Maximizing Routing Throughput with Applications to Delay Tolerant Networks

by

Mengxue Liu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2018 by the
Graduate Supervisory Committee:

Andrea Richa, Chair
Thienne Johnson
Violet R. Syrotiuk
Guoliang Xue

ARIZONA STATE UNIVERSITY

August 2018

ABSTRACT

Many applications require efficient data routing and dissemination in Delay Tolerant Networks (DTNs) in order to maximize the throughput of data in the network, such as providing healthcare to remote communities, and spreading related information in Mobile Social Networks (MSNs). In this thesis, the feasibility of using boats in the Amazon Delta Riverine region as data mule nodes is investigated and a robust data routing algorithm based on a fountain code approach is designed to ensure fast and timely data delivery considering unpredictable boat delays, break-downs, and high transmission failures. Then, the scenario of providing healthcare in Amazon Delta Region is extended to a general All-or-Nothing (Splittable) Multicommodity Flow (ANF) problem and a polynomial time constant approximation algorithm is designed for the maximum throughput routing problem based on a randomized rounding scheme with applications to DTNs. In an MSN, message content is closely related to users' preferences, and can be used to significantly impact the performance of data dissemination. An interest- and content-based algorithm is developed where the contents of the messages, along with the network structural information are taken into consideration when making message relay decisions in order to maximize data throughput in an MSN. Extensive experiments show the effectiveness of the above proposed data dissemination algorithm by comparing it with state-of-the-art techniques.

DEDICATION

To my husband, and

my parents

ACKNOWLEDGMENTS

I would like to first thank my advisor, Dr. Andrea Richa for her endless support during this journey. Her insightful suggestions and guidance significantly helped me achieving my goal. I'm always grateful for having her as my advisor.

I would also like to wholeheartedly appreciate all my committee members, Dr. Thienne Johnson, Dr. Violet R. Syrotiuk and Dr. Guoliang Xue for their valuable comments on my research and I really appreciate their help in my proposal defense and thesis defense. I would like to thank Dr. Yinong Chen for always being so kind and supportive during my work and study. I also would like to thank Dr. Stefan Schmid, Matthias Rost, Dr. Alon Efrat, Dr. Mauro Margalho Coutinho, Rachit Agarwal for their great collaboration on my research. I also want to thank the graduate advisors at CIDSE and the technical staffs for helping me on my requests.

I am very thankful for Jin Zhang, Xinhui Hu, Zahra Derakhshandeh, Xiang Zhang, Joshua Daymude, Taeyeong Choi, Xinsheng Li, Shengyu Huang, Ziming Zhao, Yilin Wang, Jung Hyun Kim, Parth Nagarkar who helped me a lot during my study at ASU.

Lastly, I am very grateful for my parents and my husband for always being there whenever I needed them.

TABLE OF CONTENTS

Page

LIST OF TABLES vii

LIST OF FIGURES viii

CHAPTER

1 Introduction 1

1.1 Transport Systems in the Amazon Delta Riverine Scenario 3

1.2 The All-or-Nothing (Splittable) Multicommodity Flow Problem 5

1.3 Mobile Social Networks with Interest- and Content-Based Dissemination 6

1.4 Dissertation Outline 8

2 Background and Related Work 10

2.1 Delay Tolerant Networks 10

2.2 CoDPON Architecture 12

2.3 Simulation Tools 13

2.4 Fountain Codes 14

2.5 Mobile Social Network 16

2.6 All-or-Nothing Flow (ANF) 19

3 Healthcare Supported by Data Mule Networks in Remote Communities of the Amazon Region 22

3.1 Introduction 22

3.2 Contributions 22

3.3 Simulations 24

3.4 Simulation Results For NS2 Simulator 27

3.5 Simulation Results For The ONE Simulator 28

3.6 Conclusion 32

CHAPTER Page

4 Robust Data Mule Networks with Remote Healthcare Applications in the Amazon Region: A Fountain Code Approach 34
4.1 Introduction 34
4.2 Contributions 34
4.3 Formal CoDPON model 35
4.4 Connection Graph 37
4.5 Robust Opportunistic Routing 41
4.6 Experiments 44
4.6.1 Transmission Affected by Rain 45
4.6.2 Transmission Loss 47
4.6.3 Boat Breakdown and Delay 48
4.7 Conclusion 50
5 A Constant Approximation for Maximum Throughput Routing 52
5.1 Introduction 52
5.2 Contributions 52
5.3 The All-or-Nothing (Splittable) Multicommodity Flow Problem 53
5.4 Randomized Rounding and Analysis 55
5.5 Application in Delay Tolerant Networks 60
5.6 Case Study 1: Riverine Amazon Region 61
5.6.1 Scenario 61
5.6.2 Experimental Results 62
5.7 Case Study 2: German50 Network 68
5.7.1 Scenario 69
5.7.2 Experimental Results 69

CHAPTER Page

5.8 Conclusions 73
6 Interest- and Content-Based Data Dissemination in Mobile Social Networks 75
6.1 Introduction 75
6.2 Contributions 76
6.3 System Model and Problem Formulation 76
6.3.1 System Model of MSN 76
6.3.2 Problem Formulation 79
6.4 Interest- and Content-Based Dissemination 80
6.4.1 Data File Analysis 81
6.4.2 Network Structure Analysis 82
6.4.3 Relay Selection 84
6.5 Evaluation Results and Analysis 84
6.5.1 Evaluation Setup 86
6.5.2 Real Trace: INFOCOM06 Trace 86
6.5.3 Real Trace: MIT Reality Trace 87
6.5.4 Evaluation Metrics 88
6.5.5 Results for INFOCOM06 Trace 88
6.5.6 Results for MIT Reality Trace 91
6.6 Conclusion 94
7 Conclusions 97
7.1 Future Work 98
REFERENCES 101

LIST OF TABLES

Table Page

3.1 Simulation Parameters 26

4.1 Simulation parameters 46

4.2 Fixed parameters 47

5.1 Average number of commodities received for different commodity sizes . 62

6.1 User Interest Profile for Each User 78

6.2 Content Representative for messages 80

LIST OF FIGURES

Figure Page

2.1 GUI of ONE Simulator 14

2.2 Fountain Code encoding and decoding process 16

3.1 Boats exchanging ultrasound files 23

3.2 Delivery of ultrasound files at the State capital 23

3.3 Connection time Simulation 25

3.4 Map of Marajó Archipelago with remote communities 25

3.5 Amount of data transferred 28

3.6 Number of medical exams transmitted 29

3.7 Packet Statistics 30

3.8 Number of Megabytes 31

3.9 Overhead Ratio 31

3.10 Latency Average (in Seconds) 33

4.1 Example of connection graph and respective max flow 41

4.2 Overall Robust Routing process 44

4.3 Percentage of files successfully received at final destination, for different number of generated packets with varying transmission loss probability, with 95% confidence intervals. 49

4.4 Percentage of files successfully received at final destination, for different number of generated packets with varying transmission loss probability and boats being delayed and broken, with 95% confidence intervals. . . . 50

5.1 Example of The All-or-Nothing (Splittable) Multicommodity Flow Problem 54

5.2 Edge Load for Commodity Size 7000MB in Amazon Scenario 63

Figure Page

5.3 Edge Capacity Violations for Commodity Size 7000MB in Amazon Scenario 64
5.4 Least Edge Load for Commodity Size 7000MB in Amazon Scenario 64
5.5 Number of Files Delivered after Randomized Rounding for Commodity Size 7000MB in Amazon Scenario 65
5.6 RR and LP comparison in terms of commodity delivery in Amazon Scenario 66
5.7 $(\alpha, \beta)-$distribution for Commodity Size 7000MB in Amazon Scenario . . 66
5.8 $(\alpha, \beta)-$distances for Commodity Size 7000MB in Amazon Scenario 67
5.9 $(\alpha, \beta)-$distribution for Commodity Size 7000MB in Amazon Scenario after scaling down edge capacities 68
5.10 $(\alpha, \beta)-$distances for Commodity Size 7000MB in Amazon Scenario after scaling down edge capacities 69
5.11 Average Edge Capacity Violation in German 50 Scenario 70
5.12 Number of Files Delivered after Randomized Rounding in German 50 Scenario 71
5.13 Least number of Edge Capacity Violation in German 50 Scenario 71
5.14 $(\alpha, \beta)-$distribution for German 50 Scenario 72
5.15 $(\alpha, \beta)-$distances for German 50 Scenario 73
5.16 $(\alpha, \beta)-$distribution for German 50 Scenario after scaling down edge capacities 74
5.17 $(\alpha, \beta)-$distances for German 50 Scenario after scaling down edge capacities 74
6.1 Using SVD to find row representatives for message content profile 82

Figure Page

6.2 PageRank score for node v on dynamic networks 83
6.3 Delivery performance comparison under different buffer size settings, and default TTL = 300 mins 89
6.4 Delivery performance comparison under different message TTL settings, and default buffer size = 100MB 90
6.5 Delivery costs comparison under different buffer size settings, and default TTL = 300 mins 91
6.6 Delivery costs comparison under different message TTL settings, and default buffer size = 100MB 92
6.7 Delivery performance comparison under different buffer size settings, and default TTL = 300 mins 93
6.8 Delivery performance comparison under different message TTL settings, and default buffer size = 100MB 94
6.9 Delivery costs comparison under different buffer size settings, and default TTL = 300 mins 95
6.10 Delivery costs comparison under different message TTL settings, and default buffer size = 100MB 96

Chapter 1

INTRODUCTION

Many network applications aim to find optimal paths and schedules for carriers to bring the data from sources to destinations, in order to maximize the throughput of data in the network. At a high level, if we can assume that network connectivity is always available, one can model this problem as a maximum throughput multi-commodity flow problem (formally defined in Section 1.2), where each commodity represents the flow of data from a given source node s to a given destination node t. If there is lack of continuous network connectivity, then one would have to think of data flowing for each commodity over a delay-tolerant network (DTN). A Delay-Tolerant Network (DTN) architecture [10] provides a common method for interconnecting heterogeneous gateways or proxies that employ store-and-forward message routing in order to overcome communication disruptions.

