Routing Full-Flooding protocol. SnW is an example of a Blind-Routing Partial-Flooding protocol. PROPHET is an example of a Guided-Routing protocol with a first in first out (FIFO) packet selection mechanism. MAXPROP is an example of a Guided-Routing protocol which favors packets with minimum number of hops. We compare the routing protocols with each other and with the performance results of optimal routing with minimum hops. We conducted several simulation experiments to show the impact of changing buffer capacity, packet lifetime, packet generation rate, and number of nodes on the performance metrics. The chapter is concluded by providing guidelines to develop an efficient DTN routing protocol. This work has been published in [7].

5.1 Epidemic Routing Protocol

Epidemic routing [78] is the first routing protocol proposed for sparse networks. Each packet generated is assigned a unique ID that is associated with it and all its copies till they are dropped or delivered to the destination. The list of all the packets IDs in a node's buffer is called the summary vector. When two nodes meet, they exchange their summary vector. All data packets that are stored in one node and not in the other are ordered on a first come first serve (FCFS) basis to be transmitted to the other node. Packet transfer then starts until the contact duration ends. Assuming that the contact duration is long enough to transfer all the uncommon packets, then the two nodes will have the same packet list after their contact ends. Given unlimited buffer size, long enough contact durations, unlimited lifetime for the data packets, and non-infinite partitioned network, *Epidemic* routing guarantees the delivery of all the packets to their destinations. In addition, it guarantees the lowest end-to-end delay, because each packet is routed on all possible paths from the source, and one of the copies will be on the shortest path. Procedure 1 shows a pseudo-code of the *Epidemic* protocol during the contact of two nodes.

The main drawback of *Epidemic* routing is its huge consumption of the limited resources, such as memory, energy and contact duration. Later work have been proposed to reduce the inefficient resource consumption. In [69], the authors used additional infrastructure (infostations) that serves as multiple destinations for the same packet. In [14], *Epidemic* routing is used with the aid of autonomous agents to collect and distribute the data. It has been used in [33] as a fallback when no other method can perform better. Learning the movement of other nodes and using an efficient buffer management mechanism reduces the overhead of *Epidemic* routing [23]. Other protocols use relatively small number of allowed copies or hops [28, 30, 56, 78, 70, 72, 74]. Analysis of the delay in Epidemic routing is presented in [71, 72].

Procedure 1 Epidemic Routing Protocol

```
Procedure Name: OnContact
Input: node a, node b, integer ContactDuration
DropExpiredPackets(a,b) /* Drop packets with their lifetime expired in both nodes */
ExchangeSummaryVector(a,b)
if ContactDuration > 0 then
   pkt=GetPacket(a)
   if pkt then
      if NotReceivedBefore(pkt,b) then
         if IsDestination(pkt,b) then
            SendPacket(pkt,a)
            ConsumePacket(pkt,b)
         else
            SendPacket(pkt,a)
            StorePacket(pkt,b)
         end if
         ContactDuration=ContactDuration-size(pkt)
      end if
   end if
end if
```

5.2 Spray and Wait (SnW) Routing Protocol

DTNs usually involve devices that are energy-sensitive in which saving energy becomes one of its main objectives, if not the main one. Energy consumption is mostly incurred in the communication process (transmission and reception). To save energy, it is required to decrease the number of transmissions and receptions. Motivated by this fact, the authors in [74] proposed the Spray-and-Wait (SnW) routing protocol. The idea of SnW is to limit the number of packet copies in the network. A packet copy, transferred from a node to another, is associated with the number of further copies allowed for the second node to distribute. This number is decreased by the number of transfers for this packet at each node. When the allowed number of copies reaches one, the carrying node stops generating more copies of the packet and keeps its single copy until it either meets the destination or the packet is dropped because of a buffer overflow or lifetime expiry. A binary version of SnW is also proposed in [74], in which each node is allowed to use half the number of copies allowed for the packet and the other half is left for the receiving node. The pseudo-code for the binary SnW is shown in Procedure 2.

Both versions of SnW, regular and binary, proved to perform better than the full flooding protocol, *Epidemic*, in terms of average packet delay and energy consumption. However, SnW still suffers from the blind selection of the next-hop nodes which may degrade the packet delivery ratio.

5.3 PROPHET Routing Protocol

The Probabilistic Routing Protocol using History of Encounters and Transitivity ($PROPHET$) is proposed in[50]. The protocol estimates a node metric called delivery predictability, $P(a,b)$, at each node a for each destination b. When two nodes meet, they update their delivery predictability towards each other. Then the two nodes exchange their delivery predictability list towards other nodes to each other to update their delivery predictability towards the other nodes using the following equations:

- Direct update:

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

  where $P_{init} \in [0,1]$ is an initialization constant. This update is done when the two nodes a and b come into direct contact with each other.

Procedure 2 Binary Spray And Wait Routing Protocol

```
Procedure Name: OnContact
Input: node a, node b, integer ContactDuration
DropExpiredPackets(a,b) /* Drop packets with their lifetime expired in both nodes */
ExchangeSummaryVector(a,b)
if ContactDuration > 0 then
  pkt=GetPacket(a)
  if pkt then
    if NotReceivedBefore(pkt,b) then
      if IsDestination(pkt,b) then
        SendPacket(pkt,a)
        ConsumePacket(pkt,b)
      else
        NrOfCopies=GetNrOfCopies(pkt,a)
        if NrOfCopies > 1 then
          SendPacket(pkt,a)
          StorePacket(pkt,b)
          SetNrOfCopies(pkt,a,NrOfCopies/2)
          SetNrOfCopies(pkt,b,NrOfCopies/2)
        end if
      end if
      ContactDuration=ContactDuration-size(pkt)
    end if
  end if
end if
```

- Transitive update:

$$P_{(a,b)} = P_{(a,b)_{old}} + (1 - P_{(a,b)_{old}})P_{(a,c)}P_{(c,b)}\beta$$

  where $\beta \in [0, 1]$ is the transitivity constant which reflects the impact of transitivity on the delivery predictability. This equation updates the delivery predictability of node a towards node a through the transitive contact between a and c.

- Aging:

$$P_{(a,b)} = P_{(a,b)_{old}}\gamma^k$$

  where $\gamma \in [0, 1]$ is the aging constant, and k is the number of time units that have elapsed since the last time the metric was aged. This equation decreases the delivery predictability by the time passed without direct between the two nodes a and b.

PROPHET provides a partial guiding towards the destination by tracing the contacts between nodes and assigning weights to these contacts whether they were directly or through intermediate nodes. Therefore, *PROPHET* is expected to outperform the blind protocols in the delivery ratio. On the other hand, it is expected that the average packet delay may increase due to waiting for for a good next node in the path. A pseudo-code for *PROPHET* is provided in algorithm 3.

5.4 MAXPROP Routing Protocol

The *MAXPROP* protocol proposed in[12] estimates a node metric, $P(a, b)$, similar to *PROPHET*. When two nodes meet, they strengthen the link between each other by adding a constant α which is set to equal 1 in the protocol. Then the two nodes divide their delivery predictability towards all the nodes including each other by $1 + \alpha$ so that the sum of all delivery predictability remains 1.

$$\begin{aligned} P_{(a,b)} &= P_{(a,b)} + 1 && \text{direct contact between } a \text{and } b \\ P_{(a,c)} &= P_{(a,c)}/(1 + \alpha) && c \text{is every other node including } b \end{aligned}$$

where $\alpha \in [0, 1]$ is the updating constant which is set to 1 in their work.

The node metric is used only when the hop count of the packet is greater than a certain threshold. The main contribution of MAXPROP is in its buffer management. Packets are sorted according to their hop count, if the hop count is below a certain threshold. Otherwise, packets are sorted with their delivery predictability. In this way, *MAXPROP* favors packets with less hop count to spread in the network.

Procedure 3 PROPHET Routing Protocol

```
Procedure Name: OnContact
Input: node a, node b, integer ContactDuration
DropExpiredPackets(a,b) /* Drop packets with their lifetime expired in both nodes */
ExchangeSummaryVector(a,b)
UpdateDeliveryPredictability()
if ContactDuration > 0 then
   pkt=GetPacket(a)
   if pkt then
      if NotReceivedBefore(pkt,b) then
         if IsDestination(pkt,b) then
            SendPacket(pkt,a)
            ConsumePacket(pkt,b)
         else
            DPn1=DeliveryPredictability(pkt,a)
            DPn2=DeliveryPredictability(pkt,b)
            if DPn2 > DPn1 then
               SendPacket(pkt,a)
               StorePacket(pkt,b)
            end if
         end if
         ContactDuration=ContactDuration-size(pkt)
      end if
   end if
end if
```

Procedure 4 MAXPROP Routing Protocol

```
Procedure Name: OnContact
Input: node a, node b, integer ContactDuration
DropExpiredPackets(a,b) /* Drop packets with their lifetime expired in both nodes */
ExchangeSummaryVector(a,b)
UpdateDeliveryPredictability()
SortPackets() /* Using MAXPROP sorting criteria */
if ContactDuration > 0 then
    pkt=GetPacket(a)
    /* pkt is the packet with the minimum hop count, or higher delivery predictability */
    if pkt then
        if NotReceivedBefore(pkt,b) then
            if IsDestination(pkt,b) then
                SendPacket(pkt,a)
                ConsumePacket(pkt,b)
            else
                SendPacket(pkt,a)
                StorePacket(pkt,b)
            end if
            ContactDuration=ContactDuration-size(pkt)
        end if
    end if
end if
```

Table 5.1: Protocol Parameters

Protocol	Parameter	Value
	Initialization constant	0.75
PROPHET	Transitivity constant	0.25
	Aging constant, γ	0.98
SnW	Initial number of copies	5

5.5 Performance Metrics and Simulation Setup

To compare the performance of the heuristic protocols with each other and with the performance results of the optimal routing, we built a DTN simulator in MATLAB. The simulator takes as inputs the starting times and durations of node contacts. The optimal routing problem is solved using an open source mixed integer linear programming package LPSOLVE [1]. We conducted the experiments using real life movement scenarios of pedestrians and vehicles in a city. These inputs are generated and recorded using the ONE simulator [44].

We implemented a first-in-first-out (FIFO) queuing system in Epidemic, SnW, and PROPHET, as they did not specify their queuing policy in their work. Each point in the results figures is the average of ten repetitive experiment results with a degree of confidence 0.95, using the confidence method explained in Section 3.3. We consider three metrics to measure the performance of the different protocols, which are Delivery ratio, Delivery cost and Average packet delay. The metrics are explained in Section 3.2.

We conducted four set of experiments to study the impact of the following parameters:

- Buffer Capacity,
- Packet lifetime or time-to-live (TTL),
- Node density by changing the number of nodes in the network, and
- Traffic load by changing the packet generation rate.

Table 5.1 shows the values used for the parameters of the protocols in all the experiments, while Table 5.2 shows the network parameters.

Table 5.2: Network Parameters

Parameter	Pedestrians	Vehicles
#Hosts	5,20,30,45,60	5,10,15,25,30
Speed	0.5-1.5 m/s	2,7-13.9 m/s
Movement	ShortestPath MapBased Movement	
Buffer capacity	2-10 Mbytes	
Packet TTL	2-10 Hours	
Average Packet Inter-generation time	10,30,60,300,600 seconds	
Transmission speed	5 Mbps	
Simulation time	12 Hours	

5.6 Simulation Results

Table 5.3 shows the summary of the results obtained from the different experiments. We presented the minimum and maximum values for the three performance metrics in the five experiments. Delivery cost is measured in number of packets, and average delay is measured in minutes. Protocols are ordered from the highest to lowest in delivery ratio, and from lowest to highest in other metrics. The detailed experiments are shown in the following subsections.

5.6.1 Impact of Varying the Buffer Capacity (B)

From Figure 5.1, it can be seen that increasing the buffer capacity increases the delivery ratio of all the protocols, as long as the amount of bytes of the propagating packets are more than the buffer space. The delivery ratio settles when the buffer space is larger than that of the propagating data, i.e. no packets are dropped because of buffer overflow. The delivery ratio of both optimal routing formulations settles at a smaller value of the buffer capacity because it propagates only a single copy of each packet. It is also noticed that among the distributed heuristic (non-optimal) protocols, *MAXPROP* protocol provides the highest delivery ratio, and *Epidemic Routing* provides the lowest one.

Increasing the buffer capacity decreases the dropped packets, as shown in Figure 5.2. Once the amount of dropped packets approaches zero, the delivery ratio will become constant.

The high delivery ratio for *MAXPROP* is achieved with a large cost (network overhead)

Table 5.3: Summary of Performance Comparison of DTN Routing Protocols: EPIDEMIC, PROPHET, SnW, MAXPROP, MINH and MIND

Impact of	Buffer		TTL		N		TL	
Delivery Ratio	MIND	0.83-0.90	MIND	0.79-0.90				
	MINH	0.74-0.90	MINH	0.79-0.90				
	MAXPROP	0.57-0.90	MAXPROP	0.80-0.90	MAXPROP	0.90-0.97	MAXPROP	0.43-0.98
	PROPHET	0.45-0.81	SnW	0.78-0.81	SnW	0.81-0.94	SnW	0.33-0.95
	SnW	0.42-0.81	PROPHET	0.69-0.82	PROPHET	0.82-0.88	PROPHET	0.26-0.95
	EPIDEMIC	0.36-0.72	EPIDEMIC	0.74-0.80	EPIDEMIC	0.74-0.77	EPIDEMIC	0.74-0.77
Comments	Numbers shown are the minimum and maximum values. Ordered from high to low. The higher the better.							
Delivery Cost	MINH	0.21-0.23	MINH	0.22-0.44				
	MIND	0.94-0.97	MIND	0.86-0.94				
	PROPHET	3.1-3.5	PROPHET	2.8-3.5	SnW	4.2-5.9	SnW	4.6-5.8
	SnW	3.7-4.2	SnW	4.2-4.4	PROPHET	3.7-7.0	MAXPROP	28-36
	MAXPROP	4.5-5.1	MAXPROP	5.1-5.3	MAXPROP	5.1-6.7	PROPHET	20-29
	EPIDEMIC	5.6-6.9	EPIDEMIC	6.5-6.8	EPIDEMIC	6.8-8.6	EPIDEMIC	36-46
Comments	Numbers shown are the minimum and maximum values. Ordered from low to high. The lower the better.							
Average Delay	MIND	58-59	MIND	46-59				
	MAXPROP	58-63	MAXPROP	45-58	MAXPROP	18-58	MAXPROP	24-60
	SnW	62-72	SnW	46-62	EPIDEMIC	19-62	EPIDEMIC	24-72
	EPIDEMIC	63-73	EPIDEMIC	45-62	SnW	39-62	SnW	40-81
	PROPHET	72-75	PROPHET	46-72	PROPHET	24-72	PROPHET	39-82
	MINH	108-118	MINH	54-116				
Comments	Numbers shown are the minimum and maximum values. Ordered from low to high. The lower the better.							

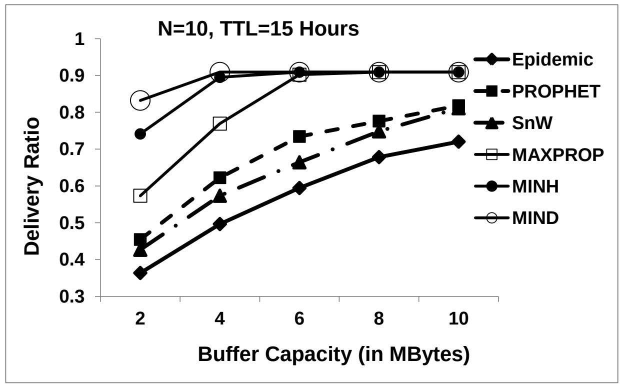


Figure 5.1: Impact of changing buffer capacities on Delivery Ratio.

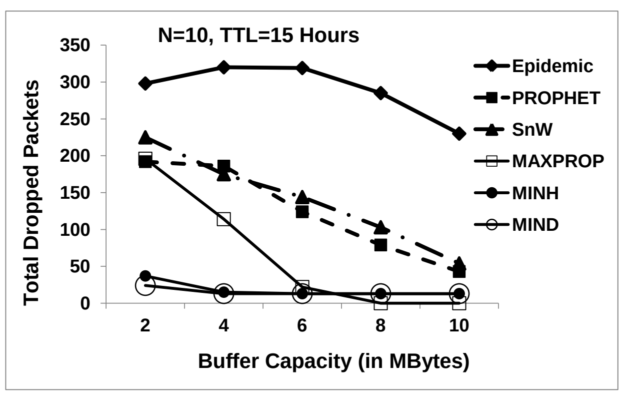


Figure 5.2: Impact of changing buffer capacities on Dropped Packets.

of delivering a packet, as shown in Figure 5.3. The large network overhead also represents high energy consumption during transmissions and receptions. Both optimal objectives guarantee the least cost because they only propagate a single packet copy. The low cost of *SnW*, despite its blind routing as *EPIDEMIC*, is because of its partial flooding. The highest cost is for *EPIDEMIC* because of its full flooding behavior.

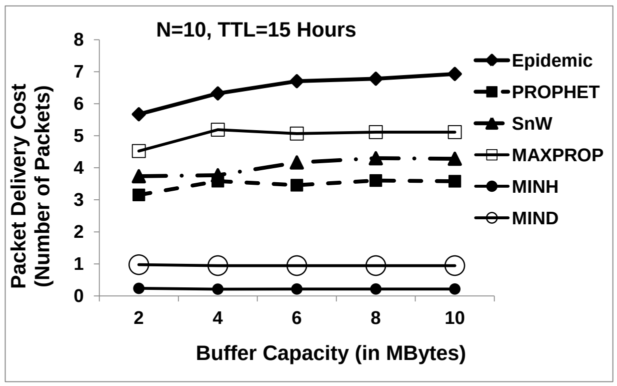


Figure 5.3: Impact of changing buffer capacities on the cost of packet delivery.

In Figure 5.4, the optimal protocol MINH has the highest delay because its optimality is in number of hops, while *MIND* achieves the lowest delay. *MAXPROP* provides the lowest delay among the heuristic protocols because of its efficient node selection and buffer management mechanism.

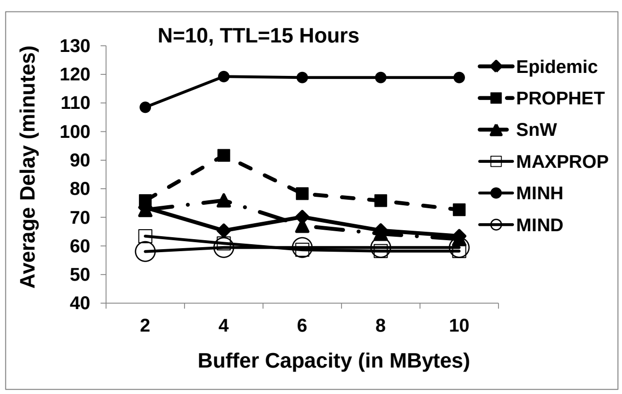


Figure 5.4: Impact of changing buffer capacities on Average Packet Delay.

Impact of Varying the Packet Lifetime (*TTL*)

Increasing the lifetime (TTL) of data packets, increases the delivery ratio up to a maximum value, as shown in Figure 5.5. On the other hand, it overloads the buffer space available which may lead to an increase in dropping the stored packets. Overloading effect is significant in the case of *EPIDEMIC* where delivery ratio is found to be decreasing at values $TTL > 4$ hours as a result of the increased dropping because of buffer overflow, as shown in Figure 5.6.

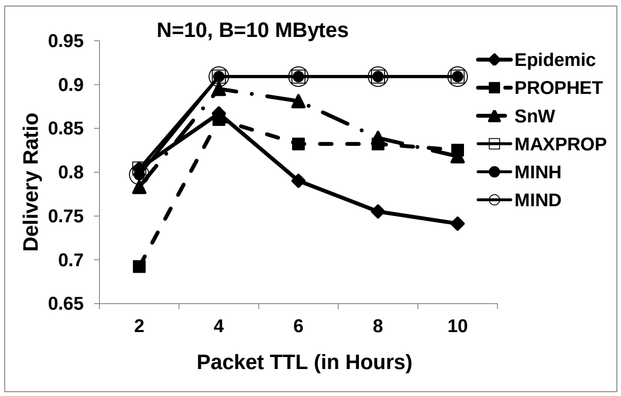


Figure 5.5: Impact of changing packet TTL on Delivery Ratio.

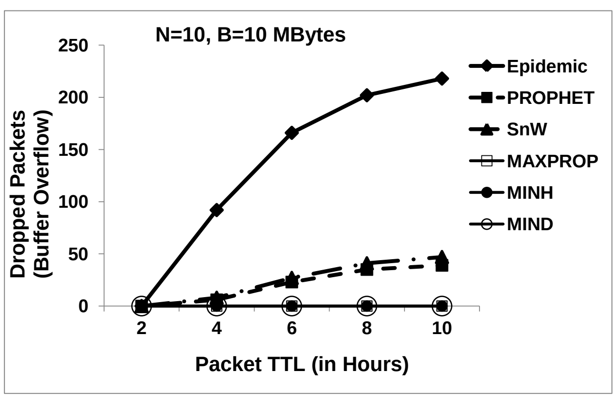


Figure 5.6: Impact of changing packet TTL on Dropped Packets.

Although transmissions and receptions increase with the higher values of TTL, delivered packets also increased. Therefore, packet delivery cost is found to be almost constant as shown in Figure 5.7.

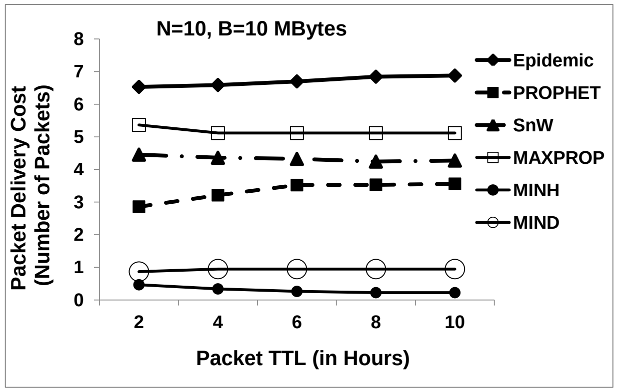


Figure 5.7: Impact of changing packet TTL on cost of packet delivery

It is intuitive to expect a higher average packet delay with increasing the packets TTL (Figure 5.8), because packets that were supposed to be dropped are allowed to live and reach the destination but with higher delay values. Once the delivery ratio of a protocol settles, the average delay is found to settle.

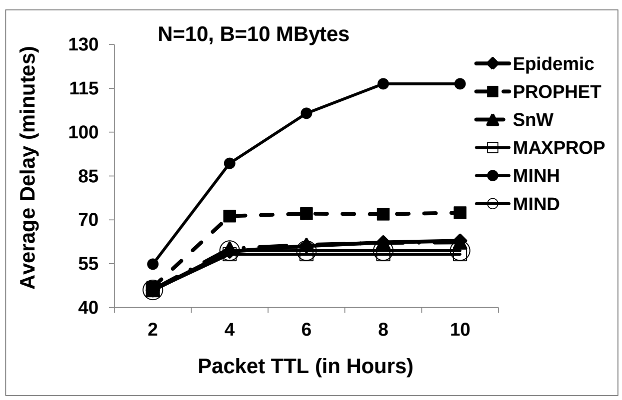


Figure 5.8: Impact of changing packet TTL on Average Packet Delay.

5.6.2 Impact of Varying the Number of nodes (N)

In these experiments, we did not solve the optimal routing because of its long processing times. Therefore, we compare only the heuristic protocols. Increasing the number of nodes, while fixing the packet generation rate, improves the connectivity of the network nodes and allows for more packets to be delivered, as shown in Figure 5.9, but with increased number of hops, as shown in Figure 5.10. Increasing the number of nodes allows also for increased number of packet copies to be generated, as shown in Figure 5.11, unless the protocol limits the number of copies as in SnW. Increasing the node density provides faster paths to destinations which decreases the average packet delay, as shown in Figure 5.12.

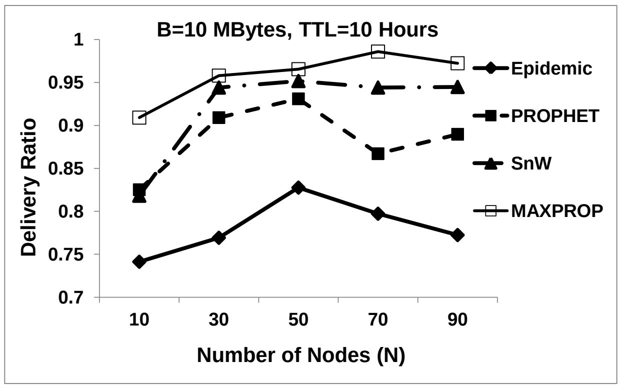


Figure 5.9: Impact of changing number of nodes on Delivery Ratio.

Increasing the buffer capacity decreases the dropped packets, as shown in Figure 5.2. Once the amount of dropped packets approaches zero, the delivery ratio will become constant.

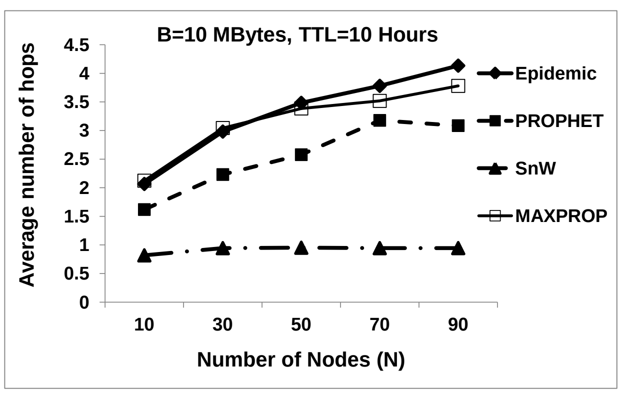


Figure 5.10: Impact of changing number of nodes on number of hops traversed per packet.

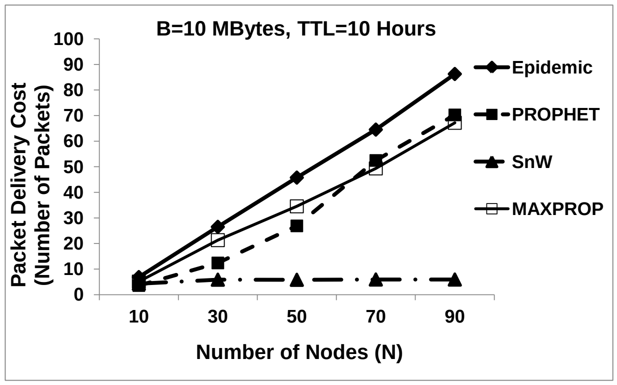


Figure 5.11: Impact of changing number of nodes on the cost of packet delivery.

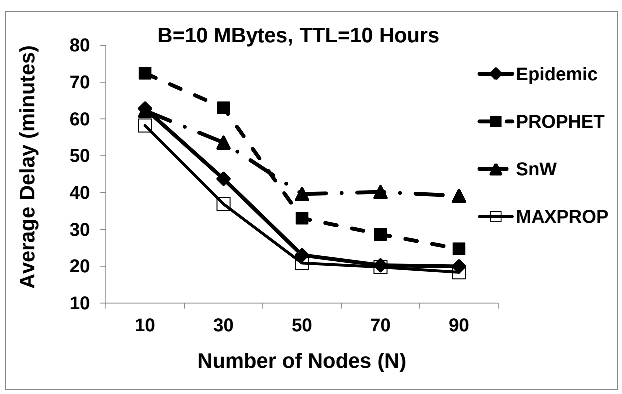


Figure 5.12: Impact of changing number of nodes on Average Packet Delay.

Impact of Varying the traffic load (*TL*)

In this experiment, we ran the simulation five times with different packet generation rates. The inter-generation times are drawn uniformly from the intervals: 8-12, 25-35, 55-65, 250-350, and 550-650 seconds. Therefore, the average generation rates are found to be 379, 121, 66, 12, and 6 packets/hour respectively. Increasing the traffic load, in a network with fixed number of nodes, causes the overloading of buffers and increases the dropping rate, as shown in Figure 5.13. Therefore, the delivery ratio drops significantly, see Figure 5.14. Packet propagation decreases in the network because many packets are dropped, which maintains an almost non-varying delivery cost, as shown in Figure 5.15, with an increasing end-to-end delay, as shown in Figure 5.16.

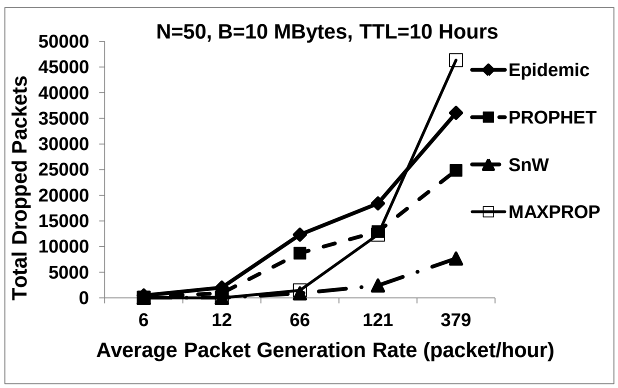


Figure 5.13: Impact of changing the traffic load on Dropped Packets.

5.7 Summary & Conclusion

DTN routing protocols vary according to the amount of information they acquire to take the routing decision. Blind-Routing protocols do not collect any information about the network and, therefore, they do not have a node selection mechanism. They just spread the packets so that one of the copies might reach the destination. The performance is improved if the packet spreading is limited. Guided-Routing protocols seek the possible paths to destinations, by selecting the relay nodes, which improves the delivery ratio. A packet selection mechanism helps in reducing end-to-end delays. A buffer management mechanism helps in providing buffer space for newly generated and arriving packets. In this chapter,

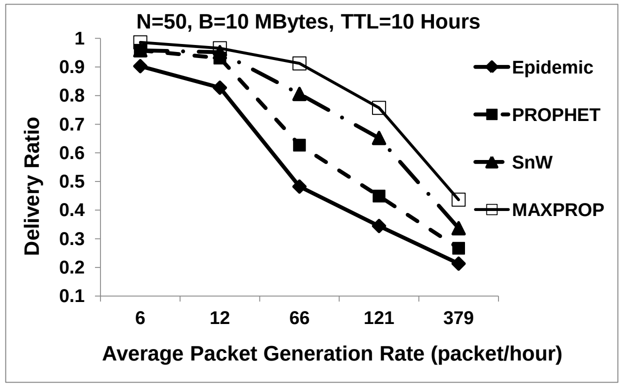


Figure 5.14: Impact of changing the traffic load on Delivery Ratio.

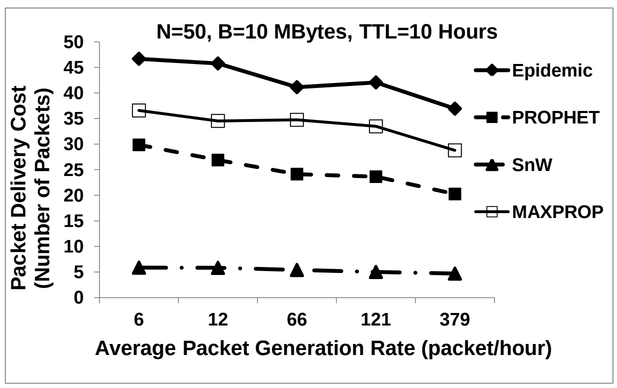


Figure 5.15: Impact of changing the traffic load on cost of packet delivery

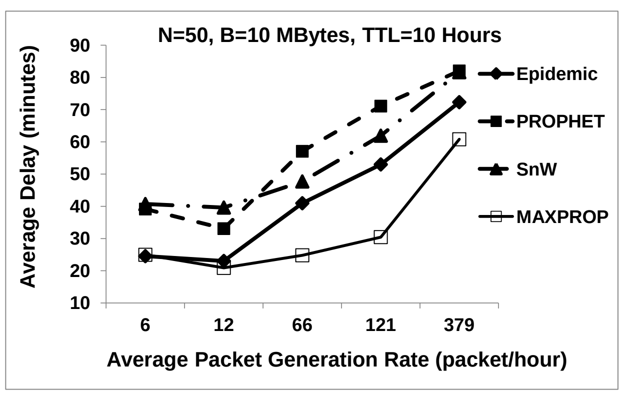


Figure 5.16: Impact of changing the traffic load on Average Packet Delay.

we presented and compared four well-known DTN routing protocols together with an optimal routing formulation. EPIDEMIC is an example of a Blind-Routing Full-Flooding protocol. SnW is an example of a Blind-Routing Partial-Flooding protocol. PROPHET is an example of a Guided-Routing protocol with a first in first out (FIFO) packet selection mechanism. MAXPROP is an example of a Guided-Routing protocol which favors packets with minimum number of hops. We conducted simulations using real life scenarios of vehicles and pedestrians roaming in a city. Results show the outperformance of MAXPROP in delivery ratio and average delay, and the outperformance of SnW and PROPHET in delivery cost. An efficient routing protocol should integrate the node selection, the packet selection, and the buffer management mechanisms to obtain the best performance.

Chapter 6

An Eco-Friendly Routing Protocol for Delay Tolerant Networks

According to the analysis of the distributed heuristic (non-optimal) routing protocols, we come out with the following:

- Full flooding protocols (e.g., *EPIDEMIC*) has the highest delivery ratio and the lowest average packet delay if provided large storage space (relative to the volume of traffic generated). However, in reality, storage space dedicated to delay-tolerant data is expected to be small. In addition, data packets in DTN (bundles) are large in size occupying more space than regular packets.
- Limited flooding protocols (e.g., *SnW*) has low network overhead, which is mapped to low energy consumption. The delivery ratio is almost the same as that of the full flooding protocols at small buffer capacities.
- Guided routing protocols provide high delivery ratios, with middle network overhead (between that of the limited flooding and full flooding), especially at large buffer capacities. On the other hand, the average packet delay is always higher than blind routing.

According to this analysis, an efficient routing protocol should combine the cons of both guided and limited flooding protocols to achieve high delivery ratio with low network overhead. Therefore, it should exploit the available information in the network to guide the packets to the shortest paths towards their destinations. In addition, it should limit the number of copies for each packet in the network to decrease the network overhead and save energy.

We propose a distributed heuristic routing protocol ,Eco-Friendly Routing for DTN (*EFR-DTN*), that aims at reducing the network overhead while providing same or better delivery ratio. The protocol uses a node selection metric that predicts the route to destination (similar to PROPHET), while limiting the number of copies (similar to SnW) to reduce network overhead. We prove by simulation that we can reduce network overhead without losing delivery performance. This work has been published in [3].