Transport systems are a great example of DTNs. Some transport systems typically rely on a fixed time schedule according to which nodes (e.g., boats, trains, buses, etc.) are moving between different locations (or stations). Such networks are hence also known as *time-schedule networks*: each node is scheduled to move along a predefined route on the network at a pre-defined time. Other such systems work in a more ad-hoc fashion, with nodes exchanging information whenever they meet.

While one may think of transport networks in the context of transporting passengers or goods, there exist several additional interesting use cases in the context of communication networks. Let us give three examples:

- *Data-mule networks in the Amazon region:* In a recent case study, together with collaborators from Brazil and the US, we show the feasibility of leveraging boats

as data mule nodes to carry medical ultrasound videos taken from pregnant women in remote and isolated communities in the Amazon region in Brazil, to the main city of that area [43]. Such videos are needed by physicians to perform remote analysis and follow-up routine of prenatal examinations.

- *Delay-tolerant Internet-of-Things (IoT) and social networks:* The quickly growing number of "smart things" equipped with wireless communication devices along with the ubiquitous deployment of Internet-of-Things technology introduces additional opportunities for data transmission [3, 9]. In such networks, smart devices may be moving between stationary nodes (the HotSpots) at which data can be uploaded from resp. downloaded to resp. from the Internet. Similar concepts exist in the context of delay-tolerant and mobile social networks, where the moving nodes may be people, and the stationary nodes may be classrooms, restaurants, shopping malls, etc. Many data routing and dissemination methods have been proposed to efficiently deliver data in mobile social networks, such as [32, 49, 51]. Recently, we developed an interest- and content-based approach [44, 45], which utilizes user interest, data content, as well as network structure to maximize the number of interested recipients.

- *Cost-Efficient transmissions in Internet Service Provider (ISP) networks:* Many scientific and industrial applications requiring the transfer of large, multi-Tbytes of data across continents are delay-tolerant. Accordingly, ISPs can reduce transmission costs by an intelligent scheduling of communications, taking advantage of already-paid-for off-peak bandwidth resulting from diurnal traffic patterns and percentile pricing [40].

Time-scheduled networks are time evolving and lack continuous network connectivity. However, one can transform such a network into a static connection graph

(as we show in 4.4), such that the maximum flow in this connection graph is equal to the maximum transmission capacity of the original time-scheduled network, i.e., is equal to the maximum total amount of data that can be successfully transferred from the original sources to the destinations. Therefore, one can alternatively view the problem of maximizing throughput in a time-scheduled network as a multicommodity flow problem in the corresponding connection graph, where one would like to maximize the number of commodities that are successfully transmitted from the respective sources to destinations.

In this thesis, we focus on three specific cases of maximizing throughput in delay tolerant and related networks, namely in (i) *Transport Systems in the Amazon Delta Riverine Scenario*, in (ii) *the All-or-Nothing (Splittable) Multicommodity Flow Problem* and in (iii) *Mobile Social Networks with Interest- and Content-Based Dissemination*. We give a brief introduction to these three scenarios in the sections to follow.

1.1 Transport Systems in the Amazon Delta Riverine Scenario

Efficient and effective data routing is of great challenge in many applications, such as providing healthcare to remote communities in the Brazilian Amazon. In the scenario of the riverine communities along the Amazon river and its tributaries in Brazil, there is a lack of even the most basic telecommunication infrastructure. Thus, finding alternative ways to enable communication at low cost is of vital importance for the people living in the region. One possibility is to utilize boats as data mules [6], since boats constitute the only mode of transportation in the region. Considering no strict requirements on delay, and if communications can bear disconnections, we can build a collaborative network and a data collection system to mitigate the telecommunication infrastructure problems. A prototype [21, 20] (see also [19] for more information) is

under construction at the *Laboratory for Research on Alternative Technologies Related to the Amazon (PETALA)*, at Universidade da Amazonia (UNAMA), Brazil, in partnership with the University of Arizona and Arizona State University, USA, and also in collaboration with the hospital Santa Casa de Belém and doctors in the region (see [19]).

Mobile Ubiquitous LAN Extension (Data Mule) refers to vehicles moving between remote areas that effectively create data communication links [63]. These vehicles usually carry a computer with a storage device and a limited telecommunication module (usually Wi-Fi). Data Mule and DTN technologies are both tolerant to disconnections in the network and are often complementary. They open doors to integrate hundreds of applications that were not possible before, mostly due to the high costs, or even infeasibility, of implementing a physical networked infrastructure in some scenarios.

Located in the Amazon region in northern Brazil, the Marajó archipelago, occupies a vast area of $104,142\text{km}^2$ and is an example of a scenario where implementing a networked infrastructure, if indeed feasible, would impose very high costs, that would not scale with the scarcely distributed population. Only 43% of the entire archipelago's population of $487,010$ inhabitants live in urban areas according to the last Brazilian census [33]. Boats are by far the main method of transportation in the region since there are no bridges connecting the islands to the mainland, some cities are completely built on water, etc. In these places, socio-infrastructure problems abound, especially health related problems, as there are few physicians available to the remote communities.

Most medical care in the region is done through sporadic government programs which involve the displacement of a medical team from the main city in the region, Belém, to serve the remote population of the Marajó archipelago, especially in the

outlying areas. An alternative would be to have local nurses or technicians perform routine clinical examinations, such as ultrasounds on pregnant women, electrocardiograms, or the monitoring of post-scalping restorative surgery patients [19], and have the examination records sent to the doctors in Belém for evaluation. However, due to the lack of modern communication infrastructure in these communities — e.g., no cellular networks and no Internet connection and of fast river transportation, one must fully utilize the regular ferry boats, with a predefined time schedule of routes and stops at the villages, and potentially also some of the local fisherman boats, as data mule nodes for timely delivery of healthcare data records (e.g., ultrasound videos, electrocardiogram records, post-surgery exam records and interview) to physicians in the city for remote analysis. An alternative would be to store the healthcare data records into USB drives and have them physically delivered to Belém. However, this would not be as practical or seamless as implementing the data mule network, since it would not allow for direct boat-to-boat transmission, basically eliminating the fisherman boats from consideration, and incurring higher communication delays.

1.2 The All-or-Nothing (Splittable) Multicommodity Flow Problem

Many applications such as transportation networks usually involve pre-defined time schedules and need to deliver data from source to destination, also in the scenario of Amazon Delta Region, medical files need to be transferred from remote communities to the main city. In order to maximize the files (commodities) that are successfully transmitted from source to destination in these routing problems, we could alternatively model them as an All-or-Nothing (Splittable) Multicommodity Flow (ANF) problem, which will be formally defined next. In ANF, one is given a set of (splittable) multicommodity flows, the goal is to select and route a subset of flows (all-or-nothing) such that throughput (the total number of satisfied commodities) is

maximized.

More formally, we model the network as a capacitated directed graph $G(V, E)$, where V is the set of nodes and E is the set of edges. Each edge e has a given capacity $c(e) > 0$. We are given a set of k commodities $\mathcal{F} = \{F_1, ..., F_k\}$, each with equal demand d. Each commodity $F_i \in \mathcal{F}$ is defined by a pair (s_i, t_i) where s_i and t_i denote respectively the source and destination for that commodity. Commodity F_i is satisfied if d units of this commodity can be successfully routed in the network. The objective is to maximize the total throughput in the network, where the throughput is measured in terms of the total number of commodities that are concurrently satisfied in a valid multicommodity flow. Note that we allow splittable flows, which means we do not insist that the flow for a commodity F_i be non-splittable or even that the flow on each edge be integral. The only assumption we make is that, for any i, the network has enough capacity to route d units of flow from s_i to t_i, if we were to consider commodity F_i alone in the network. The load of an edge e is equal to $\sum_i f_{i,e}$, where $f_{i,e}$ is the flow for commodity i on edge e, and splittable flow means that $f_{i,e}$ may not be d for any commodities.

1.3 Mobile Social Networks with Interest- and Content-Based Dissemination

In recent years, mobile social networks (MSNs) [23] emerged with the wide usage of hand-held mobile devices, such as smart phones and smart watches. This enables us to create, forward, and exchange information at levels that one could not envision a few years ago. MSNs are a particular case of delay tolerant networks (DTNs), where node mobility can be used to exploit contact and forward opportunities. An MSN can be represented by a graph where each node represents a user and each edge indicates the direct wireless communication between two corresponding users. In an MSN, a connection between two users emerges as the users come in direct contact with each

other and disappear when they are too far to allow for direct wireless communication. Over time, mobile nodes establish connections that may allow data to be transferred along a path linking disconnected parts of the network.

There are many challenges to be addressed in MSNs due to the node mobility and the absence of static end-to-end paths in the network. Consider the following example: a data source node tries to send a message to a destination node, where there is no direct path between the two nodes. However, with node mobility, the information can be carried by some mobile nodes and opportunistically relayed and forwarded. Using an appropriate routing scheme, the message can get closer to the destination node step by step until its arrival. Furthermore, when using smart phones as the media to transfer data, even with a large capacity battery, it is not realistic to repeatedly download and upload data, since these activities will drain the battery extremely fast. Therefore, it is essential to come up with an efficient and effective method that reduces redundant data transmissions and still enables fast delivery.

An application scenario in MSNs is that of data dissemination, where the source wants to spread out a message to as many nodes interested in the message as possible. For instance, commercial businesses want to propagate advertisements and coupons to potential customers. A key challenge is that the network information is unknown beforehand, including knowledge of node connections and user interests. The topologies, link latencies, and bandwidths of MSNs evolve over time, and each individual does not know its contacts in advance. Forwarding decisions need to be made based on limited "local" information, such as the set of neighbors of a node and the corresponding connection bandwidths given the current configuration of the MSN. Simply flooding the network by sending all the messages to all the nodes that each node encounters would be undesirable, as this would result in high bandwidth usage, severe network congestion, and large power consumption.