6.1 Protocol Design

The *EFR-DTN* protocol can be explained as follows:

- Each packet generated is assigned a unique ID that is associated with it and all its copies till they are dropped or they reach the destination. The list of all the packets IDs in a node's buffer is called the summary vector.
- When two nodes meet, they exchange their summary vectors. All data packets that are stored in one node and not in the other are ordered on a first come first serve (FCFS) basis to be transferred to the other node. Packet transfer then starts until the contact duration ends.
- We used the selection criteria used in the PROPHET protocol, so that a packet is transferred only if its delivery predictability at the other node is higher than that at its current node. The delivery predictability is updated according to the following equations:

$$P_{(a,b)} = \begin{cases} P^{old}_{(a,b)} + (1 - P^{old}_{(a,b)})\alpha & \text{directly} \\ P^{old}_{(a,b)} + (1 - P^{old}_{(a,b)})P_{(a,c)}P_{(c,b)}\beta & \text{indirect} \\ P^{old}_{(a,b)}\gamma^k & \text{otherwise} \end{cases}$$

  where $P^{old}_{(a,b)}$ is the delivery predictability at node a for destination b before the update and $P_{(a,b)}$ is the new value after update, $\alpha \in (0, 1]$ is the updating factor, $\gamma \in [0, 1]$ is the aging constant, k is the number of time units that have elapsed since the last time the metric was aged, and $\beta \in (0, 1]$ is the transitivity constant which reflects the impact of transitivity on the delivery predictability.

- The flooding of data packets is limited using the Binary SPRAY-and-WAIT (SnW) mechanism, so that a node keeps half the number of copies and assigns the other half to the receiving node.
- After a copy of the packet is transferred, the local copy may be dropped depending on the delivery predictability of the node. The higher the delivery predictability of the sender node, the lower is the probability to drop its local copy, and vice versa.
- When the expiry time of a data packet approaches, the carrying node switches to partial flooding of the network with limited copies of the packet. This gives a last chance to the dying packet to catch the destination before it is dropped.

The pseudo-code for the proposed protocol is presented in Procedure 5.

6.2 Performance Metrics and Simulation Setup

We built a DTN simulator in MATLAB. The simulator takes as inputs the starting times and durations of the nodes contacts. We consider a sparse mobile network where nodes are connected to each other at discrete time intervals via wireless links. Nodes communicate when they get into the communication range of each other. In such event, they are said to be in contact. The inter-contact time, that is the interval between two contacts of the same pair of nodes is modeled using the power law distribution explained in Section 3.5. To avoid partitioning, each node is made to contact with at least one node. The maximum number of nodes to contact with is drawn from a uniform distribution of up to one fifth of the total number of nodes in the network. Packets are created at every node using Poisson distribution, and assigned a random destination. When generated, each packet is associated with a unique identifier, time of creation and a time to live (TTL). Table 6.1 shows the values used for the network parameters.

Each node records all the packets it receives to avoid receiving two copies of the same packet. We study the impact of varying the buffer capacity, the traffic load, packet time-to-live (TTL) values, and number of nodes in the network on the performance of the routing protocols: Epidemic, PROPHET, Binary SPRAY-AND-WAIT (SnW) and our proposed protocol (EFR-DTN). We consider three metrics to measure the performance of the different protocols, which are Delivery ratio, Delivery cost and Average packet delay. The metrics are explained in Section 3.2.

Procedure 5 EFR-DTN Routing Protocol

```
Procedure Name: OnContact
Input: a,b,ContactDuration
DropExpiredPackets(a,b) /*Drop packets with their lifetime expired in both nodes*/
ExchangeSummaryVector(a,b)
Age = CurrentTime − LastMeetingTime(a, b)
Γ_ab= UpdateDeliveryPredictability(Γ^old_(a,b), Age, γ, α, β)
if ContactDuration > 0 then
  if pkt=GetPacket(a) then
    if NotReceivedBefore(pkt,b) then
      if IsDestination(pkt,b) then
        SendPacket(pkt,a)
        ConsumePacket(pkt,b)
      else
        if ExpiryTime < ExpiryThreshold then
          NrCopies=GetLastChance(pkt,a)
          if NrCopies > 1  then
            StorePacket(pkt,b)
            SetLastChance(pkt,a,NrCopies − 1)
            SetLastChance(pkt,b,1)
          end if
        end if
      else
        DPa=DeliveryPredictability(pkt,a)
        DPb=DeliveryPredictability(pkt,b)
        NrCopies=GetNrOfCopies(pkt,a)
        if DPb > DPa and NrCopies > 1 then
          SendPacket(pkt,a)
          StorePacket(pkt,b)
          SetNrOfCopies(pkt,a,NrCopies/2)
          SetNrOfCopies(pkt,b,NrCopies/2)
          U=UniformRandomNumber
          if U > DPa then
            DropPacket(pkt,a)
          end if
        end if
      end if
      ContactDuration=ContactDuration-size(pkt)
    end if
  end if
end if
```

Table 6.1: Network And Protocols Parameters

Protocol	Parameter	Value
	Simulation time	12 *hours*
ALL	Minimum packet arrival time	1 *minute*
	Minimum inter-contact time	4 *hours*
	Updating factor, α	0.75
PROPHET, EFR-DTN	Transitivity constant, β	0.25
	Aging constant,γ	0.98
SnW, EFR-DTN	Initial number of copies	10
EFR-DTN	Last-Chance number of copies	10

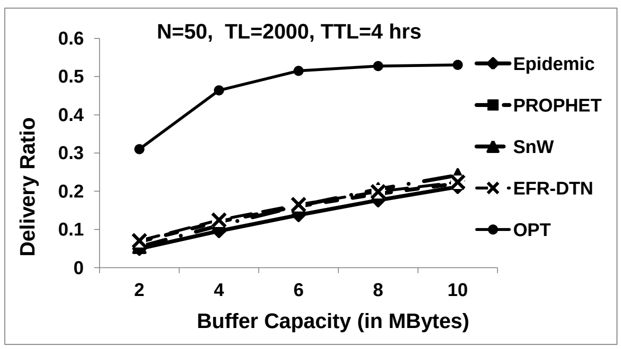


Figure 6.1: Impact of varying buffer capacity on the delivery ratio

6.2.1 Impact of Varying the Buffer Capacity (B)

As shown in Figure 6.1, the delivery ratio of the optimal routing is higher than that of the heuristic protocols. This can be justified by the ability of the optimal routing to find routes that optimally distribute packets among the network and avoid congestion. In addition, the optimal routing spreads only a single copy of the packet which helps in reducing node buffer overflow. Regarding the heuristic protocols, except for Epidemic, they almost have the same or near values of the delivery ratio. Because of the limited buffer capacities, Epidemic cannot compete with the other protocols. Epidemic performance should be better at large buffer capacities and less traffic loads.

Figure 6.2 shows the impact of varying the buffer capacity on the total number of transmissions per delivered packet. The number of transmissions include packets delivered, dropped and those that are still in the buffers. Because the optimal protocol minimizes the

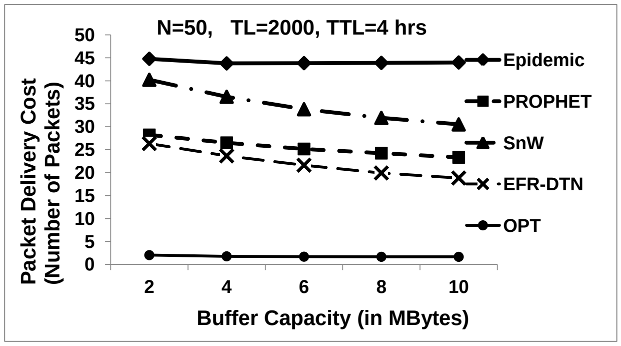


Figure 6.2: Impact of varying buffer capacity on the delivery cost

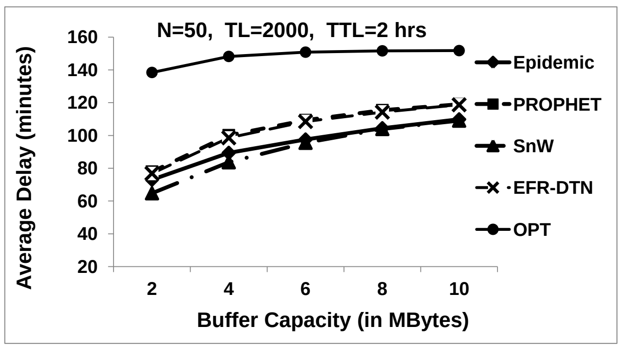


Figure 6.3: Impact of varying buffer capacity on the average packet delay

number of contacts (hops), and spreads a single copy of each packet, it significantly reduces the number of transmissions. Among the heuristic protocols, EFR-DTN and PROPHET extremely outperforms the others because of their intelligent routing. Our proposed protocol, EFR-DTN, proves its lowest number of transmissions among all the heuristic protocols.

Guided routing protocols usually increase the average packet delay because they require the packets to wait till the best next hop. Blind routing protocols do not have a node selection criteria, which means that packet copies move to the first next hop. Therefore, average packet delay of both Epidemic and SnW is lower than that of PROPHET and EFR-DTN, as shown in Figure 6.3.

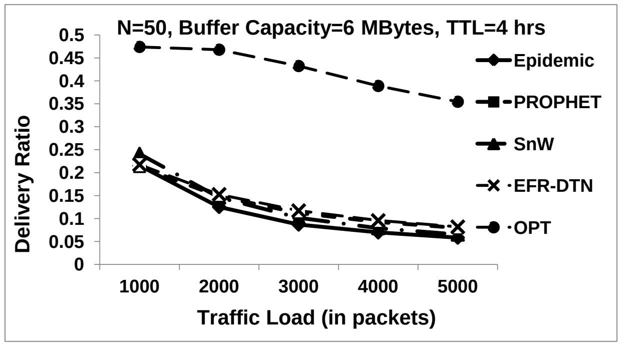


Figure 6.4: Impact of varying traffic load on the delivery ratio

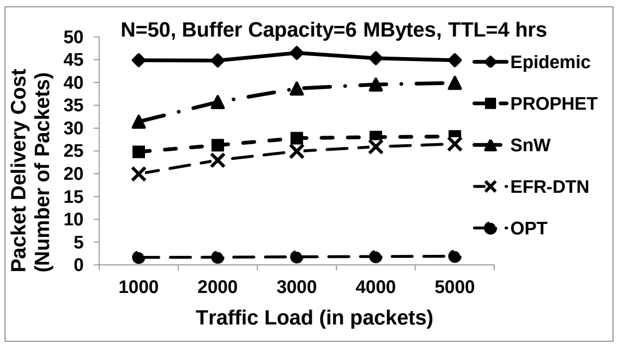


Figure 6.5: Impact of varying traffic load on the delivery cost

6.2.2 Impact of Varying the Traffic Load (TL)

As shown in Figure 6.4, injecting more packets into the network causes the dropping of many of stored packets, and therefore, decreases the delivery ratio for all the heuristic protocols. The optimal protocol is capable of handling the difficult situation even at higher traffic loads. Our proposed protocol, EFR-DTN, have the same behavior as the other protocols, while maintaining its outperformance in the delivery ratio.

Number of transmissions, Figure 6.5, are almost constant with increasing traffic loads for all the protocols. This behavior can be explained by the continuous fullness of the buffers because of the high traffic loads and the low buffer capacities. The buffers are full most of the time and the number of transmissions depend on the packets stored in the queues, not those injected into the network. Therefore, it is almost constant with all the introduced traffic rates. It is worth noting that EFR-DTN maintains the lowest delivery cost with all the traffic loads.

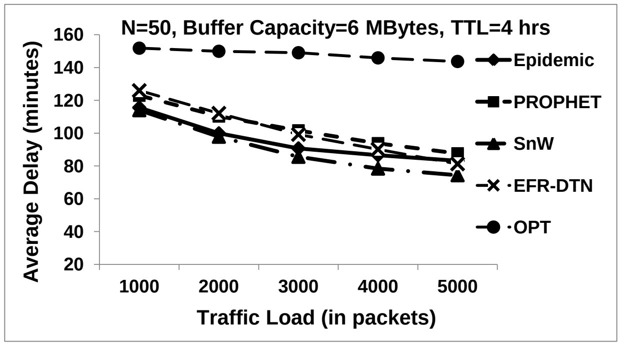


Figure 6.6: Impact of varying traffic load on the average packet delay

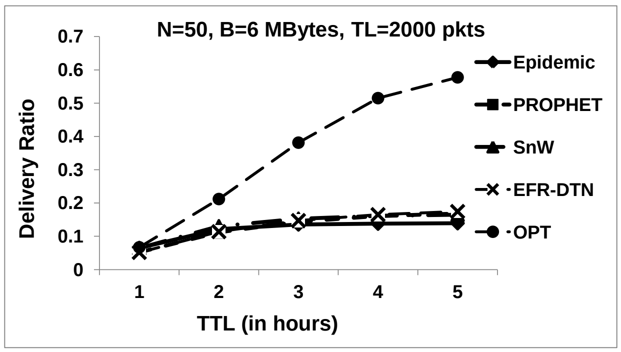


Figure 6.7: Impact of varying packet lifetime on the delivery ratio

The behavior of the average delay with respect to traffic loads is the same as that with buffer capacities. Guided routing protocols have lower delay than blind routing protocols, as shown in Figure 6.6. The optimal routing OPT is the worst in delay, because its optimality is in number of hops.

6.2.3 Impact of Varying the Packets Time-To-Live (TTL)

It is observed from Figure 6.7 that the delivery ratio increases with increasing the packets TTL. This is because increasing the packet lifetime increases the packet chance so that one of its copies reach the destination, instead of being dropped earlier. This increase in delivery ratio is small in the heuristic protocols because the impact of dropping packets due to buffer overflow is more significant than giving longer lifetime for the packets.

As explained in Section 6.8, the number of transmissions are almost constant for all the

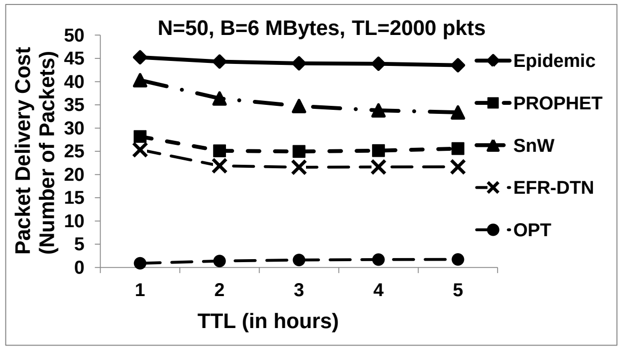


Figure 6.8: Impact of varying packet lifetime on the delivery cost

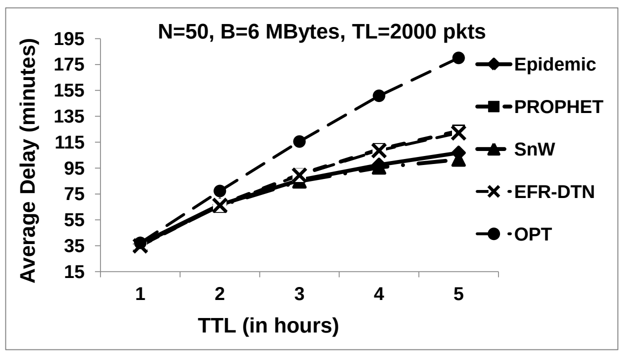


Figure 6.9: Impact of varying packet lifetime on the average packet delay

protocols. Our proposed protocol, EFR-DTN, maintains the lowest energy delivery cost among all the heuristic protocols.

Figure 6.9 shows the same behavior of average packet delay. The delay increases with increasing the TTL, because more packets succeed to reach the destination after being provided longer lifetime.

6.2.4 Impact of Varying the Number of Nodes (N)

Increasing the number of nodes increases their contact frequency and the total network storage space. This increases both the delivery ratio, Figure 6.10, and the delivery cost, Figure 6.11. EFR-DTN and PROPHET have the highest delivery ratio among heuristic protocols, and the lowest delivery cost. Epidemic consumes huge energy and storage resources with increasing the number of nodes.

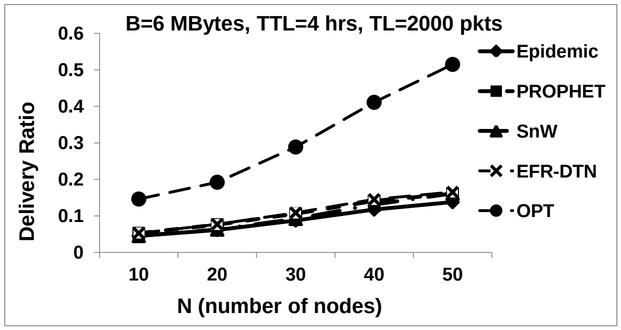


Figure 6.10: Impact of varying the number of nodes on the delivery ratio

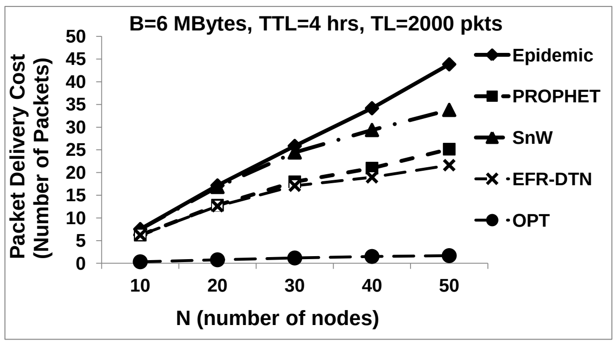


Figure 6.11: Impact of varying the number of nodes on the delivery cost

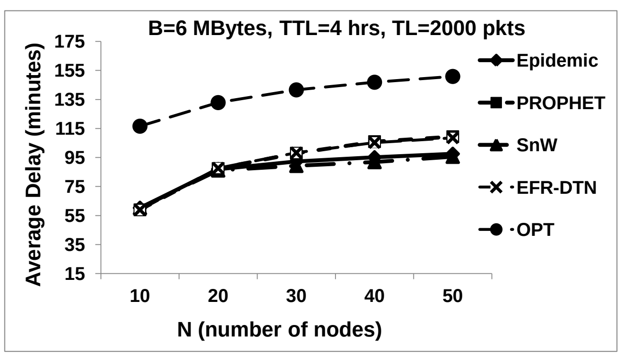


Figure 6.12: Impact of varying the number of nodes on the average packet delay

The average delay increases, as shown in Figure 6.12, because of the increased number of nodes which increases the average number of hops to destination.

6.3 Summary & Conclusion

To increase delivery ratio of data packets in DTN, full flooding blind routing protocols, such as Epidemic, tend to spread multiple copies of the same packet to increase the probability of one of them reaches the destination. The more the number of copies, the higher the storage and energy consumption. Limited flooding protocols, such as SnW, have better resource consumption because of their reduced replication of data packets. However, they are still blinded in terms of node selection. Guided routing protocols, such as PROPHET, collect information from the network to predict the route to destination. Therefore, they can provide better delivery ratio with a middle performance of resource consumption. Given that analysis, we sought to develop a routing protocol that combines the advantages of both guided and blind routing protocols. Our objective was to achieve same or better delivery ratio as that provided by guided routing while minimizing the number of packet copies to reduce resource consumption. We designed and implemented an Eco-Friendly Routing protocol, EFR-DTN, that combines the advantages of guided and blind routing protocols to achieve an acceptable delivery ratio of packets without consuming huge energy and storage resources. Simulation results show that the proposed protocol reduced number of transmissions compared to the routing protocols while maintaining higher delivery ratio and middle average packet delay.

Chapter 7

Social Groups Based Routing For Delay-Tolerant Networks (SGBR)

In this chapter, we propose a new protocol based on social grouping among the network nodes to maximize data delivery ratio while minimizing network overhead by efficiently spreading the packet copies in the network. Our objective is to develop a routing protocol that spreads a small number of packet copies to reduce network overhead, while guiding the packet copies using only local information to reach the destination. To achieve that goal, we exploit the social grouping characteristic of DTN nodes. We consider two nodes to belong to the same social group if they contact each other frequently compared to their contacts with other nodes. An example of such grouping would be the taxi companies in a big city. Taxis that belong to the same company contact each other more frequently, because they either meet in the city roads or in the company garage during their break times. Taxis that belong to different companies only meet each other by coincidence in the city roads. Therefore, taxis that belong to the same company are considered to have strong social connection among each other and weak social connections with the taxis of other companies.

Previous DTN routing protocols focused on using inclusive social metrics, which predicts the path from source to destination by including nodes with strong social connections. The disadvantages of this approach is the need to collect network wide information to better predict the path to destination. We claim that this work is the first to propose an exclusive social metric, which sprays message copies by excluding nodes that are not expected to contribute significantly to the delivery of the message. Using exclusive metrics reduces the need to collect network wide information, while improving the performance metrics. This work has been published in [4] and [8].

7.1 Protocol Description

The proposed protocol aims to utilize the grouping property of nodes. In a typical city, people who live in a neighborhood can have their computers and smartphones connected by constituting a mesh network during evening and night. In the morning, many of them go to their work taking common or different paths. At work, people get connected to their office mates and it may extend to the whole building according to signal strength and obstacles. This process is repeated almost daily for the majority of people. The connectivity between any two random nodes (e.g smartphones or laptops) in this network depends on the social relation between the two people carrying or moving with the devices. If two nodes are in the same neighborhood area for evening and night or in the same office morning and afternoon and are connected frequently during these periods, they are considered to have strong social connection. Otherwise, if they are occasionally connected, then they are considered to have weak social connection.

We consider nodes, frequently meeting each other, to belong to the same social group where they are expected to meet each other again frequently. They are also expected to have around the same social relation with other nodes. In that sense, each node may consider itself a representative of the group to distribute its packets to other groups. Therefore, a node that has a packet destined to other nodes outside its group tends to forward the packet copies to other groups. From our protocol perspective, it is useless to keep several copies of the same packet inside one social group.

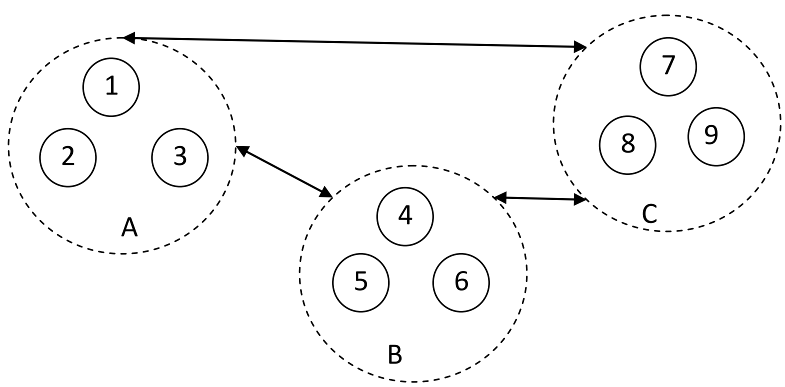


Figure 7.1: A network that consists of three social groups

To illustrate the protocol idea, assume that we have a small network of nine nodes constituting three groups, as shown in Figure 7.1. Node mobility is random direction, and the inter-contact rate between two nodes is exponentially distributed with average λ [73].

Table 7.1: SGBR Parameters

Symbol	Description
$\lambda_{i,j}$	Average rate of contact between node i and node j
RT_m	Remaining time till lifetime of message m expires
Γ_{ab}	Degree of connectivity between nodes a and b
α	The updating factor
γ	The aging constant
k	number of time units elapsed since the last contact
Cth	Connectivity threshold
Dth	Dropping threshold

A node i in group $G(i), G(i) \in \{A, B, C\}$, contacts with another node j in the same group, $G(j) = G(i)$, at a rate $\lambda_{i,j}$, and contacts with a node k in a different group $G(k), G(k) \in \{A, B, C\}, G(k) \neq G(i)$ with rate $\lambda_{i,k}$, where $\lambda_{i,j} >> \lambda_{i,k}$. Let us consider a message m that is generated at node i with lifetime (time-to-live TTL) L_m, and destined to node d in another group. Copies of message m are propagated to a set of nodes N_m. The remaining time until the lifetime of message m, and its copies, expires is RT_m. Assuming that there is enough buffer space in each node, the dropping criteria is only because TTL expires. The probability that message m will not be delivered because of lifetime expiry is then the probability that all the nodes that have copies of m will not meet the destination node d before RT_m expires. Then the probability that any copy of message m will be delivered before its lifetime expires, given that none of the copies is delivered yet, is expressed by:

$$\begin{aligned} &P\{\text{Message } m \text{ will be delivered}\} = \\ &1 - \prod_{n \in N_m} e^{-\lambda_{n,d} RT_m} \end{aligned} \tag{7.1}$$

From Equation 7.1, it can be concluded that the higher the λ values, the higher will be the probability to deliver the message, especially if we are restricted to small number of copies for each message. As an example, assume that message m is generated at node "1" in group A, and destined to node "6" in group B. Node "1" knows that node "6" is not in its group, but does not know which group it belongs to. There are two possible routing decisions:

- Forward a message copy to a node in group A
- Forward a message copy to a node in any group other than group A.

In the former case, after forwarding to another node in group A, say node "2", there will be two copies of the same message in group A ($N_m = \{1, 2\}$) . Then, the probability to deliver the message to node "6" in group B, assuming that the probability of any node in group A to meet any node in group B is the same, will be

$$P\{\text{Message } m \text{ will be delivered} \mid \text{forwarded inside group } A\} = 1 - e^{-(2\lambda_{1,6})RT_m} \tag{7.2}$$

In the latter case the other group may be group B to which the destination "6" belongs or group C. Since node "1" does not know which group "6" belongs to, the probability to transfer a copy to one of the two groups is equal. Let us assume that it will meet either node "4" from group B or node "7" from group C, and it will forward a copy to either one it meets because both are not in its group. In this scenario, the probability to deliver the message becomes

$$\begin{aligned} &P\{\text{Message } m \text{ will be delivered} \mid \text{forwarded to a node in group } B \text{ or group } C\} \\ &= 1 - (0.5e^{-(\lambda_{1,6}+\lambda_{4,6})RT_m} + 0.5e^{-(\lambda_{1,6}+\lambda_{7,6})RT_m}) \end{aligned} \tag{7.3}$$

If we assume that $\lambda_{7,6} \approx \lambda_{1,6}$, and given that $\lambda_{4,6} >> \lambda_{1,6}$, then:

$$\begin{aligned} &(0.5e^{-(\lambda_{1,6}+\lambda_{4,6})RT_m} + 0.5e^{-(\lambda_{1,6}+\lambda_{7,6})T_m}) \approx \\ &(0.5e^{-(\lambda_{4,6})RT_m} + 0.5e^{-(2\lambda_{1,6})RT_m}) << e^{-(2\lambda_{1,6})RT_m} \end{aligned}$$

Therefore the probability to deliver a message, if it is replicated to a different group, is much higher than if it is replicated within the same group.

7.2 Protocol Design

To implement the protocol, a node should know the nodes that belong to its group (having strong connection) and those that are not. To measure how strong is the connection between two nodes, we use the degree of connectivity, Γ_{ab}, which is strengthened by frequent meetings between nodes a and b, and weakened by the time elapsed since the last meeting. When two nodes meet, they update their degree of connectivity, Γ_{ab}, using the following equation:

$$\Gamma_{ab} = (\Gamma_{a,b})_{old}\gamma^k + (1 - (\Gamma_{a,b})_{old}\gamma^k)\alpha \tag{7.4}$$

where:

- Γ_{ab} is the degree of connectivity between nodes a and b,
- $(\Gamma_{a,b})_{old}$ is the degree of connectivity before executing the equation,
- α, $\alpha \in (0, 1]$, is the updating factor,
- γ, $\gamma \in [0, 1]$, is the aging constant, and
- k is the number of time units that have elapsed since the last time nodes a and b have met.

Initially, we set $\Gamma_{ab} = 0$. As time proceeds, and because of nodes mobility, contacts between different node pairs occur. Upon each contact, the two nodes involved in that contact update their degree of connectivity. Each node maintains a table of degrees of connectivity to the other nodes it makes a contact with. After some time, depending on the map size and the speed of the nodes, a node will be able to classify a node it contacts whether it belongs to its group or not. This classification may change with time if the two nodes have reduced or increased their contact frequency.

Packets are sorted by their traversed hop count so that the packet with the minimum hop count is the first to be forwarded, and the packet with the maximum hop count is the first to be dropped. This mechanism encourages newly generated packets to spread in the network, and assumes that packets with large hop count are less probable to reach the destination. Further details of the protocol are explained in the following steps:

- Each packet generated is assigned a unique ID that is associated with it and all its copies till they are dropped or they reach their destinations. The list of all the packet IDs in a node's buffer is called the *Summary Vector*.
- Each node a has a degree of connectivity Γ_{ab} to every other node b that is strengthened by their frequent meetings, using equation 7.4. Based on the degree of connectivity, the two nodes decide to forward or not to forward their packets, except those that are destined to the other node as they should be delivered to the other node regardless of their degree of connectivity.
- Before two contacting nodes start transferring data packets, they exchange their *Summary Vectors*. Packets that are destined to the other node are put on the head of the transmission queue. Other packets that are not destined to the other node and is not in its buffer are sorted based on their traversed hop counts, so that packets with the minimum hop count will be transferred first.

- Packets that are not destined to the other node are transferred only if the degree of connectivity is less than Connectivity Threshold Cth, which indicates that the two nodes do not belong to the same group. In addition, each packet has a limited number of copies to be spread using the Binary SPRAY-and-WAIT (SnW) mechanism. Packet transfer continues for the contact duration.

- After a packet is transferred, it may be dropped from the sender node if the degree of connectivity is greater than the dropping threshold Dth, which ensures that the receiving node is far from being in the same group.

- If the buffer of the receiving node is full, the packet with the largest hop count is dropped to create a space for the forwarded packet to be stored.

The pseudo-code for the proposed protocol is presented in Procedure 6.

7.3 Performance Metrics and Simulation Setup

To test the proposed protocol and compare its performance with the optimal routing and other heuristic protocols, we built a DTN simulator in MATLAB. The simulator takes as inputs the starting times and durations of the nodes contacts. The optimal routing problem is solved using an open source mixed integer linear programming package LPSOLVE [1]. We conducted two sets of experiments to compare the routing protocols:

- The first set compares the routing protocols using a model for the node contacts driven from real trace data [16].

- The second set uses real life movement scenarios of pedestrians and vehicles in a city. These inputs are generated and recorded using the ONE simulator [44].

We consider three metrics to measure the performance of the different protocols, which are Delivery ratio, Delivery cost and Average packet delay. The metrics are explained in Section 3.2.

We implemented a first-in-first-out (FIFO) queuing system in Epidemic,SnW, and PROPHET, as they did not specify their queuing policy in their work. Each point in the results figures is the average of ten repetitive experiment results with a degree of confidence 0.95, using the confidence method explained in Section 3.3. Table 7.2 shows the values used for the parameters of the protocols in all the experiments. The individual setup of the two sets of experiments is detailed in the following subsections.

Procedure 6 Social Groups Based Routing Protocol

```
Procedure Name: OnContact
Input: a,b,ContactDuration
DropExpiredPackets(a,b) /*Drop packets with their lifetime expired in both nodes*/
ExchangeSummaryVector(a,b)
Age = CurrentTime − LastMeetingTime(a, b)
Γ_ab = (Γ_ab)_old γ^Age + (1 − (Γ_ab)_old γ^Age)α
if ContactDuration > 0 then
   /* Sort the packets so that those with minimum hops come first */
   SortPackets(a)
   if pkt=GetPacket(a) then
      if NotReceivedBefore(pkt,b) then
         if IsDestination(pkt,b) then
            SendPacket(pkt,a)
            ConsumePacket(pkt,b)
         else
            /* Forward a copy under protocol conditions*/
            NrCopies=GetNrOfCopies(pkt,a)
            if Γ_ab < Cth and NrCopies > 1 then
               SendPacket(pkt,a)
               StorePacket(pkt,b)
               SetNrOfCopies(pkt,a,NrCopies/2)
               SetNrOfCopies(pkt,b,NrCopies/2)
               if Γ_ab > Dth then
                  DropPacket(pkt,a)
               end if
            end if
         end if
         ContactDuration=ContactDuration-size(pkt)
      end if
   end if
end if
```

Here $\Gamma_{ab} = (\Gamma_{ab})_{old}\gamma^{Age} + (1 - (\Gamma_{ab})_{old}\gamma^{Age})\alpha$.

Table 7.2: All Experiments Protocols Parameters

Protocol	Parameter	Value
	Initialization constant	0.75
PROPHET [50]	Transitivity constant	0.25
	Aging constant,γ	0.98
SnW [74], SGBR	Initial number of copies	5
	Updating factor	0.45
	Aging constant,γ	0.98
SGBR	Connectivity Threshold	0.5
	Dropping Threshold	0.5

Table 7.3: Model-based Network Parameters

Parameter	Value
#Hosts	30
Buffer capacity	2-10 MBytes
Packet TTL	2-10 hours
Packet size	500 KBytes
Packet Interarrival time	250-350 *seconds*
Transmission speed	5 $MBps$
Minimum packet arrival time	$4minutes$
Minimum inter-contact time	3 *hours*
Simulation time	12 *hours*

7.3.1 Model Based Simulations

In this set of experiments, we compare the routing protocols using a model for the node contacts driven from real trace data [16]. The model is explained in Section 3.5. The maximum number of nodes to contact with is drawn from a uniform distribution of up to one fifth of the total number of nodes in the network. Packets are created at every node using Poisson distribution, and assigned a random destination. When generated, each packet is associated with a unique identifier, time of creation and a time to live TTL. Table 7.3 shows the values used for the network parameters.

Each node records all the packets it receives to avoid receiving two copies of the same packet. We study the impact of varying the buffer capacity (B), and the packets TTL on the performance metrics of the protocols: Epidemic [78], PROPHET [50], Binary SPRAY-AND-WAIT (SnW) [74], MAXPROP [12], the proposed protocol SGBR, and the optimal routing OPT.

7.3.2 Real Movement Simulations

We used the ONE simulator [44] to generate the movement scenario. The network consists of vehicles and pedestrians moving around a city. We set the mobility model as map-based movement, where vehicles and pedestrians are restricted to move in predefined paths and routes derived from real map data. We used the map data of the Helsinki downtown area (roads and pedestrian walkways) provided with the simulator. Nodes choose a random point on the map and then follow the shortest route to that point from their current location. More details about using the ONE simulator can be found in Section 3.6

We conducted two experiments with different number of nodes. The first experiment uses a small number of nodes and less number of messages created so as to reduce the complexity of the calculations when implementing the optimal routing protocol. The second experiment expands the network to large number of nodes and packets without implementing the optimal protocol because of the long processing time.