Content-based routing schemes which rely on the network structure and user interest profiles have been studied in the literature [9, 22, 17, 32, 49, 51, 65, 68]. There are also various works studying the impact on personalized content [42]. However, these works did not consider the impact of message content during data dissemination. Since users' preferences on message content can have a significant influence on the result of the dissemination, we propose an interest- and content-based data dissemination scheme, that takes the message content as well as the network structure and user interests into consideration.

1.4 Dissertation Outline

This dissertation is organized in the following way:

- In Chapter 2, we give an introduction of some basic concepts and terminology used in this thesis, and an overview of the existing works in related areas.
- In Chapter 3, we investigated the feasibility of using boats as data mules and present a preliminary evaluation of our proposed telemedicine infrastructure by simulations.
- In Chapter 4, we introduce robust opportunistic routing with fountain codes using boats as data mules to improve the message delivery and also incorporate the unpredictability of the Amazon riverine scenario into the simulation model.
- In Chapter 5, we extend our work to multiple sources and multiple sinks problem in a more general network, and present a constant approximation algorithm for a fundamental maximum throughput routing problem: the All-or-Nothing (Splittable) Multicommodity Flow (ANF) problem based on randomized rounding with the application to DTN and general static networks.

- In Chapter 6, we design an effective and efficient data dissemination scheme, which considers both data content and network structure for data dissemination in the mobile social networks.
- In Chapter 7, we conclude this dissertation.

Chapter 2

BACKGROUND AND RELATED WORK

In this section, we describe the relevant background and related work discussed in this dissertation.

2.1 Delay Tolerant Networks

Delay-tolerant networking (DTN) is an approach to computer network architecture that seeks to address the technical issues in a heterogeneous network that may lack continuous network connectivity, resulting in a lack of instantaneous end-to-end paths. In these challenging environments, popular ad hoc routing protocols such as AODV and DSR fail to establish routes. This is due to these protocols trying to first establish a complete route and then, after the route has been established, forward the actual data. However, when instantaneous end-to-end paths are difficult or impossible to establish, routing protocols must take to a "store and forward" approach, where data is incrementally moved and stored throughout the network in the hope that it will eventually reach its destination. A common technique used to maximize the probability of a message being successfully transferred is to replicate many copies of the message in the hope that one will succeed in reaching its destination. This is feasible only in networks with large amounts of local storage and internode bandwidth relative to the expected traffic. In many common problem spaces, this inefficiency is outweighed by the increased efficiency and shortened delivery times made possible by taking maximum advantage of available unscheduled forwarding opportunities. In others, where available storage and internode throughput opportunities are more tightly constrained, a more discriminating algorithm is required.

DTNs have been extensively discussed in various applications over the past few years, such as [29, 30, 34, 41, 74]. The first terrestrial DTN applications were proposed for sensor networks [67, 6, 38]. Researchers have integrated DTN into remote communities such as Daknet [54], which provided a low-cost digital communication allowing remote villages to leap from the expense of traditional connectivity solutions and the deployment of a full-coverage broadband wireless infrastructure. Other work involves addressing the communication and information access needs of remote rural villages that lack of modern communication technologies, such as MotoPost proposed in [52] and many more [55, 5].

In [58], the authors have considered a routing problem where nodes are moving and carrying information, and the edge between nodes might appear or disappear at certain times, making traditional mobile ad hoc routing protocols unusable. They proposed a space and time routing framework, constructed a space-time routing table which can choose the next hop node from the current and future neighbors using both arrival time and destination to calculate the next node, aiming at minimizing the delay time for transferring the information from source to destination.

In [3], the authors proposed a routing algorithm that aims at computing shortest routes based on a stochastic model of real-life bus traces in an urban network. They use buses as data carriers to deliver data in a timely manner to its final destination, tackling quasi-deterministic mobility scenarios. Their proposed routing algorithm outperforms other approaches that aim at minimizing the expected traversal time or at maximizing the delivery probability in the bus network. They did not consider direct data transmission between data carriers (buses).

2.2 CoDPON Architecture

Continuous Displacement Plan Oriented Network (CoDPON), proposed in [21], is a specific data mule system inspired by the air traffic control system. It consists of four components: nodes, DACT (data application in transit, data divided into logical units), displacement plans, and the communication protocol. There are three types of nodes: vehicles, peer base stations (PBSs), and hot spots. Vehicles act as mules, carrying the data spread along the path. The vehicles used on a CoDPON network operate on predefined routes, with prescheduled (days and times) departures and arrivals. In the Amazon scenario the vehicles are the ferry boats sailing along the river. Every boat has a computer housed in an airtight box and uses solid state disks to ensure more robust and persistent storage. The solid state disks prevent damage due to motion of boats.

Peer base stations are fixed nodes, previously mapped, where vehicles make scheduled stops. Peer base stations are usually located on the river shore. Hot spots are special nodes that exist only in areas that have a connection to the Internet. They work as a faster gateway between CoDPON nodes. Unfortunately, solutions such as the ones proposed in [63], which reduce latencies by equipping the MULEs with an always-on connection (such as a cellular or satellite phone), are not available in some scenarios (e.g., there is a very limited telecommunications coverage in Amazonia). A DACT represents a minimum unit of data transferred between nodes and has a self-meaning that allows pieces of data to be separated from others without losing consistency.

A displacement plan is applied to each boat. It contains basic information about the entire route of the boat, including starting point, destination, stopovers, and estimated time of journey and anchoring. Journey time is the time required for

a boat to travel its entire route. Anchor time usually is the time for passengers to embark or disembark at village piers. Each boat has its own displacement plan stored on board, including a table containing the hydrographic distances between peer base stations. The communication protocol is based on the principle of a data mule. It does not require changes in the TCP stack, as is the case with the DTN.

2.3 Simulation Tools

There are various existing open-source simulators, such as NS-2 and ONE. The Opportunistic Network Environment (ONE) [38] is a Java based simulator targeted for research in DTNs and its variants such as, Opportunistic Mobile Networks (OMNs). Apart from letting users simulate different scenarios quickly and in a flexible manner, the ONE also provides an easy way to generate statistics from the simulations performed. The ONE simulator can be run on Linux, Windows, or any other platform supporting Java. ONE is capable of

- generating node movement using different movement models,
- routing messages between nodes with various DTN routing algorithms and sender and receiver types, and
- visualizing both mobility and message passing in real time in its graphical user interface.

Figure 2.1 shows the GUI of the ONE simulator. On the top is the toolbar, where the simulation can started or paused , one can adjust the GUI update interval, and also zoom in and zoom out. In the middle we use OPENJUMP to draw the map of the Amazon Delta Region and the boats will move following the routes that we draw. When clicking on each node, boat or PBS, the messages its carrying will be shown

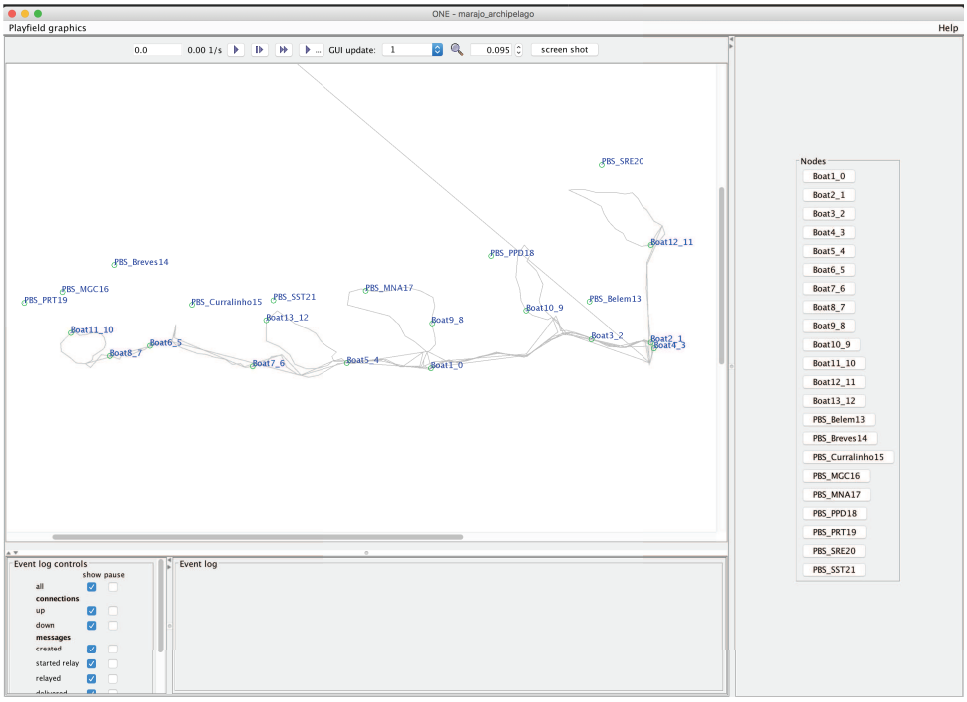

Figure 2.1: GUI of ONE Simulator

on the bottom. The event log will record all the traces about when a connection is up and down and message transfer.

2.4 Fountain Codes

The original CoDPON model and standard routing algorithms do not handle the situation where packets can be lost during the transmission. Fountain codes approaches are often used to tackle the challenge of packet loss and file recovery. Fountain codes are sparse-graph codes for channels with erasures [48]. Applications arise in the Internet, where files are partitioned into multiple small packets and each packet is either successfully transferred or not received at its destination node. Instead of simply partitioning a file into some number K of packets, fountain codes generate

a total of $N > K$ packets using random functions of the whole file. Once any N' packets are received successfully, where N' is slightly greater than the original number of packets K, but smaller than N, the whole file can be recovered. Hence fountain codes allow for some of the packets to be dropped without compromising the integrity of a file. Fountain codes are widely used in many applications and work, such as [62, 69, 72, 5].

Figure 2.2 shows an example of Fountain Code encoding and deconding process. There are K original data packets, here we have x_1, x_2, x_3, x_4, the packets are uniformly randomly selected to be encoded. N encoded packets are $y_1, y_2, y_3, y_4, y_5, y_6$, as we can see, encoded packet y_1 is a function of x_1 and x_2, y_2 is a function of x_1, x_2 and x_4, and so on. If we can receive any 5 y packets, then we can decode to get the original x.

In [48], three basic fountain code approaches are discussed. The random linear fountain code generates subsets of random K bits and tries to recover the original file if sufficient packets are received. The number of packets required to have probability $1-\delta$ of success is $K + \log_2 1/\delta$. Another approach is given by the LT codes proposed in [47], where an encoder is used based on a degree distribution that defines a graph connecting encoded packets to source packets, and a respective decoder is later used for recovering the original file. The LT codes retain the good performance of the random linear fountain code and also reduce the encoding and decoding complexities. A third approach, called Raptor codes [64], concatenates a weakened LT code with an outer code patching the gaps in the LT code, achieving linear time encoding and decoding.

Other fountain code approaches that involve unequal error protection (UEP) are discussed in [4, 56]. With UEP, different packets may have different weights and the redundancies are of different levels, so that more important packets have higher

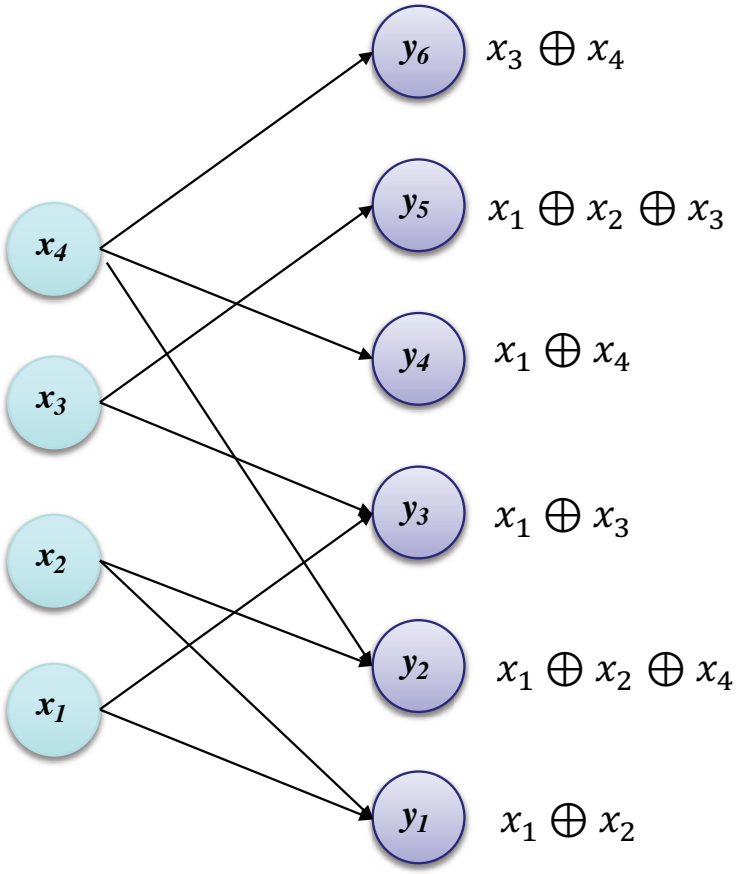

Figure 2.2: Fountain Code encoding and decoding process

probability of being delivered.

2.5 Mobile Social Network

There is a large body of research on data dissemination in MSNs as well. The Epidemic algorithm of [68] is a straightforward data routing and dissemination algorithm, where all the contents or messages are flooded into the network. It provides an upper bound on the delivery ratio since it utilizes every possible contact opportunity in a buffer unconstrained network. However, it results in high network congestion

and energy usage in practice and is also very sensitive to the nodes' buffer sizes, given the high communication overhead of this protocol. Other opportunistic-based strategies, such as BinarySW [65], were proposed to ease the traffic load, where nodes only spread the message to a limited number of its neighbors. Other routing protocols never replicate messages [9], in order to reduce the overhead in the network. Some more recent developments include [71], where the authors created a home-aware community model, and thus the minimum expected delivery delays of nodes could be computed via a reverse Dijkstra algorithm. However, these works did not consider the impact of users' interests and the contents of the messages for data routing and disseminations.

Social feature-based DTN routing and dissemination algorithms have been proposed to exploit the social network characteristics for efficient relay selection. In [49], Mei *et al.* proposed a social-aware stateless routing scheme SANE, based on the homophily [17] phenomenon where people with similar interests tend to meet more often. In SANE, each individual is represented by a vector that describes the topics he/she is interested in. A message will only be forwarded to the person who is interested in the message, by comparing the cosine similarity between the message content and the individual's interest. Similarly in [51], the authors spread the messages only to the nodes who are interested in the messages. A limitation of this strategy is that it does not take into account the network structure when relaying/spreading messages. LABEL [31] tries to cluster user nodes with the same label in one community, and assume people from the same community will meet more often, and the dissemination process is purely based on whether a node is having the same interest label that belongs to the same community. The SGBR dissemination scheme is designed in [2], where people are also clustered into groups based on the meet frequency instead of interest label compared to [31], and it only forwards the message to the node within the

same group and it will check also whether the node has the data that is being relayed. These types of approaches such as [31] fail to capture the network structure and do not consider the opportunity of the dissemination among different communities, and [2] fail to utilize node interest for better relay the data to the correct nodes.

Bubble rap [32] on the other hand, only considers the importance and popularity of a node in the network. By bubbling up the messages to the nodes with high centrality [73], the algorithm keeps forwarding the messages to nodes which have higher centrality until they reach the community where the destination node resides. Once the messages are in the local community, a local protocol is used to make effective forwarding decisions. Some other techniques, such as SimBet [22], use other network structural information like betweenness and similarity metrics to help make forwarding decisions. In [66], a software defined network is integrated into a mobile social network and data can be forwarded using social features. However, none of these works considers the impact of the actual message content in making forwarding decisions.

Sociability dissemination is developed in [28], where the authors design a metric called sociability and use it for choosing relay nodes. Sociability is calculated based on the number of social interactions from one node with other nodes, either using a direct one hop connection, or a multi-hop connection, and a node with higher sociability scores have higher probability of meeting the destination nodes. In this case, the algorithm takes account into network structure and assigns a higher relay probability for well connected nodes, however, it does not consider the content and node interests, and this may relay lots of data to recipients who are not interested in them.

2.6 All-or-Nothing Flow (ANF)

The study of routing in Multi-Commodity Flow problems is motivated by many real-world applications as well as the important role that flows and cuts play in combinatorial optimization [11]. There are two flavors of optimization problems related to our work: the *Maximum Edge-Disjoint Paths (MEDP)* [27] problem and the *All-or-Nothing Flow (ANF)* [15] problem. ANF is a "relaxed version" of MEDP for non-integral flows: In ANF, the goal is to select a largest subset of commodities that can be simultaneously fractionally routed from source to destination with regard to capacity constraints, whereas in MEDP the flow needs to be integral since the goal is to find the maximum number of edge disjoint paths. In other words, the MEDP problem considers a set of pairs to be routable if they can be connected using edge-disjoint paths; the ANF problem considers a set of pairs to be routable if there is a feasible multicommodity flow that fractionally routes one unit of flow from source to destination for each routed pair. Both problems, MEDP and ANF, are NP-hard [15]. More specifically, ANF is APX-hard even in the case when the underlying graph is a tree, and there exists a 2-approximation algorithm for the tree and 4-approximation algorithm when commodities are associated with weights: Kawarabayashi et al. [37] propose a constant-approximation algorithm in planar graphs, and also proved that the integrality gap is $O(1)$. Charbonneau et al. [61] studied the MEDP problem in planar graphs, and showed that a constant approximation is possible also with congestion 2, where congestion is defined by the largest edge capacity constraint violation ratio, which improves on [14] where the congestion is 4. Chekuri et al. [12] study the multicommodity flow and cut problem in polymatroidal networks, where there are submodular capacity constraints on the edges incident to a node, by analyzing the dual of the flow relaxations via continuous Lovász extension; the underlying graph

could be either directed or undirected. Chuzhoy [18] presented an approximation ratio of $\Omega(1/poly \log k)$ with constant congestion at most 14, using an efficient randomized algorithm in undirected graphs.

There are several existing approximations for ANF, which are closely related to this work. The study of the *Symmetric All or Nothing Flow (SymANF)* problem in directed graphs with symmetric demand pairs was initiated in [11]. In SymANF, the input pairs are unordered and a pair (s_i, t_i) is routed only if both the ordered pairs (s_i, t_i) and (t_i, s_i) are routed, and the goal is to find a maximum subset of the given demand pairs that can be routed. The authors provide a poly-logarithmic approximation with constant congestion for SymANF, by extending the well-linked decomposition framework of [13] to the directed graph setting with symmetric demand pairs.

Our work differs from Chekuri et al. [11] in that their results depend on a more restricted assumption of unit edge capacity and symmetric unit demand. My work considers a more general setting and our result of constant approximation with poly-logarithmic congestion is not directly comparable to theirs. Based on their observation, from previous work on the hardness of the ANF problem, the throughput for the SymANF with constant congestion c is hard to approximate to within a factor of $(\log |V|)^{\Omega(1/c)}$.

The most closely related research to our work is probably [15], where the authors present an approximation algorithm for the general ANF problem with constant congestion and approximation ratio of $\Omega(1/(\log^3 |V| \log\log |V|))$ based on hierarchical graph decomposition. While Chekuri et al. [15] present an approximation algorithm for the ANF problem which achieves constant β and α in $\Omega(\frac{1}{\log^3 |V| \log\log |V|})$, our approach keeps α constant and allows β not to be constant. However, it is not clear how one can obtain a constant throughput by modifying the proposed algorithm, while

keeping the constant congestion.

Several other applications also involve the ANF, such as [36], which solved the ANF to help for designing route networks for container ships, in particular, the authors studied the ANF with transit time constraints and proved that including time constraints does not necessarily increase the computational time. Finally, our work leverages randomized rounding techniques presented by Rost et al. [57] in the different context of virtual network embedding problems (i.e., flow endpoints are subject to optimization).