Small Network Simulations

Figure 3.3, in Section 3.6, shows the Helsinki map with ten nodes (5 pedestrians and 5 vehicles) roaming in the city. Tables 7.4 and 7.2 show the network and protocols parameters. We study the impact of changing the buffer capacity with a large TTL value. After that we study the impact of changing the packets TTL with a large buffer capacity.

Large Network Simulations

In this set of experiments, we do not include the optimal protocol in the simulations because the large number of nodes, contacts, and messages require long processing times. We simulate the non-optimal protocols using the Helsinki map with 50 nodes. Tables 7.4 and 7.2 show the network and protocols parameters used in the experiments.

Table 7.4: Real-life Network Parameters(P for Pedestrians and V for Vehicles)

Parameter	Small Network		Large Network	
	P	V	P	V
#Hosts	5	5	30	20
Speed (m/s)	0.5-1.5	2.7-13.9	0.5-1.5	2.7-13.9
Buffer capacity	2-10 MBytes		5-25 MBytes	
Packet Interarrival time	250-350 seconds		25-35 seconds	
Movement	ShortestPath MapBased Movement			
Packet TTL	2-10 Hours			
Transmission speed	5 MBps			
Simulation time	12 Hours			

7.4 Simulation Results

We used a mixed integer linear programming package, LPsolve [1], to solve the optimal formulation. Table 7.5 shows the summary of the results obtained from the different experiments. We presented the minimum and maximum values of the four performance metrics in the two scenarios. Delivery cost is measured in number of packets, and average delay is measured in minutes. Protocols are ordered from the highest to lowest in delivery ratio, and from lowest to highest in other metrics. Minimum and maximum are approximate values taken over all the buffer and TTL values in the same scenario. The summary table is followed by the detailed experiments.

7.4.1 Model Based Simulations

We conduct two sets of experiments to study the impact of changing buffer capacity and packets TTL on the performance of the DTN routing protocols. The two sets are detailed in the following.

Impact of Varying the Buffer Capacity (B)

From Figure 7.2, we observe that increasing the buffer capacity increases the delivery ratio for all the protocols, as long as the amount of Mbytes of the propagating packets are more than the buffer space. In the optimal protocol, the delivery ratio saturates at a small

Table 7.5: Summary of Performance Comparison of DTN Routing Protocols: EPIDEMIC, PROPHET, SnW, MAXPROP, SGBR and OPT

	Model Based		**Real Movement**			
	N=30		N=10		N=50 (without OPT)	
Delivery Ratio	OPT	0.3-0.75	OPT	0.75-0.9		
	SGBR	0.25-0.7	SGBR	0.6-0.9	SGBR	0.5-0.9
	MAXPROP	0.25-0.7	MAXPROP	0.6-0.9	MAXPROP	0.4-0.8
	SnW	0.2-0.55	SnW	0.45-0.85	SnW	0.3-0.7
	PROPHET	0.15-0.45	PROPHET	0.45-0.85	PROPHET	0.25-0.7
	EPIDEMIC	0.15-0.4	EPIDEMIC	0.35-0.7	EPIDEMIC	0.2-0.6
	Numbers shown are the min and max values over all buffer and TTL values					
Delivery Cost	OPT	0.9-1.1	OPT	0.2-0.4		
	SGBR	7-15	SGBR	3	SGBR	5
	SnW	7-15	SnW	3-5	SnW	7-10
	PROPHET	14-20	PROPHET	3-3.5	PROPHET	20
	MAXPROP	20-30	MAXPROP	5	MAXPROP	30
	EPIDEMIC	24-30	EPIDEMIC	6.5	EPIDEMIC	40
	Numbers shown are the minimum and maximum values					
Average Delay	SGBR	60-180	MAXPROP	40-65	MAXPROP	40-100
	MAXPROP	60-180	SGBR	45-70	SGBR	45-110
	EPIDEMIC	70-200	EPIDEMIC	45-75	EPIDEMIC	45-110
	SnW	70-200	SnW	45-75	SnW	50-110
	PROPHET	70-240	PROPHET	45-90	PROPHET	55-110
	OPT	70-240	OPT	55-120		
	Numbers shown are the minimum and maximum values.					

value for the buffer capacity because it propagates only a single copy of each packet. It is also noticed that among the distributed heuristic protocols (non-optimal), MAXPROP and SGBR protocols provide the highest delivery ratio, and Epidemic provides the lowest one.

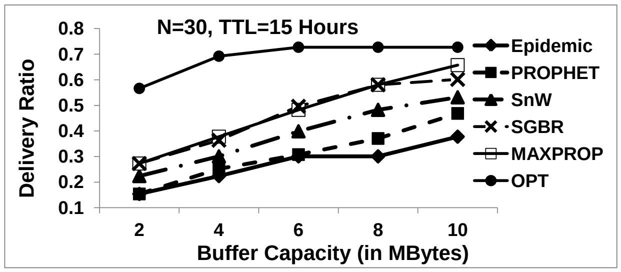


Figure 7.2: Impact of changing buffer capacities on Delivery Ratio.

Increasing the buffer capacity decreases the dropped packets, as shown in Figure 7.3. Once the amount of dropped packets approaches zero, the delivery ratio will saturate.

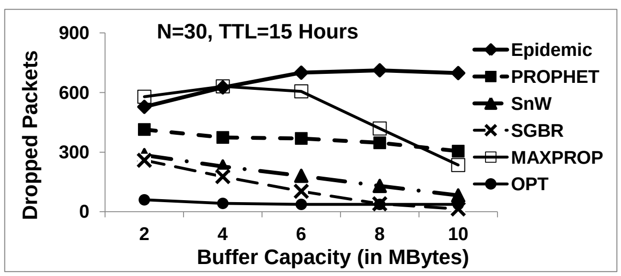


Figure 7.3: Impact of changing buffer capacities on Dropped Packets.

Figure 7.4 shows the cost of delivering one packet. The higher the cost the larger the network overhead and the higher energy consumption during transmissions and receptions. The optimal formulation guarantees a minimum overhead because it only propagates a single copy of each packet. The least cost protocols approaching the optimal results are SGBR and SnW. However, SGBR is favored over SnW because it outperforms SnW in terms of delivery ratio, as shown in Figure 7.2. The highest cost is for EPIDEMIC as expected because of its flooding behavior.

There is a trade-off between maximizing delivery ratio and minimizing packet delays. If the concern is to maximize the delivery ratio, then packets would have to wait in the node buffer till the next contact in its optimal route that guarantees its delivery and other

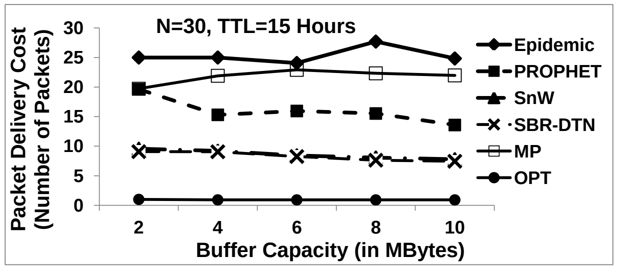


Figure 7.4: Impact of changing buffer capacities on the cost of packet delivery.

packets delivery. Therefore, maximizing delivery ratio introduces higher delays as shown in Figure 7.5. The optimal protocol has the highest delay because its optimality is not in terms of delay. Although the MAXPROP and SGBR provides the highest delivery ratio among the non-optimal protocols, they provide also the lowest delays because of their efficient buffer management mechanisms.

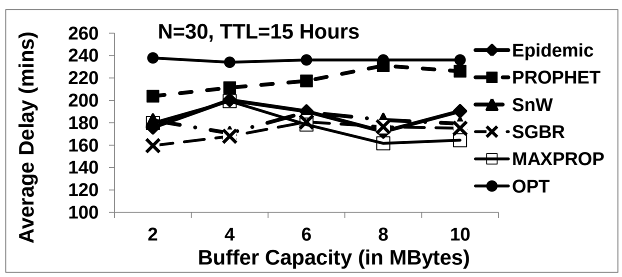


Figure 7.5: Impact of changing buffer capacities on Average Packet Delay.

Impact of Varying the Packet Lifetime (*TTL*)

Figure 7.6 shows that by increasing the *TTL* value of the packets, the delivery ratio increases for all protocols up to an upper bound and then settles. This means that there is no benefit of assigning higher *TTL* values, while it adds a burden to the memory resources of the nodes.

The delivery cost, Figure 7.7, shows that SGBR and SnW have the minimum cost among the non-optimal protocols, because they are the two protocols with bounded replication of packets.

It is intuitive to expect a higher average packet delay while increasing the packets *TTL*

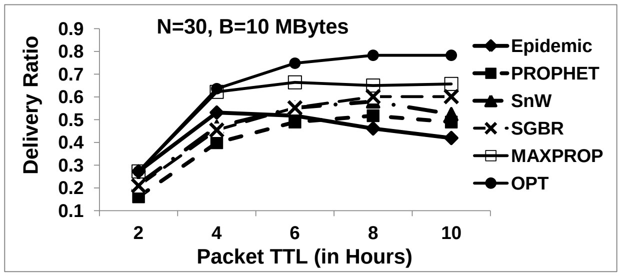


Figure 7.6: Impact of changing packet TTL on Delivery Ratio.

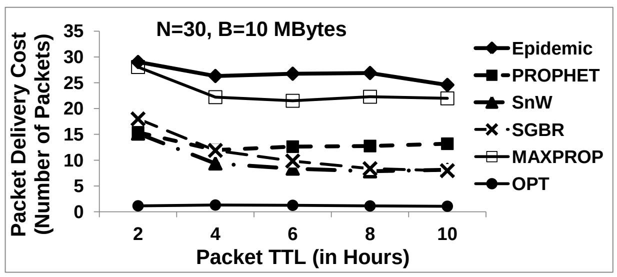


Figure 7.7: Impact of changing packet TTL on cost of packet delivery

(Figure 7.8), because packets that were supposed to be dropped are allowed to live and reach the destination but with higher delay values.

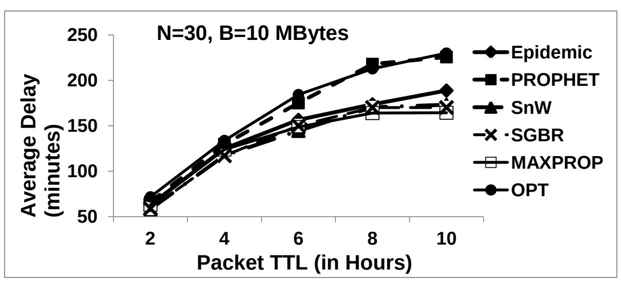


Figure 7.8: Impact of changing packet TTL on Average Packet Delay.

Packets dropped (Figure 7.9) are reduced significantly for EPIDEMIC and MAXPROP, because they are the two most protocols propagating packets. Other protocols have a slight reduction in packets dropped.

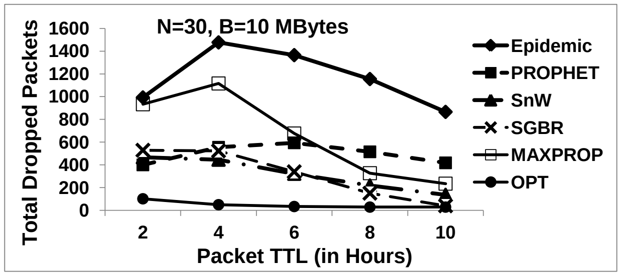


Figure 7.9: Impact of changing packet TTL on Dropped Packets.

7.4.2 Real Movement Based Simulations

We conduct two sets of experiments with different network sizes, small ($N = 10$) and large ($N = 50$), to study the performance of the DTN routing protocols. In the large network size ($N = 50$), only the heuristic protocols are studied, the optimal results are not included because it requires large processing times. The two sets are detailed in the following.

Small Network Simulations, $N = 10$

We conduct two sets of experiments to study the impact of changing buffer capacity and packets TTL on the performance of the DTN routing protocols. The two sets are detailed in the following.

Impact of Varying the Buffer Capacity (B) As shown in Figure 7.10, increasing buffer capacity increases delivery ratio for all the protocols. This is justified by the reduction of dropped packets due to buffer overflow (Figure 7.11). SGBR and MAXPROP are approaching the optimal results, especially at large buffer capacities.

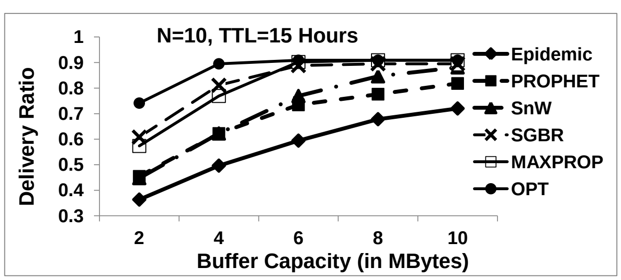


Figure 7.10: Impact of changing buffer capacities on Delivery Ratio in a small network.

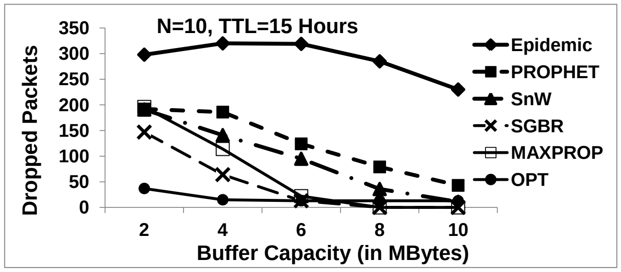


Figure 7.11: Impact of changing buffer capacities on Dropped Packets in a small network.

Increasing the buffer capacity, allows for more packets to be stored, and therefore, more packets to be propagated through the network, as shown in Figure 7.12, and more energy will be consumed. The lowest cost for delivering a packet, in terms of network overhead and energy consumption, is achieved by SGBR and SnW.

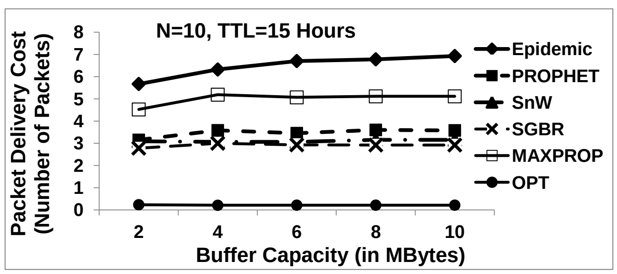


Figure 7.12: Impact of changing buffer capacities on cost of packet delivery

Finally, it is shown in Figure 7.13 that MAXPROP and SGBR provides the lowest average delay for delivered packets because of their efficient buffer management mechanisms.

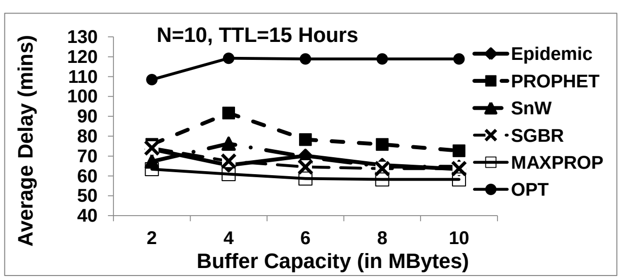


Figure 7.13: Impact of changing buffer capacities on Average Packet Delay in a small network.

Impact of Varying the Packet Lifetime (TTL) Having plenty of buffer space, we study the impact of changing the TTL of the packets. Figure 7.14 shows that giving more lifetime to the packets increases the delivery ratio up to a certain point, then it settles. This result agrees with the results driven from the model-based simulations. Consistent with the previous results, the SGBR and the MAXPROP approach the optimal results more than the other protocols.

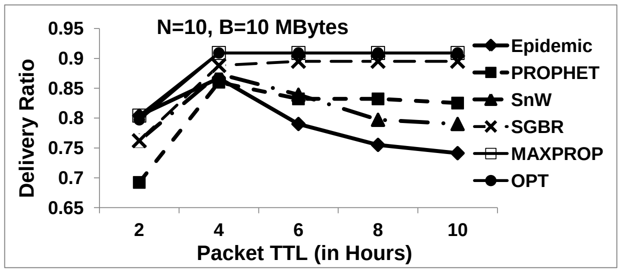


Figure 7.14: Impact of changing packet TTL on Delivery Ratio in a small network.

As shown in Figure 7.15, there is not a significant increase in network overhead while increasing the packets TTL which is because of the large buffer space provided. SGBR still provides the lowest cost of delivering a packet among the non-optimal protocols.

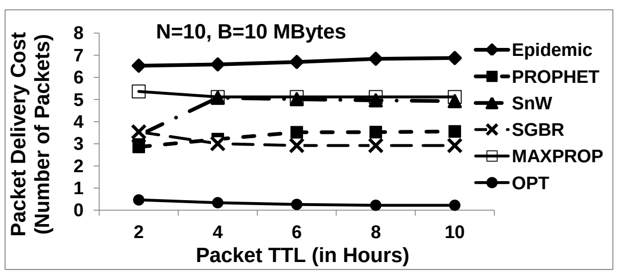


Figure 7.15: Impact of changing packet TTL on cost of delivering a packet.

As shown in Figure 7.16, The delay of the optimal protocol is always the highest because its optimality is in minimizing the number of hops which requires delaying the packets in the buffers to catch an optimal contact.

Large Network Simulations, $N = 50$

We conduct two sets of experiments to study the impact of changing buffer capacity and packets TTL on the performance of the DTN routing protocols. The two sets are detailed

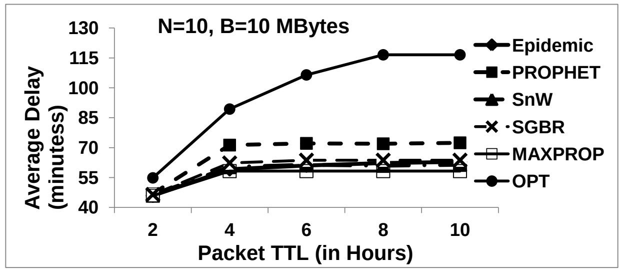


Figure 7.16: Impact of changing packet TTL on Average Packet Delay in a small network.

in the following.

Impact of Varying the Buffer Capacity (B) As shown in Figures 7.17, 7.18, and 7.19, SGBR achieves the highest delivery ratio with the lowest cost for delivering a packet. The good management of replicating and routing packets in the network is represented in the small number of dropped packets, Figure 7.19, where SGBR has the lowest number of packets dropped.

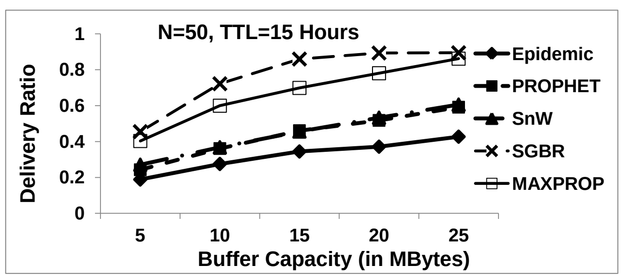


Figure 7.17: Impact of changing buffer capacities on Delivery Ratio in a large network.

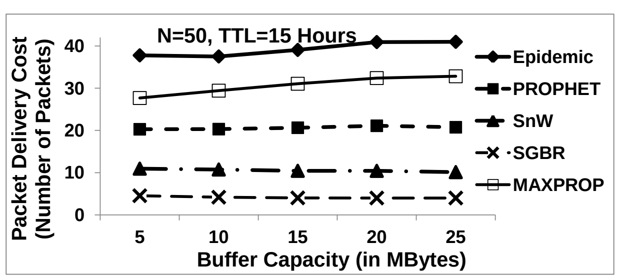


Figure 7.18: Impact of changing buffer capacities on cost of delivering a packet.

MAXPROP achieves the lowest delay among all the protocols. However, this comes with a high cost in network overhead, Figure 7.18, and therefore, energy consumption.

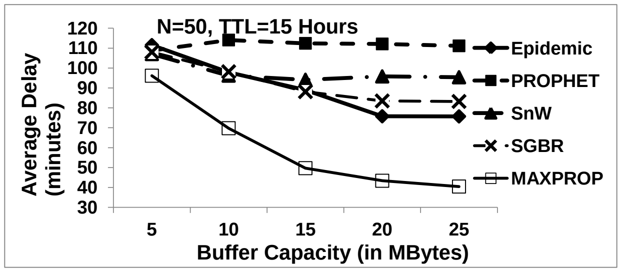


Figure 7.19: Impact of changing buffer capacities on Average Packet Delay in a large network.

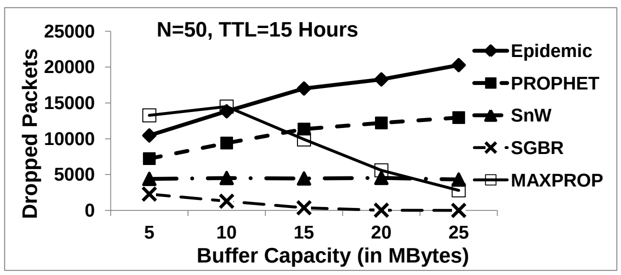


Figure 7.20: Impact of changing buffer capacities on Dropped Packets in a large network.

Impact of Varying the Packet Lifetime (TTL) The same performance could be seen when varying the TTL values, Figures 7.21, and 7.22, having a large buffer space of $B = 20MBytes$. SGBR outperforms the other protocols, except in average delay, Figure 7.23, where MAXPROP has the best performance.

7.5 Summary & Conclusions

Delay Tolerant Networks (DTN), lack end-to-end connections between data sources and destinations. This require the intermediate nodes to store data packets for long periods of time which violates one of the basic assumptions of traditional routing protocols and triggers the development of new ones. In this chapter, we provided a heuristic routing

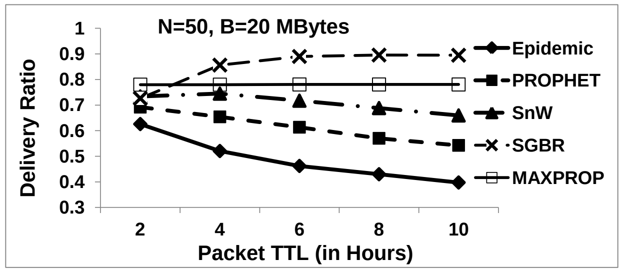


Figure 7.21: Impact of changing packet TTL on Delivery Ratio in a large network.

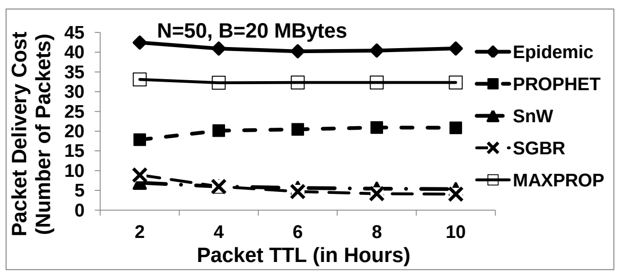


Figure 7.22: Impact of changing packet TTL on cost of delivering a packet.

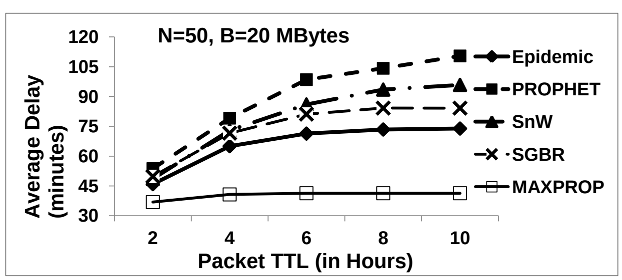


Figure 7.23: Impact of changing packet TTL on Average Packet Delay in a large network.

protocol, SGBR, that utilizes the social relations among the network nodes to reduce redundant copying of packets. We compared SGBR to the optimal results, and four other heuristic protocols: EPIDEMIC, SnW, PROPHET, and MAXPROP. We used traces extracted from a real-life simulator, where vehicles and pedestrians roam in a city according to its given map. Simulation results show that the proposed protocol significantly reduces number of transmissions which reduces the network overhead and the packet delivery cost (up to 25% of the EPIDEMIC protocol), while keeping same or higher delivery ratio (up to double that of the EPIDEMIC protocol, and approaching the optimal results). Table 7.5 provides a summary of all the experiment results. The table shows that our protocol, SGBR, among all the distributed protocols, achieves the best performance in delivery cost and the highest delivery ratio as MAXPROP. In terms of average delay, MAXPROP is the best performance. Because of the limited resources, EPIDEMIC is the worst performance in delivery ratio and cost, but middle performance in average delay. To judge a protocol performance, we should consider all the metrics of interest. In our case, we consider delivery ratio and delivery cost as metrics of interest.

Chapter 8

Supporting Fair Cooperation in DTN

Delay Tolerant Networks (DTNs) comprise nodes with small and limited resources, such as power and storage space. The constraint of resources, together with the mobility and sparsity of DTN nodes, causes an intermittent connection among the nodes and requires delay tolerance of their applications. In such a challenging environment, decentralized routing protocols are implemented with the main concern of maximizing data delivery and minimizing resource usage. These protocols rely on the participation of the network nodes in receiving and storing data packets, collecting and processing information about the network topology to find the next best-hop, and replicating and spreading packets of each other. From a network perspective, all nodes are required to participate in delivering packets of each other. From a node perspective, minimizing resource consumption is a critical requirement. We define fair cooperation as the degree of cooperation where all nodes are satisfied with their participation in the network routing services. We propose a distributed method to calculate a node utility function that will be used to achieve fair cooperation. We implement the method into well-known DTN routing protocols and compare their performance. Results show that by tuning the parameters of the utility function, we could obtain fair cooperation among DTN nodes with improved network performance in terms of delivery ratio and cost.

8.1 Supporting Fair Cooperation in DTNs

A node's ultimate goal in a DTN network is to have its packets served by the other nodes, by storing, replicating and moving them through the network to their destinations, while saving its own energy, by not offering the same services to others. However, this behavior

Table 8.1: Credits and Debits Table of node $i = 1$

Node ID	Credit to $i = 1$	Debit to $i = 1$
2	0	1
3	3	0
4	2	1
5	1	3

is unacceptable, and is being discouraged using trust management systems. Therefore, a node's acceptable goal is to have its packets served, while providing around the same amount of service to packets of the other nodes. This service exchange is recorded by each node in the network in a credits-and-debits table (CDT). Each entry in the CDT of node i records the service provided by (credits) and to (debits) node $j \in \mathrm{N}$, where N is the set of all nodes. Service is measured in number of packets. Table 8.1 shows an example of the CDT of node $i = 1$ in a network of five nodes. The first entry is for node $j = 2$, which shows that node i forwarded one packet for node j, while i did not receive a proof (Acknowledgment) that j forwarded any of i's packets.

When a packet copy reaches its destination, an acknowledgment packet (ACK) is generated. One of the information fields included in the ACK packet is the "route", which contains the list of forwarding nodes of the acknowledged packet. An ACK is generated for the first Cth packet copies reaching the destination, and not just the first one. The ACK is passed on through node contacts till it reaches the source node. When the source node receives the ACK of one of its packets, it updates the credit table at the entries of the forwarding nodes, by incrementing each entry by one, if the acknowledged packet satisfies all the rewarding conditions. We set the rewarding conditions as follows:

1. Only nodes included in the ACK "route" field are rewarded.
2. The packet should not traverse more than H number of hops.
3. Only the first Cth copies reaching destination are rewarded.

The first condition guarantees that rewarding goes only to serious cooperating nodes that will seek to find the best next-hop nodes. The second condition guarantees that the forwarding nodes will do the effort of selecting the nodes that could provide the shortest path to destination. The third condition is to encourage nodes to service the packets, because there are multiple copies of the same packet and the probability to be the first to reach the destination is not high. Other conditions could be added, such as the packet delay or the forwarding node cooperation degree. However, we will suffice with the above conditions for simplicity, because our concern is to introduce the framework for rewarding and not the rewarding metrics themselves.

When there is a contact between two nodes, they exchange a list of all acknowledged packets and stored packets information in each node. A packet information includes its ID, source, destination and size. After receiving the lists, each node decides which of the packets to receive. Then it sends back a list of accept-to-receive packet IDs. By intuition, a node refuses to receive a packet that one of its copies is stored in its buffer, or its delivery to the destination has been reported. Other than that, a node receives packets that it expects to be rewarded by servicing them. Knowing that packets with more than H hops are not rewarded, a node will not receive a packet with more than $H - 2$ hops so far. In addition, each node i calculates a utility function, u_i, using its CDT, as follows:

$$u_i = \sum_{j \in \mathrm{N}, j \neq i} g_{ji} p_{ji} - c_{ij} p_{ij} \tag{8.1}$$

where:

- p_{ji} is the number of packets serviced by node j in favor of node i, which is recorded in j's credit entry in i's CDT,
- g_{ji} is the benefit gained by node i for each packet serviced by node j,
- p_{ij} is the number of packets serviced by node i in favor of node j, which is recorded in j's debit entry in i's CDT, and
- c_{ij} is the cost incurred by node i for each packet serviced in favor of node j,

For simplicity, we will assume that g_{ji} and c_{ij} are independent of j and i. Therefore, we could express g_{ji} and c_{ij}, for any i and j, as g as c. For example, using the values in Table 8.1, and setting $g = 3$ and $c = 1$, node 1 can calculate its utility function as follows:

$$u_1 = 3 \times (0 + 3 + 2 + 1) - 1 \times (1 + 0 + 1 + 3) = 3 \times 6 - 5 = 13$$

The parameters g and c are arbitrary, and can be set as a function of other network parameters, such as energy cost and value of data spread. The ratio g/c represents the willingness of a node to pay for each packet service. In our experiments, and for simplicity, we assume that a node should pay for each hop in the route of a packet generated and spread by that node into the network. If the node requires the number of hops to be no more than three, then it should expect to pay for three hops for each packet, i.e., it should offer service to three other packets.

8.2 Simulation Setup And Results

We built a DTN simulator in MATLAB. The simulator takes as inputs the starting times and durations of node contacts. The optimal routing problem is solved using an open source mixed integer linear programming package LPSOLVE [1]. We conducted three sets of experiments to set the parameters and show the performance of the proposed method:

- The first set studies the impact of changing the ratio of benefit coefficient to service cost, g/c, on the performance metrics. .
- The second set studies the impact of changing the minimum utility threshold, Uth, on the performance metrics.
- The third set studies the impact of changing the maximum number of hops rewarded, H, on the performance metrics.

We consider four metrics to measure the performance of the different protocols, which are:

- Delivery ratio and Delivery cost: These metrics are explained in Section 3.2.
- Network Utility, NU: The summation of all node utilities.
- Unfairness Index, UF: The variance of all node utilities.

We implemented a first-in-first-out (FIFO) queuing system in *Epidemic*,SnW, and $PROPHET$, as they did not specify their queuing policy in their work. Each point in the results figures is the average of ten repetitive experiment results with a degree of confidence 0.95, using the confidence method explained in Section 3.3. Table 8.2 shows the values used for the protocol parameters in all the experiments. Tables 8.3 shows the network parameters. The individual setup of the two sets of experiments is detailed in the following subsections.

8.2.1 Impact of varying the ratio g/c

The two figures, Figures 8.1 and 8.2, show the increase of both the unfairness and network utility with increasing the g/c ratio. As for this scenario, it can be noticed that the rate of increase of the unfairness index increases at $g/c > 3$, while the rate of network utility maintains the same rate. Therefore, for the following experiments, we used the ratio $g/c = 3$. However, this ratio should be checked for different scenarios, such as different number of nodes, their mobility and amount of data generated.

Table 8.2: Supporting Fair Cooperation - Experiment Protocol Parameters

Protocol	Parameter	Value
PROPHET [50]	Initialization constant	0.75
	Transitivity constant	0.25
	Aging constant, γ	0.98
SnW [74], SGBR	Initial number of copies	5
SGBR	Updating factor	0.45
	Aging constant, γ	0.98
	Connectivity Threshold	0.5
	Dropping Threshold	0.5

Table 8.3: Network Parameters

Parameter	Pedestrians	Vehicles
#Hosts	15	5
Speed	0.5-1.5 m/s	2,7-13.9 m/s
Movement	ShortestPath MapBased Movement	
Buffer capacity	10 Mbytes	
Packet TTL	10 Hours	
Average Packet Inter-generation time	600 seconds	
Transmission speed	5 Mbps	
Simulation time	12 Hours	

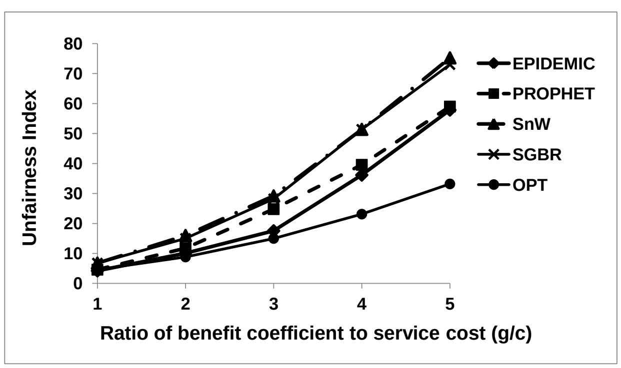


Figure 8.1: Impact of changing g/c ratio on the Unfairness Index.

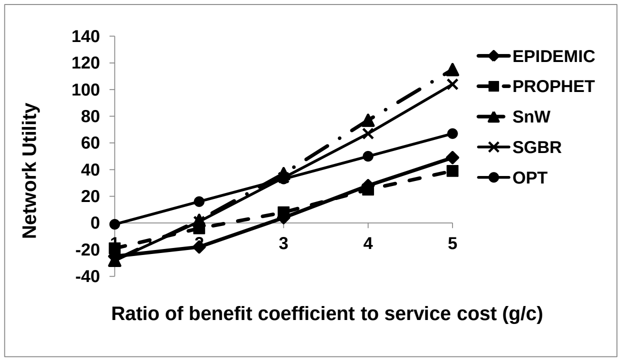


Figure 8.2: Impact of changing g/c ratio on the Network Utility.

8.2.2 Impact of varying the minimum utility threshold Uth

The minimum utility threshold (Uth) represents the generosity of a node to provide forwarding services to the other nodes without receiving corresponding service. If $Uth \approx 0$ (fair cooperation point), then a node provides around the same amount of service as it receives, given that the cost of the service provided is different from that received. Practically, Uth should be a little less than zero, because if $Uth = 0$, nodes will wait for each other to start providing service and the network will be in a service initiator deadlock. In the following set of experiments, we set the benefit coefficient $g = 3$, and maximum number of hops rewarded $H = 5$.