In terms of our case study in the context of Delay-Tolerant Networks, there also exists much related work, e.g., in the context of Daknet [54] which integrates remote communities and provides low-cost digital communication allowing remote villages to leap from the expense of traditional connectivity solutions and the deployment of a full-coverage broadband wireless infrastructure. Other work involves addressing the communication and information access needs of remote rural villages that lack of modern communication technologies, such as MotoPost proposed in [52]. In [3], the authors proposed a routing algorithm that aims at computing shortest routes based on a stochastic model of real-life bus traces in an urban network. They use buses as data carriers to deliver timely data to its final destination, tackling quasi-deterministic mobility scenarios. Their proposed routing algorithm outperforms other approaches that aim at minimizing the expected traversal time or at maximizing the delivery probability in the bus network. They did not consider direct data transmission between data carriers (buses).

Chapter 3

HEALTHCARE SUPPORTED BY DATA MULE NETWORKS IN REMOTE COMMUNITIES OF THE AMAZON REGION

3.1 Introduction

A major difficulty for healthcare in isolated areas, like the Marajó archipelago, occurs due to the small number of physicians who reside in the region. The vast majority of those just make periodic visits to the communities in the region. Among the most affected by the infrequent medical visits in the communities are pregnant women, who need regular monitoring. Hence, we considered the proposed application of remote ultrasound monitoring. The CoDPON system will act as a data mule network, carrying and delivering ultrasound files from the remote communities (see figure 3.1) to the state capital, Belém, for analysis by physicians (see figure 3.2).

The ultrasound parameters used in the simulation process were obtained by consulting a team of experts on ultrasounds, including OBGYN doctors in Belém, who operate in the Marajó region. According to the team, the ultrasound examination is a dynamic diagnostic method in which the physicians acquire the ultrasound videos and interpret them in a dynamic way. Two synchronized videos are required for the remote analysis to be efficient: the first video shows the image generated by the equipment and the second video shows the examination process (the position of the transducer being used by health agent on the patient's body).

3.2 Contributions

My contributions in supporting healthcare in Amazon Delta Region are three-fold:

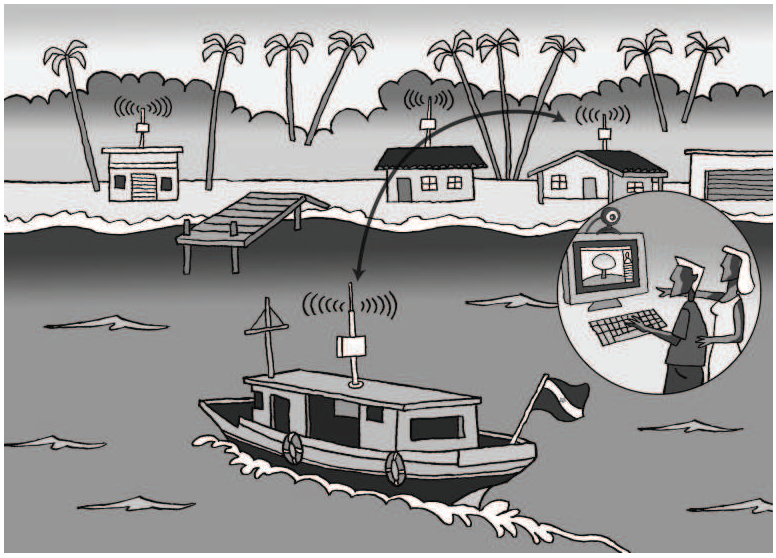

Figure 3.1: Boats exchanging ultrasound files

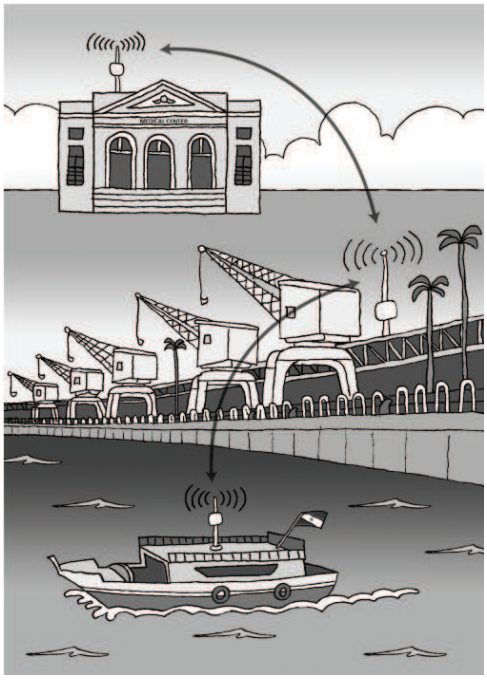

Figure 3.2: Delivery of ultrasound files at the State capital

- The Marajó archipelago is transformed into CoDPON architecture, which uses ferry boats as data mules, and remote communities as peer base stations.
- The CoDPON architecture is imported into the NS-2 simulator to evaluate the data delivery performance under various settings.
- The CoDPON architecture is imported into the ONE simulator to evaluate the data delivery performance under various settings.

3.3 Simulations

Two open source simulation tools were selected for the evaluation of the scenario described in the Marajó Archipelago. The first tool was NS-2, which is a most widely used simulator in the scientific community [35]. The NS-2 wireless module was used to inspect the connection time between boats (in motion) and Peer Base Stations and between a boat and another boat (both in motion) as shown in Figure 3.2. The other tool was the ONE simulator [38], which has rich features available for simulating DTNs with numerous mobility models.

For the simulation, an area of 104,142 Km^2 comprising the Marajó Archipelago, located in the state of Pará, Brazil was selected (the south part of the Marajó Island is shown in figure 3.4). The routes for the boats were created according to local data, obtained from the office of the public boats terminal in the city of Belém (capital of Pará). The boats are equipped with Wi-Fi IEEE 802.11 devices and they follow routes throughout the simulation area, following predefined routes. Each boat has scheduled stopovers at various Peer Base Stations. Table 3.1 describes the parameters that are used in the simulations.

The scale of the simulations conducted in NS-2 and The ONE simulator were very different: The simulation in NS-2, we evaluated the amount of data transmitted

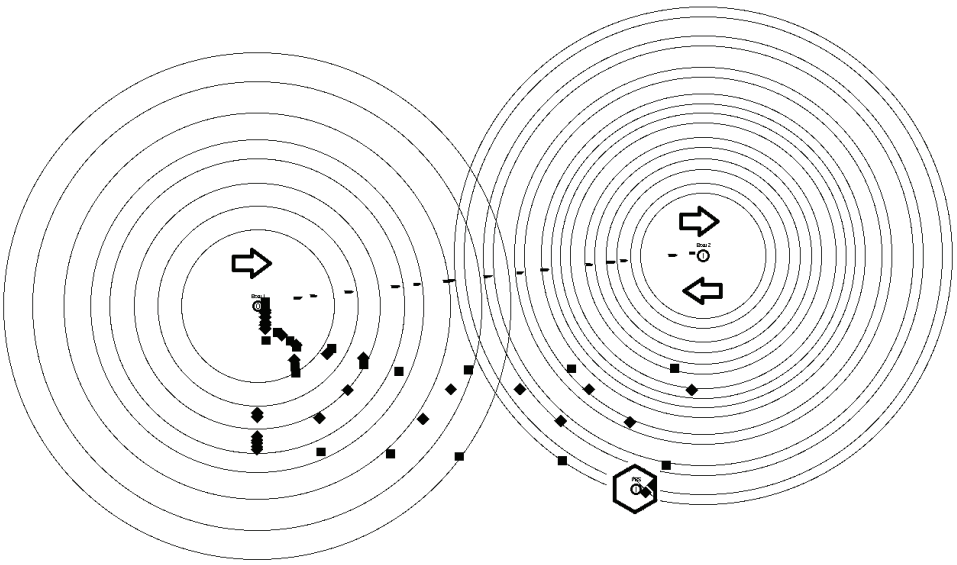

Figure 3.3: Connection time Simulation

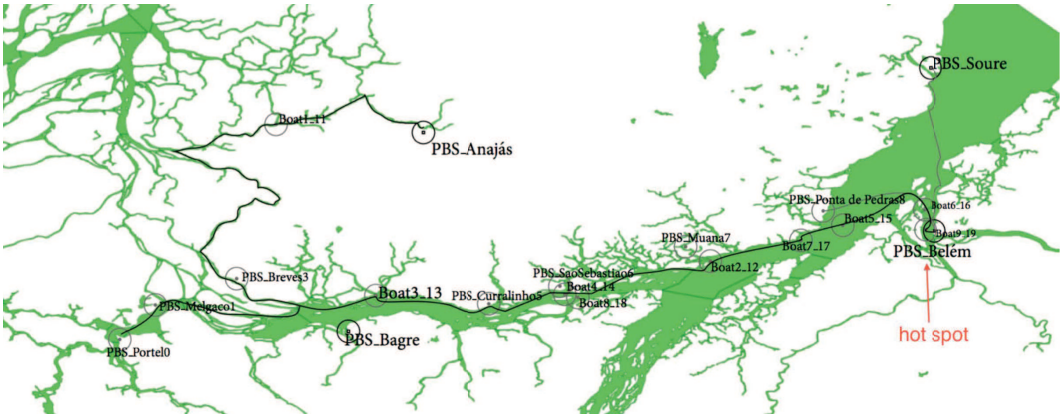

Figure 3.4: Map of Marajó Archipelago with remote communities

Parameter	NS-2	The One
Flat Grid / World Size (Km)	2000x200	22000x20000
Simulation Time (second)	1000	216000
Wireless Interface Transmit Speed (Mbps)	11	11
Wireless Interface Range (m)	100	100
Radio Frequency (GHz)	2.4	-
Path Loss Exponent (Rain)	2	-
Average boat speed (Km/h)	15	15
Number of PBS (DTN Module in Seashore)	1	11
Displacement Plans	-	9

Table 3.1: Simulation Parameters

between boats and PBSs, which required a finer grain simulation and more specifics about the radio communications. The simulations conducted in The ONE simulator concerned the flow of packets over the network, following the end-to-end boat displacement plans. As some routes are traveled for up to seventeen hours, the Time to Live (TTL, maximum time that a message could still live before arriving at the destination) value was set to 1122 minutes, 10% more than the value of the maximum travel time. When the TTL expires the message will be self-destroyed, if it has not been delivered yet.