As shown in Figure 8.3, decreasing the utility threshold (increasing the node generosity) increases the unfairness among the network nodes. This is an intuitive result because nodes are going to benefit from the generosity of others without having to pay same service in correspondence. While the impact was significant in EPIDEMIC, it was much less in the other protocols, especially at highly negative Uth values. This is because EPIDEMIC has no constraints on number of copies spread or node selection to help in improving its performance.

As for the EPIDEMIC protocol, approaching the fair cooperation point, increases the network utility, as shown in Figure 8.4. The impact is negligible on the network utility using the other protocols. It is also shown that SnW and SGBR provides the highest network utility among the non-optimal protocols, because of their packet replication constraint and the spreading mechanism.

Increasing the utility threshold (decreasing node's generosity) decreases the delivery

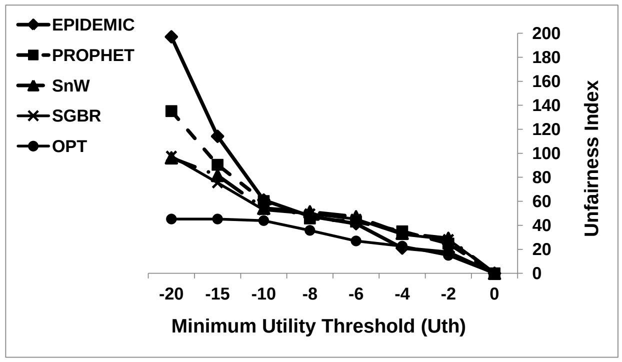


Figure 8.3: Impact of changing minimum utility threshold on the Unfairness Index.

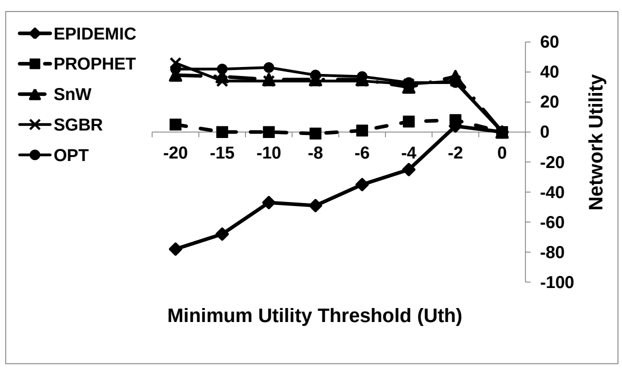


Figure 8.4: Impact of changing minimum utility threshold on the Network Utility.

ratio, as shown in Figure 8.5, because the number of transmissions and receptions decreases, as shown in Figure 8.6. The two figures, Figure 8.5 and 8.6, justify the non-increasing behavior of the protocols utility (excluding EPIDEMIC) in Figure 8.4. The utility is a function of both the delivered packets and those forwarded by nodes. A high delivery ratio may not indicate a node satisfaction, because the node may be providing service more than that received. A node is satisfied if, at least, there is a balance between service provided and received.

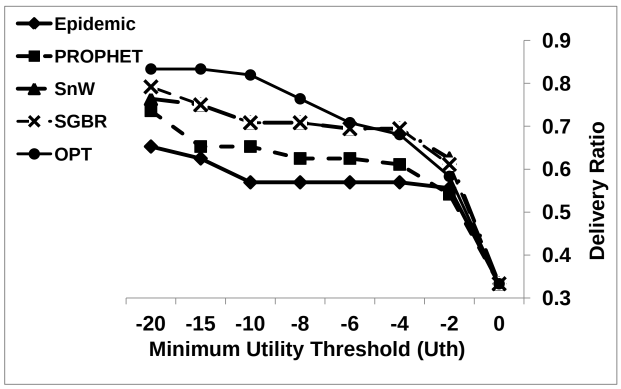


Figure 8.5: Impact of changing minimum utility threshold on the Delivery Ratio.

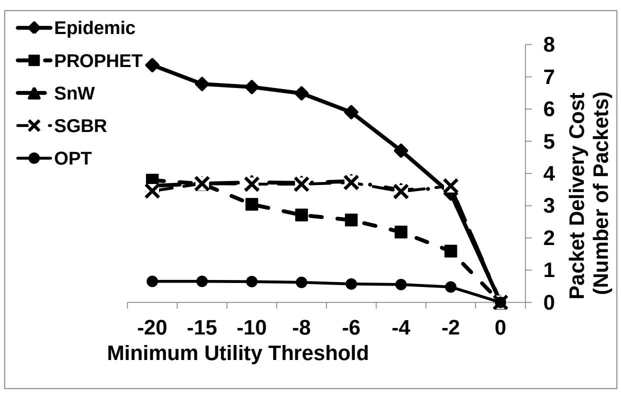


Figure 8.6: Impact of changing minimum utility threshold on the Network Overhead.

8.2.3 Impact of varying maximum number of hops rewarded H

The following set of experiments is conducted to study the impact of changing the maximum number of hops rewarded, H. The first part of these experiments is done with all nodes are willing to cooperate unconditionally ($Uth = -100$), and the other part is conducted with all nodes aim at fair cooperation ($Uth = -2$).

As shown in Figure 8.7, the unfairness increases with increasing H. However, the rate of increase and the values of the unfairness are much higher when $Uth = -100$ (Figure 8.7a). EPIDEMIC has the highest unfairness index because of its unconstrained spreading mechanism. In Figure 8.7b, the unfairness index of the other protocols settles at $H = 2$ or $H = 3$, which is the highest recorded number of hops in this scenario.

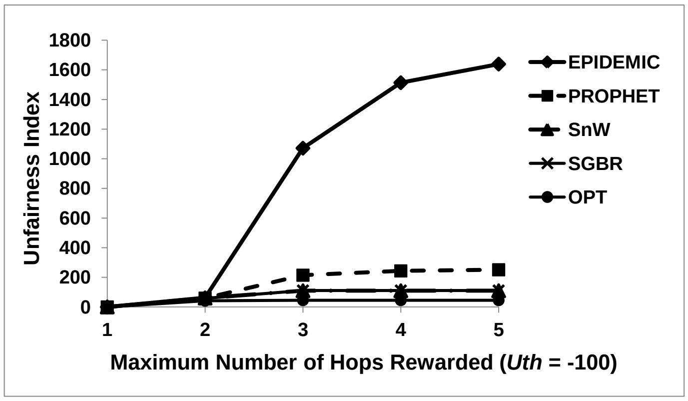
For this scenario, the highest network utility was found at $H = 2$, for all protocols. For $H > 2$, the network utility decreases for EPIDEMIC and PROPHET because they have routes with more than $H = 2$. For the other protocols, it settles at $H = 3$, because there are no routes with more than three hops.

The delivery ratio reaches its maximum value at $H = 2$, and then settles for all protocols in case of $Uth = -100$, as shown in Figure 8.9a. However, when $Uth = -2$, EPIDEMIC and PROPHET experience a slight decrease at $H > 2$, because of the increased request of service to deliver packets with more hops to destination, which is not guaranteed to be found while nodes seek fair cooperation.

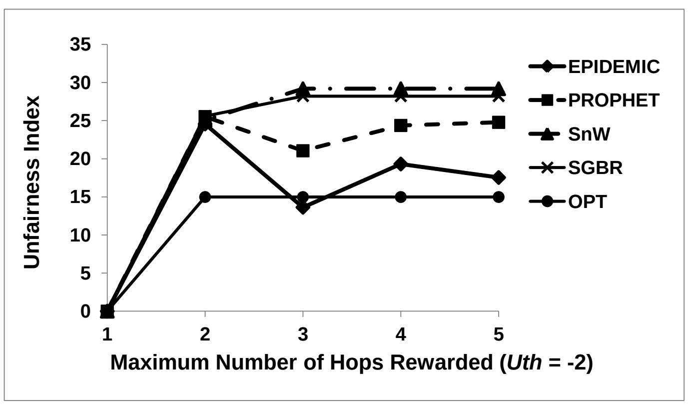
As shown in Figure 8.10a, the delivery cost increases with increasing the allowed number of hops. It settles at around $H = 3$ for all protocols except for EPIDEMIC because of its unlimited flooding behavior. The delivery cost decreases when nodes seek the fair cooperation point $Uth = -2$, as shown in Figure 8.10b. All protocols including EPIDEMIC do not incur significant change at $H > 2$

8.3 Summary & Conclusion

In this chapter, we discussed the willingness of DTN nodes to cooperate in routing services. DTNs differ from their ancestor, the ad-hoc network, in the hop-by-hop routing used, the multiplicity of packet copies spread throughout the network, and the fully decentralized nature of DTN. These differences in characteristics trigger the development of a new cooperation motivation scheme. There are two perspectives in forwarding packets in DTNs; the node perspective and the network (other nodes) perspective. The node perspective is to have its packets serviced by the network while minimizing its consumption of energy. The

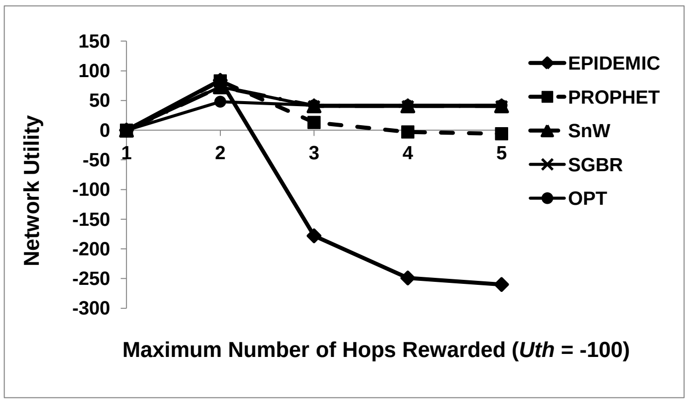


(a) $Uth = -100$

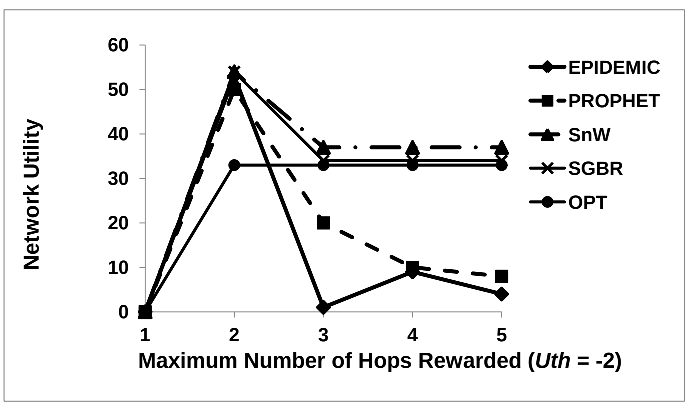


(b) $Uth = 0$

Figure 8.7: Impact of changing maximum number of hops rewarded on the Unfairness Index.

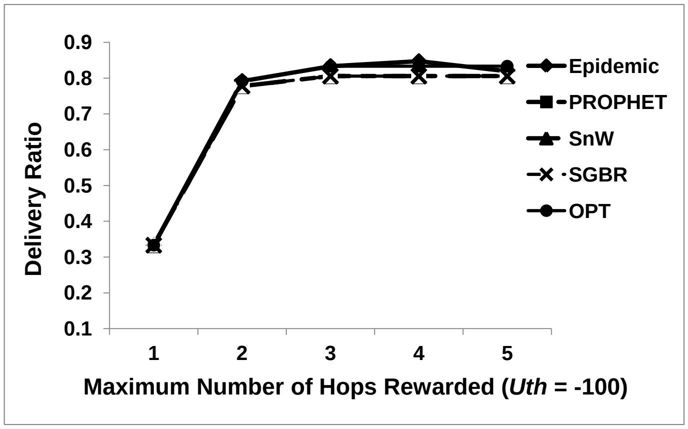


(a) $Uth = -100$

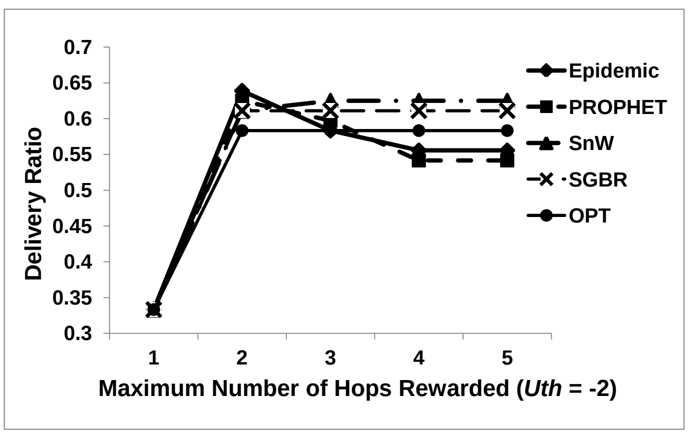


(b) $Uth = 0$

Figure 8.8: Impact of changing maximum number of hops rewarded on the Network Utility.

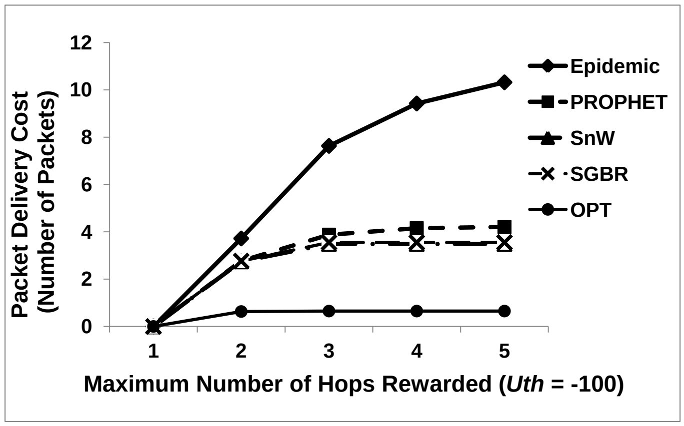


(a) $Uth = -100$

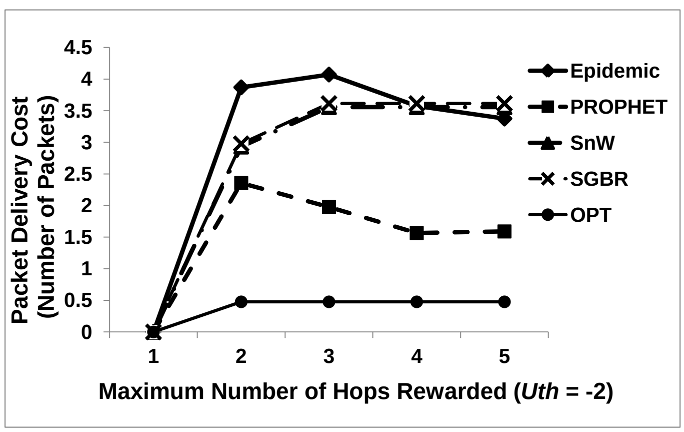


(b) $Uth = 0$

Figure 8.9: Impact of changing maximum number of hops rewarded on the Delivery Ratio.



(a) $Uth = -100$



(b) $Uth = 0$

Figure 8.10: Impact of changing maximum number of hops rewarded on the Network Overhead.

network perspective is to involve all the nodes in the forwarding service to increase delivery. Setting a fair point of cooperation requires each node to offer service around the same amount as that received. We proposed a distributed method to achieve fair cooperation, by recording packets forwarded by each node and those delivered and calculate a utility function based on these values. We implemented the proposed method into well-known protocols: EPIDEMIC, PROPHET and SnW, in addition to our protocol SGBR and the optimal routing. Results show that running the network around the fair cooperation point increases the network utility and decreases unfairness among the nodes.

Chapter 9

Conclusions and Future Work

9.1 Summary and Conclusions

Delay tolerant networks (DTN), also known as intermittently connected networks (ICN), lack end-to-end connections between data sources and destinations. This require intermediate nodes to store data packets for long periods of time which violates one of the basic assumptions of traditional routing protocols and triggers the development of new ones. Routing protocols developed for DTNs adapt themselves to this challenging environment by probabilistically sending multiple copies of data packets so that one of them reaches the destination. Nodes receiving the packets store them until they meet other nodes or meet their destinations. Simple DTN routing protocols blindly send data packets to the nodes they meet without having a selection criterion. They range from the full network flooding to the limited flooding. This approach has its drawbacks such as burdening the buffer and the inefficient use of the contact duration. Other routing protocols tend to restrict forwarding the data packets to selected nodes. Using some information collected about the network, they guide the data packets to their destinations. This approach fails when the network information cannot be collected or the network topology is changing faster than the collection rate. An efficient DTN routing protocol should integrate node selection, packet selection, and buffer management mechanisms to obtain the best performance. Our contributions in this research, can be summarized as follows:

- In Chapter 4, we formulated an optimization problem of single-copy centralized routing in DTNs, assuming the availability of present and future node contacts and buffer information. We implemented the problem with three different objectives: minimum delay, minimum number of hops, and maximum number of delivered messages. We solved the problem using parameters from simulated networks. Results show that minimizing the number of hops achieves higher delivery ratio than minimizing the delay and almost the same as maximizing the number of delivered messages for most of the buffer capacities, traffic loads and TTL values. In addition, minimizing the number of hops proves to significantly reduce the number of transmissions in the network which results in considerable energy saving. The minimum number of hops results are used as a performance benchmark to compare with the distributed non-optimal protocols.

- In Chapter 5, we studied the distributed non-optimal routing protocols developed for DTNs, and conducted a performance comparison among selected well-known protocols (Epidemic, SPRAY-AND-WAIT (SnW), PROPHET and MAXPROP) representing the different types of routing protocols (blind and guided, limited and full flooding). Results show the outperformance of MAXPROP in delivery ratio and average delay, and the outperformance of SnW and PROPHET in delivery cost. An efficient routing protocol should integrate the node selection, the packet selection, and the buffer management mechanisms to obtain the best performance.

- In Chapter 6, we developed our first DTN routing protocol, Eco-Friendly Routing for DTN (EFR-DTN), which combines the strengths of two of the previously proposed protocols to provide better delivery ratio with lower network overhead (less power consumption). The protocol utilizes node encounters to estimate the route to destination, while minimizing the number of packet copies throughout the network. Simulation results show that EFR-DTN reduced the number of transmissions compared to the routing protocols while maintaining higher delivery ratio and middle average packet delay.

- In Chapter 7, we proposed our second distributed protocol, Social Groups Based Routing (SGBR), which uses social relations among network nodes to exclude nodes that do not contribute significantly to the delivery of the packet to its destination. Using exclusive metrics reduces the need to collect network wide information, while improving the performance metrics. Simulation results showed that SGBR protocol achieves lower network overhead while achieving same or better delivery ratio compared to the other routing protocols.

- In Chapter 8, we discussed the incentives of DTN nodes to cooperate in the routing process. We investigated the degree of fair cooperation at which the network nodes are satisfied. We proposed and implemented a distributed credit-based system to keep track of and reward node participation in packet routing, by recording packets forwarded by each node and those delivered and calculate a utility function based on these values. Results show that running the network around the fair cooperation point increases the network utility and improves fairness among the nodes.

9.2 Future Research Work

This research targets improving the routing performance in DTN, by maximizing delivery ratio and minimizing energy consumption. The latter is achieved by decreasing the number of transmissions and receptions which results in minimizing the network overhead. In our work, we achieved less network overhead than the other protocols and maintained same or higher delivery ratio. This work can be extended in several ways, such as:

- **Using soft computing techniques to developing an adaptive routing protocol**: The uncertainty in the information used in the routing selection criteria motivates the usage of soft computing techniques, such as Fuzzy logic, to deal with this information to generate the routing decision. One of the advantages of using soft computing techniques is its adaptability to the dynamic environment. We expect to obtain better performance results using this technique.

- **Analyzing and developing buffer management mechanisms:** According to our analysis, efficient buffer management improves the delivery ratio and reduces the network overhead, by dropping low value packets. Some previous studies handled the buffer management in DTN, such as [12] and [47]. However, they used full flooding as the routing approach. In [51], two of the routing protocols, Epidemic and PROPHET, were tested with several queuing policies. This work can be extended by implementing different queuing policies. Analysis and testing of queuing policies in constrained routing protocols can open several research points to be added to this work.

- **Joint Routing and MAC protocol**: Energy saving is one of the main objectives in DTNs. Routing protocols contribute to energy saving by reducing number of transmissions and receptions. However, a lot of work could be done at the MAC layer to support the efficient utilization of energy such as power control and scheduling sleep and awake periods for DTN nodes.

- **Detecting selfish and malicious nodes**: In our work to support fair cooperation, we did not consider scenarios in which nodes cheat the system by spreading false ACK packets to notify the network of its cooperation, while it is not cooperating. Other security issues and selfish behavior could ruin the system unless considered in the trust management scheme. While security was not our focus in this work, it is an important aspect to consider when building a network.

References

[1] Lpsolve: Open source (mixed-integer) linear programming system.

[2] T. Abdelkader, K. Naik, and A. Nayak. Choosing the objective of optimal routing protocols in delay tolerant networks. In *Computer Engineering Conference (ICENCO), 2010 International*, pages 16 –21, dec. 2010.

[3] T. Abdelkader, K. Naik, and A. Nayak. An eco-friendly routing protocol for delay tolerant networks. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2010 IEEE 6th International Conference on*, pages 450 –457, oct. 2010.

[4] T. Abdelkader, K. Naik, A. Nayak, and N. Goel. A socially-based routing protocol for delay tolerant networks. In *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, pages 1 –5, dec. 2010.

[5] Tamer Abdelkader, Kshirasagar Naik, and Amiya Nayak. An eco-friendly routing protocol for delay tolerant networks. In *The 6th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2010)*, Niagara Falls, Canada, October 2010.

[6] Tamer Abdelkader, Kshirasagar Naik, Amiya Nayak, and Nishith Goel. A socially-based routing protocol for delay tolerant networks. In *2010 IEEE Global Telecommunications Conference (GLOBECOM 2010)*, Miami, Florida, USA, 2010.

[7] Tamer Abdelkader, Kshirasagar Naik, Amiya Nayak, Nishith Goel, and Vineet Srivastava. A performance comparison of delay tolerant network routing protocols. *submitted to IEEE Systems Journal*, 2012.

[8] Tamer Abdelkader, Kshirasagar Naik, Amiya Nayak, Nishith Goel, and Vineet Srivastava. Sgbr: A routing protocol for delay tolerant networks using social grouping. *IEEE Transactions on Parallel and Distributed Systems*, 99(PrePrints):1–1, 2012.

[9] C. Becker and G.Schiele. New mechanisms for routing in ad hoc networks. In *Fourth Plenary Cabernet Workshop*, Pisa, Italy, October 2001.

[10] E. Brewer, M. Demmer, B. Du, M. Ho, M. Kam, S. Nedevschi, J. Pal, R. Patra, S. Surana, S. Surana, K. Fall, and A10. The case for technology in developing regions. *Computer*, 38(6):25–38, May 2005.

[11] E. Bulut, Zijian Wang, and B.K. Szymanski. Impact of social networks on delay tolerant routing. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1 –6, 30 2009-dec. 4 2009.

[12] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *25th IEEE International Conference on Computer Communications. Proceedings INFOCOM 2006.*, pages 1–11, April 2006.

[13] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6):128–136, June 2003.

[14] B. Burns, O. Brock, and B. N. Levine. Mv routing and capacity building in disruption tolerant networks. In *INFOCOM 2005: 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 398–408, Miami, FL, USA, March 2005. IEEE.

[15] Iacopo Carreras, Daniele Miorandi, and Imrich Chlamtac. A simple model of contact patterns in delay-tolerant networks. *Wireless Networks*, 16:851–862, 2010.

[16] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, June 2007.

[17] Dimitris E. Charilas, Konstantinos D. Georgilakis, and Athanasios D. Panagopoulos. Icarus: hybrid incentive mechanism for cooperation stimulation in ad hoc networks. *Ad Hoc Networks*, (0):–, 2012.

[18] Zong Da Chen, H.T. Kung, and Dario Vlah. Ad hoc relay wireless networks over moving vehicles on highways. In *AACM Mobihoc*, 2001.

[19] Thomas Clausen, Philippe Jacquet (editors), Cédric Adjih, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol (OLSR). RFC 3626, pages 1-75, October 2003. Network Working Group.

[20] Vania Conan, Jérémie Leguay, and Timur Friedman. Characterizing pairwise inter-contact patterns in delay tolerant networks. In *Autonomics '07: Proceedings of the 1st international conference on Autonomic computing and communication systems*, 2007.

[21] P. Costa, C. Mascolo, M. Musolesi, and G.P. Picco. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 26(5):748 –760, june 2008.

[22] Elizabeth M. Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '07, pages 32–40, New York, NY, USA, 2007. ACM.

[23] J. A. Davis, A. H. Fagg, and B. N. Levine. Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks. pages 141–148, Oct 2001.

[24] A. Elwhishi and Pin-Han Ho. Sarp - a novel multi-copy routing protocol for intermittently connected mobile networks. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1 –7, 30 2009-dec. 4 2009.

[25] Vijay Erramilli, Mark Crovella, Augustin Chaintreau, and Christophe Diot. Delegation forwarding. In *MobiHoc '08: Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 251–260, 2008.

[26] Kevin Fall. A delay-tolerant network architecture for challenged internets, 2003.

[27] Kevin R. Fall and Stephen Farrell. Dtn: an architectural retrospective. *IEEE Journal on Selected Areas in Communications*, 26(5):828–836, 2008.

[28] R. H. Frenkiel, B. R. Badrinath, J. Borres, and R. D. Yates. The infostations challenge: balancing cost and ubiquity in delivering wireless data. *Personal Communications, IEEE [see also IEEE Wireless Communications]*, 7(2):66–71, Aug 2000.

[29] D. J. Goodman, J. Borras, N. B. Mandayam, and R. D. Yates. Infostations: A new system model for data and messaging services. In *VTC 1997: Vehicular Technology Conference*, volume 2, pages 969–973. IEEE, May 1997.

[30] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad-hoc wireless networks. In *IEEE ACM Transactions on Networking*, pages 477–486, Aug 2002.

[31] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions Network*, 10:477–486, 2002.

[32] Frederic Guidec and Yves Maheo. Opportunistic Content-Based dissemination in disconnected mobile ad hoc networks. In *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM'07)*, pages 49–54. IEEE, November 2007.

[33] Radu Handorean, Christopher D. Gill, and Gruia-Catalin Roman. Accommodating transient connectivity in ad hoc and mobile settings. In *Pervasive*, pages 305–322, Jan 2004.

[34] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 10:1576–1589, 2011.

[35] Raj Jain. *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling.* Wiley, April 1991.

[36] S. Jain, K. Fall, and R. Patra. Routing in a delay tolerant network. *Proc. SIGCOMM*, 34(4):145–158, October 2004.

[37] Sushant Jain, Michael Demmer, Rabin Patra, and Kevin Fall. Using redundancy to cope with failures in a delay tolerant network. *SIGCOMM Comput. Commun. Rev.*, 35(4):109–120, Oct 2005.

[38] Juan José Jaramillo and R. Srikant. Darwin: distributed and adaptive reputation mechanism for wireless ad-hoc networks. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, MobiCom '07, pages 87–98, New York, NY, USA, 2007. ACM.

[39] David B. Johnson. Validation of wireless and mobile network models and simulation. In *In DARPA/NIST Network Simulation Validation Workshop*, 1999.

[40] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

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

[42] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li S. Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *ASPLOS-X: Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, volume 37, pages 96–107, New York, NY, USA, October 2002. ACM Press.

[43] Hyewon Jun, Mostafa H. Ammar, Mark D. Corner, and Ellen W. Zegura. Hierarchical power management in disruption tolerant networks with traffic-aware optimization. In *Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, CHANTS '06, pages 245–252, New York, NY, USA, 2006. ACM.

[44] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA, 2009. ICST.

[45] Ari Kernen, Teemu Krkkinen, and Jrg Ott. Simulating mobility and dtns with the one (invited paper). *Journal of Communications*, 5(2), 2010.

[46] J. Konorski. Effective data-centric reputation systems for manets: A novel evaluation framework. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1 –6, june 2011.

[47] Amir Krifa, Chadi Barakat, and Thrasyvoulos Spyropoulos. Optimal buffer management policies for delay tolerant networks. In *SECON*, pages 260–268, 2008.

[48] Kyunghan Lee, Seongik Hong, Seong J. Kim, Injong Rhee, and Song Chong. Slaw: A mobility model for human walks. In *Proceedings of the 28th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2009.

[49] Qun Li and Daniela Rus. Communication in disconnected ad hoc networks using message relay. *Journal of Parallel Distributed Computing*, 63:75–86, 2003.

[50] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. *ACM Lecture Notes in Computer Science*, pages 239–254, Jan 2004.

[51] Anders Lindgren and Kaustubh S. Phanse. Evaluation of queueing policies and forwarding strategies for routing in intermittently connected networks. In *In Proceedings of the The First International Conference on COMmunication System softWAre and MiddlewaRE (COMSWARE)*, 2006.

[52] Jason Liu, Yougu Yuan, David M. Nicol, Robert S. Gray, Calvin C. Newport, David Kotz, and Luiz Felipe Perrone. Empirical validation of wireless models in simulations of ad hoc routing protocols. *Simulation*, 81:307–323, 2005.

[53] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Xuemin Shen, and B. Preiss. Pi: A practical incentive protocol for delay tolerant networks. *Wireless Communications, IEEE Transactions on*, 9(4):1483 –1493, april 2010.

[54] Christian Maihöfer and Reinhold Eberhardt. Time-stable geocast for ad hoc networks and its application with virtual warning signs. *Computer Communications*, 27(11):1065–1075, 2004.

[55] M. Musolesi, S. Hailes, and C. Mascolo. Adaptive routing for intermittently connected mobile ad hoc networks. In *IEEE WoWMoM*, pages 183–189, June 2005.

[56] Delphine Nain, Noshirwan Petigara, and Hari Balakrishnan. Integrated routing and storage for messaging applications in mobile ad hoc networks. *Mob. Netw. Appl.*, 9(6):595–604, 2004.

[57] Samuel C Nelson, Mehedi Bakht, Robin Kravets, and Albert F. Harris, III. Encounter: based routing in dtns. *SIGMOBILE Mobile Computer Communication Review*, 13(1):56–59, 2009.

[58] Sensor networking with delay tolerance (Sendt). http://down.dsg.cs.tcd.ie/sendt/.

[59] J. Ott and D. Kutscher. Drive-thru internet: Ieee 802.11b for "automobile" users. volume 1, pages 362–373, March 2004.

[60] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, 44(11):134–141, November 2006.

[61] A. Pentland, R. Fletcher, and A. Hasson. Daknet: rethinking connectivity in developing nations. volume 37, pages 78–83, 2004.

[62] C. Perkins, E. Royer, and S. Das. Rfc 3561 ad hoc on-demand distance vector (aodv) routing. Technical report, 2003.

[63] Charles E. Perkins. *Ad Hoc Networking*. Addison-Wesley Professional, December 2000.

[64] Wizzy project. http://www.wizzy.org.za/.

[65] M.T. Refaei, Vivek Srivastava, L. DaSilva, and M. Eltoweissy. A reputation-based mechanism for isolating selfish nodes in ad hoc networks. In *Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on*, pages 3 – 11, july 2005.

[66] K. Scott and S. Burleigh. Bundle protocol specification, Nov 2007.

[67] R. C. Shah, S. Roy, S. Jain, and W. Brunette. Data mules: Modeling a three-tier architecture for sparse sensor networks. In *IPSN 2008: Proceedings of the First IEEE Workshop on Sensor Network Protocols and Applications*, pages 30–41. IEEE, 2003.

[68] C. Shen, G. Borkar, S. Rajagopalan, and C. Jaikaeo. Interrogation-based relay routing for ad hoc satellite networks. In *IEEE Globecom*, November 2002.

[69] T. Small and Z. J. Haas. The shared wireless infostation model: a new ad hoc networking paradigm (or where there is a whale, there is a way). In *ACM MobiHoc*, pages 233–244, June 2003.

[70] Tara Small and Zygmunt J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 260–267. ACM Press, Aug 2005.

[71] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Sensor and Ad Hoc Communications and Networks (SECON)*, pages 235–244, Oct 2004.

[72] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, New York, NY, USA, Aug 2005. ACM Press.

[73] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Performance analysis of mobility-assisted routing. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '06, pages 49–60, New York, NY, USA, 2006. ACM.

[74] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Efficient routing in intermittently connected mobile networks: the multiple-copy case. *IEEE/ACM Trans. Netw.*, 16(1):77–90, 2008.

[75] Thrasyvoulos Spyropoulos, Thierry Turletti, and Katia Obraczka. Routing in delay-tolerant networks comprising heterogeneous node populations. *IEEE Transactions on Mobile Computing*, 8:1132–1147, 2009.

[76] Jing Su, A. Chin, A. Popivanova, A. Goel, and E. de Lara. User mobility for opportunistic ad-hoc networking. *Sixth IEEE Workshop on Mobile Computing Systems and Applications, WMCSA .*, pages 41–50, 2004.

[77] Delay tolerant networking research group. http://www.dtnrg.org.

[78] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical report, CS-2000-06 Duke University, July 2000.

[79] Randolph Y. Wang, Sumeet Sobti, Nitin Garg, Elisha Ziskind, Junwen Lai, and Arvind Krishnamurthy. Turning the postal system into a generic digital communication mechanism. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, volume 34, pages 159–166, New York, NY, USA, October 2004. ACM Press.

[80] Yong Wang, Sushant Jain, Margaret Martonosi, and Kevin Fall. Erasure-coding based routing for opportunistic networks. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 229–236, New York, NY, USA, Aug 2005. ACM.

[81] Jörg Widmer and Jean-Yves Le Boudec. Network coding for efficient communication in extreme networks. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 284–291, New York, NY, USA, 2005. ACM Press.

[82] Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys & Tutorials, IEEE*, 8(1):24–37, March 2007.

[83] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *MobiHoc 2004: Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 187–198, New York, NY, USA, May 2004. ACM.

[84] W. Zhao, M. Ammar, and E. Zegura. Controlling the mobility of multiple data transport ferries in a delay-tolerant network. In *IEEE INFOCOM*, volume 2, pages 629–637, Aug 2005.

[85] S. Zhong, J. Chen, and Y.R. Yang. Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 3, pages 1987 – 1997 vol.3, march-3 april 2003.