In order to configure the traffic generator for the simulator, the demand of real data loads in Marajó were investigated. During the sporadic government health programs, physicians perform one hundred ultrasound exams in one weekend, once a month. In other words, working at maximum capacity, they perform fifty exams per weekend day. If we consider that each ultrasound size, in Mpeg format, is around

5 Megabytes (using the minimum acceptable standards for digital compression of a fetal ultrasound video-clip), and we add the video of a health agent performing the examination, 10 Megabytes is a reasonable size for each ultrasound video set. Thus, the traffic generator of the ONE simulator was set up to work at 384kbps. The value was obtained considering that fifty ultrasounds per day are performed, each generating a 10 MB ultrasound file. Three PBSs, located at the extreme points of the Marajó archipelago, were selected as sources of video transmissions, all of them sending files to the main PBS in Belém. The average boat speed of 15 Km/h was obtained by GPS.

3.4 Simulation Results For NS2 Simulator

Once a boat stops at a PBS, it could receive DACTs, due to its large buffer (SSD disk). Therefore, the investigations were concentrated in critical points, that is, when the data transfer occurs while the boats are moving. In the simulation, three different scenarios were tested.

Scenario 1 has a boat passing within 50 meters of another boat in the opposite direction. In this case, both boats are in motion. Scenario 2 has a boat passing within 50 meters of a Peer Base Station, which is a fixed point, without stopping. Scenario 3 has two boats in movement, moving in the same direction, one traveling at twice the speed of the other. In all scenarios, connection times and amount of data transferred were analyzed (Fig 3.5).

In the first scenario, transmission occurs between seconds 172 and 320 of the simulation time line, which represents a duration of 148 seconds. This results in a total of 2.3 minutes of connection time between the boats. This would allow for the transfer of two ultrasound video sets.

In the second scenario, transmission occurs between seconds 99 and 381 of the

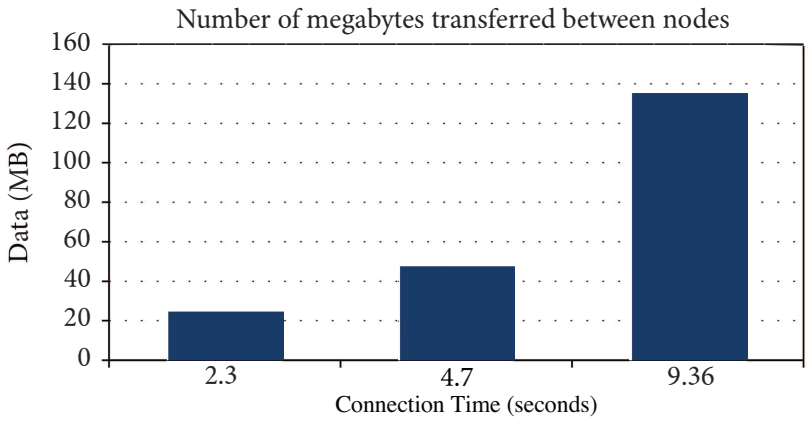

Figure 3.5: Amount of data transferred

time line simulation, which represents a total of 282 seconds. This results in a total of 4.7 minutes of connection time between a boat and a PBS. With these values it would be possible to transfer 47.62 Megabytes, which would allow the transfer of four ultrasound video sets.

In the third scenario, transmission occurs between instants 198 and 760 of the time line simulation, which represents a total of 562 seconds. This results in a total of 9.36 minutes of connection time between the boats. With these values, 135.08 Megabytes would be transferred, which would allow the transmission of thirteen ultrasound exams sets (Fig 3.6).

3.5 Simulation Results For The ONE Simulator

The map of the Marajó archipelago was obtained from IDESP (Institute of Economic, Social and Environmental Development of Pará). Using the open source GIS tool OpenJump. The map was post-processed and marked with various locations such as villages, small cities, and the route of the boat lines. The scenario used in

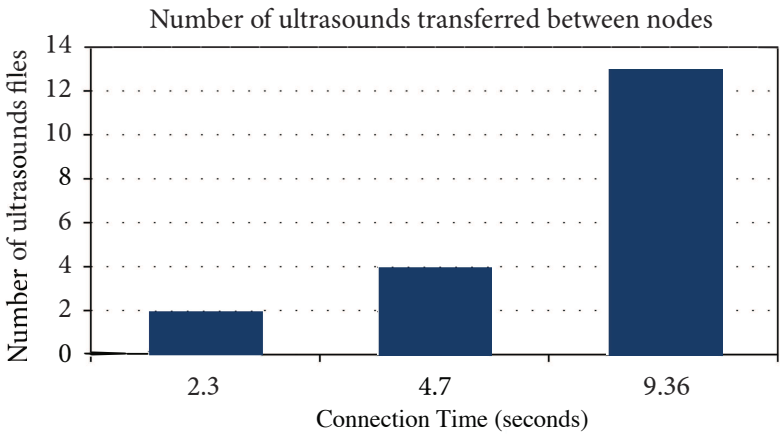

Figure 3.6: Number of medical exams transmitted

this section aims to transfer ultrasound sets from remote communities of the Amazon to Belém. The ONE simulator scenario comprises 11 PBS, and only three of them (PBS_{Anajs}, PBS_{Bagre}, and PBS_{Soure}) generated ultrasound exam files. Each boat has an associated route file defined by the OpenJump software, based on real routes obtained from the main boat station in Belém. Epidemic routing was used in the simulations. The evaluation was performed by changing the number of boats in the entire scenario, not only between source and destination. The metrics were compared with one, three, five and seven boats acting in each route. Figure 3.7 shows the relation between started, dropped, relayed and aborted packets. Started packets are the packet transmissions started between network nodes. Dropped packets are the packets that are dropped from nodes buffers and relayed packets are the packets successfully transmitted between nodes. As it can be seen, the number of dropped packets rises with the number of boats, due to the epidemic routing, which indicates a rapid saturation of the network capacity. This is due to having more boats carrying packets the wrong way (in the opposite direction of the PBS it should be going to),

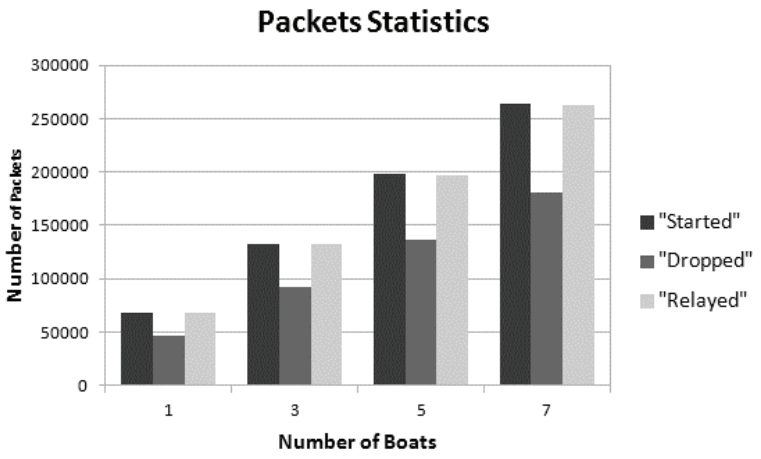

Figure 3.7: Packet Statistics

which end up wandering in the network without ever making it to Belém.

Despite the fact that the delivery probability increases with the increased number of boats as shown in Figure 3.8, the amount of wandering packets in the network grows in a much larger proportion as shown in Figure 3.9, also rapidly saturating the network capacity. This is clear when the overhead ratio is observed. For an epidemic-like protocol, the overhead ratio can be interpreted as the number of replicas for each message transmission. In an environment without QoS, this could be a problem because of the large number of messages to be delivered.

Without a prioritization mechanism, the most important messages can be delivered late and may not be relayed in time. In a CoDPON network, QoS prioritization is based on the Olympic service (QoS architecture that classifies messages as gold, silver and bronze: gold goes first, then silver, and bronze is considered best effort) [17] to avoid this problem.

Figure 3.8 shows the delivery probability, which is the probability related to the number of successfully delivered messages. This calculation excludes the messages

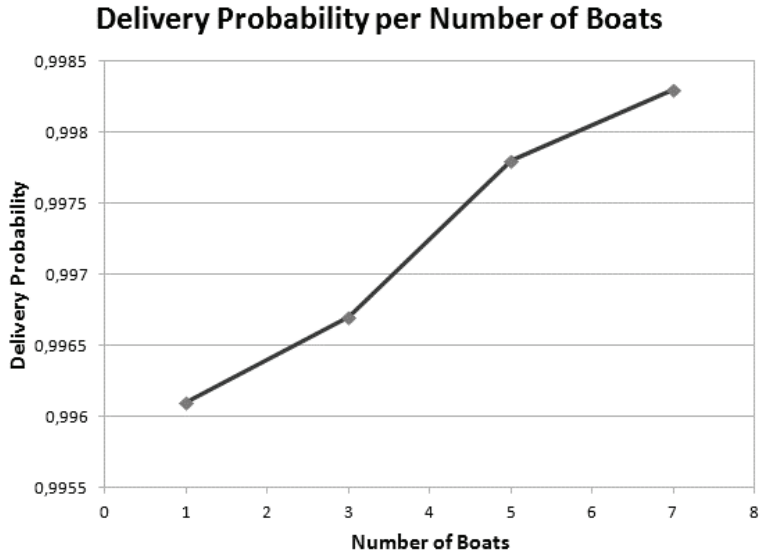

Figure 3.8: Number of Megabytes

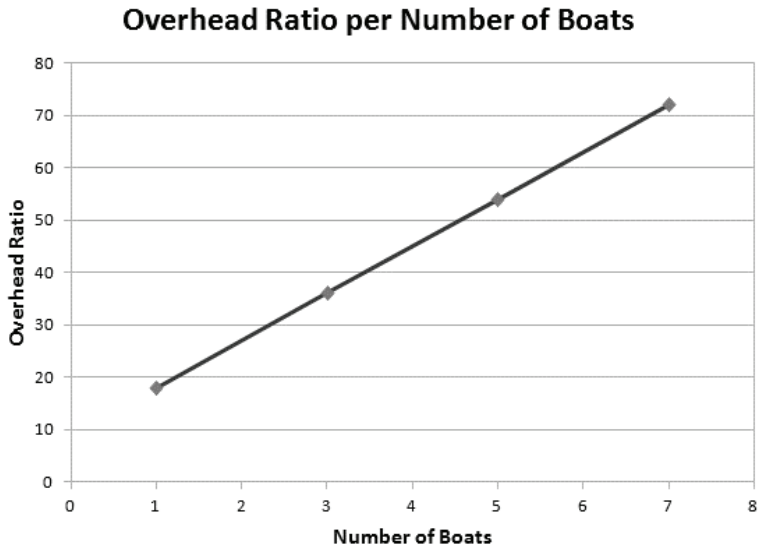

Figure 3.9: Overhead Ratio

sent in the opposite direction with boats not traveling to the destination. It grows linearly accordingly to the increase in the number of boats. This information can be used by network administrators and local government to estimate how efficient the service is. Depending on the results, more boats can receive the necessary equipment to become nodes in a CoDPON network. Figure 3.10 shows the average message latency (in seconds) from creation to delivery. It drops as more boats are used to carry the packets.

The simulation performed can be used to scale the number of boats that will act as data mules in the system and the impact of the use of traditional routing protocols like epidemic routing, for example, to decrease the latency with an increased number of boats.

It is clear that the proposed system has a high delivery ratio with high overhead. But as our goal is to provide medical services to underserviced communities, the delivery of the messages is our priority, thus the overhead is the issue that needs to be improved but it is acceptable as a starting point.

3.6 Conclusion

The demand for health services is large in the Amazon region. However, due to the lack of communication and transportation infrastructure, many patients who require special care end up unattended. One of the most important aspects of my proposed system requires the health experts to be involved in the preliminary analysis of patients medical files. For example, if a large number of neurological problems is detected in the analysis of the remote preliminary examinations files, neurologists may be included in the group of doctors in the boat caravan to the villages.

The simulations clearly indicate that epidemic routing creates a very large burden on the network, since the packet spreads in all directions without discretion. In a

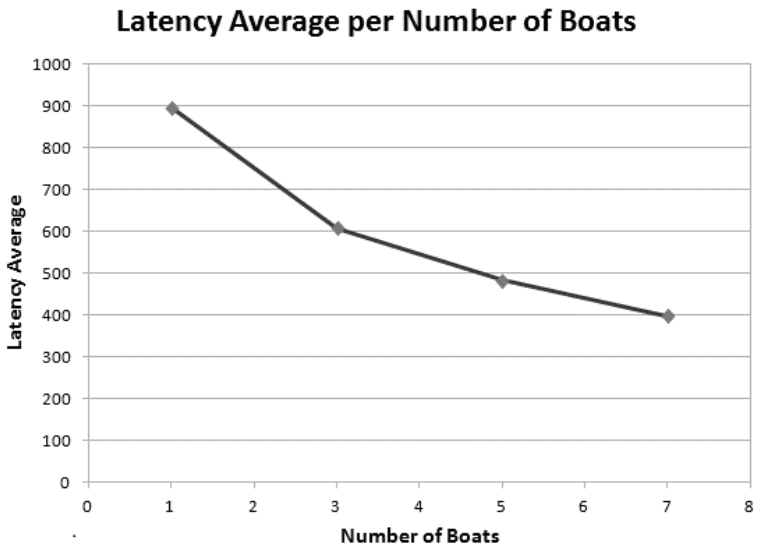

Figure 3.10: Latency Average (in Seconds)

network with known routes, as is the case of CoDPON, where the PBSs may act similarly to CheckPoint [10] spreading packets, the spread can be controlled to minimize the amount of unnecessary transfers, since there is some control available through the displacement plans.

Chapter 4

ROBUST DATA MULE NETWORKS WITH REMOTE HEALTHCARE APPLICATIONS IN THE AMAZON REGION: A FOUNTAIN CODE APPROACH

4.1 Introduction

As we described in Chapter 3, in the remote and isolated communities in the Amazon Delta Region, there is a lack of modern communication infrastructure due to the high cost of setting up satellites or cell phone services, and hence providing healthcare to these places is a major challenge. Boats are the main method of transportation since there are no bridges connecting the islands to the mainland, nor roads connecting the riverine villages to the main city Belém. Medical care is done through sporadic government programs or routine clinical examinations that will be sent to main city for further evaluation. In order to tackle this challenge, we propose the use of regularly scheduled boats as data mules to ensure fast and timely delivery of the examination records from those communities to physicians in the city for remote analysis. Unpredictable boat delays and break-downs, as well as high transmission failures due to the harsh environment in the region, mandate the design of robust delay-tolerant routing algorithms.

4.2 Contributions

The main contributions of using a fountain code approach in robust data mule networks are two-fold:

- we incorporate the *use of fountain codes in order to improve the robustness*

of the proposed opportunistic data routing algorithm (without an exponential increase in the amount of data traffic); see Section 4.5.

- we incorporate the *unpredictability of the Amazon riverine scenario into our simulation model*, accounting for boats being delayed or breaking down, as well as for the loss or slow down in data transmissions, and present extensive simulation results to evaluate my proposed use of a fountain-code based routing algorithm (Section 4.6).

While the results focus on remote healthcare applications in the Brazilian Amazon, we envision that our approach may also be used for other remote applications, such as distance education, and in other similar scenarios around the world.

4.3 Formal CoDPON model

In this section, we present a more formal description of a CoDPON, first introduced in [21, 20] and illustrated in Figure 3.4. Later, we present my *connection graph model* whose nodes correspond to the boat-boat and boat-PBS connections in the CoDPON network, showing that a max flow in the connection graph corresponds to the maximum transmission capacity that can be sustained in the original network.

Denote by $\mathcal{B}$=$\{B_1, B_2, B_3, ..., B_m\}$ and $\mathcal{P}$=$\{P_1, P_2, P_3, ..., P_n\}$ the sets of m boats and the set of n PBSs in the CoDPON, respectively. we assume that both the boats and the PBSs have infinite buffer sizes[1]. Let DP_{B_i} be the displacement plan associated with boat B_i, which describes the routing paths, and arrival and departure times of B_i at the PBSs (we assume that each boat has full knowledge of its own displacement plan). A connection is established between two boats B_i and B_j at time t if and only

[1] A reasonable assumption, given the ever-dropping costs of memory and also the much more stringent data transmission bottlenecks due to boat-boat transmissions while in transit.

if their geographical distance is within a certain constant range $r_{i,j,t}$ at time t. Note that the ranges $r_{i,j,t}$ may be all different depending on the particular boats and time (which also determine location) of the connection. However, without loss of generality and for ease of explanation, we will assume that $r_{i,j,t} = r$, for all i, j, t. Given also the sparsity of our network scenario, we will ignore considerations of interference of the wireless signal in this work, but we do assume that all communications are half-duplex (i.e., a node can either transmit or receive at a time). A connection exists between B_i and B_j at time t if and only if $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r$, where (x_z, y_z) are the coordinates of the respective boat B_z at time t.

We also assume that we have a set of f ultrasound files $\mathcal{F}=\{F_1, F_2, ..., F_f\}$, which can be transferred through senders and receivers equipped on the boats and PBSs when a connection is established. Ultrasound files are regularly uploaded onto the PBS of the respective community, as they are generated by the ultrasound technician and will be forwarded until they reach the destination hot spot in Belém. Though PBSs and boats have unlimited memory capacity, the boat-boat and the boat-PBS connections have a limit on the amount of data transfer because of the time in harbor, and of distances and speeds of the boats. To ensure the timely delivery of the files, an intelligent routing algorithm should be developed.

Since the displacement plans of the boats are given ahead of the time, we can find out all the possible boat-boat and boat-PBS connections before boats start cruising. Based on this observation, we transform our original CoDPON into a *connection graph*, which we introduce in the next section.

Note that while we use ultrasound examinations to illustrate a concrete application of our telemedicine network infrastructure, the algorithms and auxiliary models we develop would be applicable for *other telemedicine and non-telemedicine applications* which rely on data transfers from remote communities to certain destination points,

within and outside the Amazon riverine scenario that we use in this work.

4.4 Connection Graph

Based on the original boat displacement plans, we can then determine all boat-boat and boat-PBS connections $\mathcal{C}$=$\{C_1, C_2, C_3, ..., C_k\}$ where $k \leq cm(m+n)$ and c is a constant.

We can represent each connection C_i as a 4-tuple $< A_i, B_i, Up_i, Down_i >$, where A_i and B_i are the two objects (each object is either a boat or a PBS) that establish this connection, Up_i is the starting time of the connection, and $Down_i$ is the ending time of the connection.

We construct the directed connection graph using boat-boat and boat-PBS connections and files as nodes, therefore the node set is $V = \mathcal{C} \cup \mathcal{F}$ and $|V| = (k+f)$. A directed edge exists from connection node C_x to connection node C_y if and only if the two connections share a common object and $Up_x \leq Down_y$. For example, there is an edge from a connection node $< 1, 2, 300, 400 >$ to a connection node $< 2, 5, 500, 600 >$ since they share the common object 2 and $300 \leq 600$. The capacity of an edge $e = (C_x, C_y)$ is defined as

$$Cap_e = min\{M(A_x, B_x), M(A_y, B_y)\} \tag{4.1}$$

where $M(A_x, B_x) = L(C_x) \cdot v$, v is the data transfer speed (we assume it to be a constant in the absence of bad environmental conditions), and $L(C_x) = Down_x - Up_x$ is the lifetime of C_x. We could also round down the capacities of edges so that they reflect entire data packets being transmitted.

When two connections with at least one object in common overlap in time, we will take a conservative approach and assume that the overlapping transmissions from the two respective connections collide (generate interference) and hence that during the

overlap time no successful transmissions occur for these connections. Hence, whenever two connections that share an object overlap, we will not count the overlap time when computing the capacity of the edge between them.

Consider now the edges in the graph between a file node and a connection node. We represent each file F_i as a node $F_i =< p_i, Generate_i >$, where p_i is the PBS of the respective community where F_i was generated, and $Generate_i$ is the time when F_i was generated. There will be an edge from F_i to connection node $C_x =< A_x, B_x, Up_x, Down_x >$ if and only if $p_i = A_x$ or $p_i = B_x$, and $Generate_i \leq Down_x$. The capacity of a file-connection edge (F_i, C_x) is calculated as:

$$Cap_{(F_i,C_x)} = min\{FileSize, (Down_x - Generate_i) \cdot v\} \tag{4.2}$$

where v is the data transfer speed.

In order to find an optimal routing strategy that maximizes the total amount of data transferred from the several originating PBSs to the possibly multiple destination hot spots in Belém, we can add a virtual source and sink nodes to the connection graph: the virtual source node will have an edge of capacity equal to the size of file F_i into the node corresponding to F_i, for all i, and the virtual sink node will have an incoming edge of infinite capacity from every destination PBS in Belém. We claim that a maximum flow algorithm on the connection graph is equal to the maximum transmission capacity of the original CoDPON network, i.e., is equal to the maximum total amount of data that can be transferred from the several originating PBSs to the destination PBSs in Belém, assuming that there are no boat breakdowns or delays nor any transmission failures, as we prove below.

Theorem 4.4.1 *A maximum flow from the virtual source to the virtual sink node in the connection graph represents the maximum transmission capacity in the original*

CoDPON network, assuming that all boats adhere to their regular schedules and that there is no data packet transmission loss.

Proof Assume that each file is partitioned into much smaller data packets of size PS, where PS is less than the smallest capacity of an edge in the connection graph (which would trivially be true in practice). We argue that the max flow we obtain on the connection graph from the virtual source to the virtual destination nodes indeed directly gives an optimal routing scheme for the data packets that maximizes the number of data packets sent from the respective file nodes to the destination PBSs in Belém. First note that the amount of flow that can leave any of the file nodes is at most equal to the respective files size, given the capacity of the edges out of the virtual source node. Second, all the edges in the connection graph respect time, in the sense that they only connect connection and file nodes respecting the order that they occur in time; the edges also only exist when the two corresponding endpoints share a common object, so that a feasible sequence of file transfers is possible. Hence if break a flow in the connection graph as a superposition of the flow path for each data packet, we see that each of these data packet flow paths respect a sequence of valid transmission transfers for the data packet. Hence any flow in the network corresponds to a feasible routing scheme for all the data packets generated. Conversely, any routing scheme for the data packets can be converted into a feasible flow in the connection graph.

Therefore, a maximum flow in the connection graph also gives us a set of routing paths for the data packets that maximizes the number of data packets sent from the PBSs where the files are uploaded to the destination PBSs in Belém. □

Figure 4.1 shows an example of a connection graph and the respective maximum flow. We use a solid line to indicate the flow and a dashed line to indicate that the

respective edge carries no flow. We omitted edges between connections representing the same pair of objects at different points in time since they are redundant. On each edge, there's a pair of numbers $a(b)$, where a represents the capacity of the edge and b is the actual flow on the edge. There are two files F_1 and F_2 and each file has size 1000B so they all have edges from the source node with capacity 1000B. File F_1 is generated from PBS_1 at time 2:00 so it has edges to connection node C_2, which contain PBS_1. The edge capacity is limited by the connection life (30 min) and transmission speed (10B/min) instead of file size 1000B. File F_2 is generated from PBS_3 at time 1:00 so it has edges to connection nodes C_3 and C_6, which contains PBS_3.

When two connection nodes overlap, for example, node $C_2 =< 1, 2, 3{:}00, 3{:}40 >$ and node $C_5 =< 2, 5, 3{:}20, 4{:}00 >$, we need to subtract the overlap time, in this case, 20 min so the lifetime of connection C_2 is $L(C_2) = 20$ min and lifetime of connection C_5 is $L(C_5) = 20$ min. Hence, the capacity of the edge between C_2 and C_5 is 20 min multiplied by the transmission speed (10B/min) which is equal to 200B. When running the max flow algorithm on this connection graph model, the resulting flow is 300B.

Hence, we can see that, when we have limited connection time and transmission speed, the percentage of files delivered successfully could be really low even when relying on an optimal, global max flow-based algorithm.

We estimate the total time required to construct a connection graph from the given CoDPON network, and also the amount of time necessary to run a max flow algorithm on the connection graph, in terms of m, n, and f (number of boats, PBSs, and ultrasound files respectively). Since we need to iterate through each connection node and each file node in order to establish the edges out of each node in the connection graph, the edge building process will take $O(k(k+f))$ time and the number

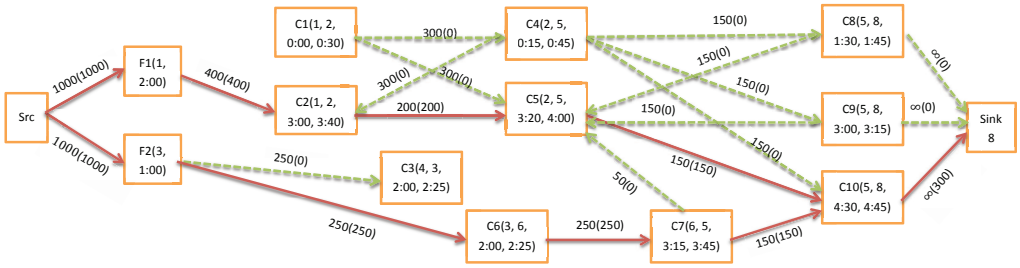

Figure 4.1: Example of connection graph and respective max flow

of edges $|E| \leq (k(k+f))$ where k is the number of connection nodes. We use the classical Edmonds Karp algorithm [26], so the running time is: $O(ke^2) = O(k(k(k+f))^2) = O(k^3(k+f)^2) = O(c^3m^3(m+n)^3(cm(m+n)+f)^2) = O(c^3m^6(m^2+f)^2) = O(Cm^6(m^4+2m^2f+f^2))$ where c and C are constants. Note that n is relatively small compared to m. Since in our scenario all m, n, f are not large numbers, we were able to use this approach for simulations that test the optimal capacity of the respective DTN network, as the size and numbers of files to be transmitted in the network increase.

4.5 Robust Opportunistic Routing

Frequent boat delays and break downs present real challenges, significantly affecting their displacement plan and connections with other boats and PBSs. Moreover, due to challenging environmental conditions in the Amazon delta region, wireless communication is often disrupted or slowed down significantly due to rain and mist,

as well as the strong current and tidal waves resulting from the *pororoca* and dense vegetation [1].

Hence, one needs to consider approaches that would enable the system to be more robust to frequent changes in the boats' displacement plans and any loss in data transmission that may occur. On one side of the spectrum would be to utilize *epidemic routing* [68] as the underlying routing mechanism. In epidemic routing, every packet of data is replicated and forwarded at every connection (boat-boat or between boat-PBS). In a system without any constraints on the capacity of data transmissions, epidemic routing would guarantee fastest delivery times for every data packet, since the many replicas of a data packet explore *all* the possible delivery paths for the given packet. Given the very high level of replication, epidemic routing is also very robust in capacity unconstrained systems, since the chance that at least one of the replicas make it to one of the destination PBSs is high. The problems with epidemic routing appear when there are limitations on the capacity of the network (i.e., on how much total data the network can sustain from source to destination nodes), since epidemic routing increases the volume of traffic in the network exponentially on the number of connections. This phenomena has already been verified for the CoDPON network in [21, 20].

On the other side of the spectrum is opportunistic routing, which never replicates a data packet, and which forwards the single original copy of a data packet during a connection if and only if the data packet could potentially get closer to its destination. More specifically, opportunistic routing translates to our scenario through the following rule: a boat A will forward a data packet to a boat B if and only if B has a PBS still to be visited in its displacement plan that is closer to Belém than those still to be visited in the displacement plan of A. A boat A may also forward a data packet to a PBS P, when there is another boat B that will stop by PBS P in the future and

whose displacement plan will get the packet closer to Belém than the displacement plan of A.

In summary, when an object A (boat or PBS) and a boat B establish a connection, and there is data on A, A will transfer as much data as possible to boat B during the connection time iff $Dist(DP_B, Belem) < Dist(DP_A, Belem)$, where DP_C indicates object C's displacement plan and the distance between the displacement plan of an object C and Belém, $Dist(DP_C, Belem)$, is equal to the minimum distance of a PBS in the displacement plan of C and a destination PBS in Belém. Similarly, a boat A will transfer data to a PBS P iff $Dist(DP_B, Belem) < Dist(DP_A, Belem)$, where B is a boat that will stop by PBS P later in time.

We propose to incorporate the use of fountain codes (in order to significantly enhance the robustness/reliability of the routing scheme while incurring just a linear overhead on the number of packets circulating in the network (rather than the exponential number of packets incurred by epidemic routing). My simulations results in the next section show that fountain codes can dramatically increase the performance of opportunistic routing under boat and packet transmission failures, keeping the successful delivery rate of the generated files at the remote communities close to 100% even at high boat delay/breakdown and packet loss probabilities. Fountain codes have been used in the literature to improve the reliability of many other systems, e.g. [48, 56, 47, 64, 4].

Figure 4.2 shows the whole process: how data packets are generated and encoded, and then transferred using our robust opportunistic routing. At first, the original data is partitioned into packets when generated, and then, before transferred to boats, the original data packets are encoded. When a boat stops by, the encoded packets are transferred during the life of the connection between PBS and boat. Boats carry the encoded packets and sail according to their displacement plans, a boat to boat

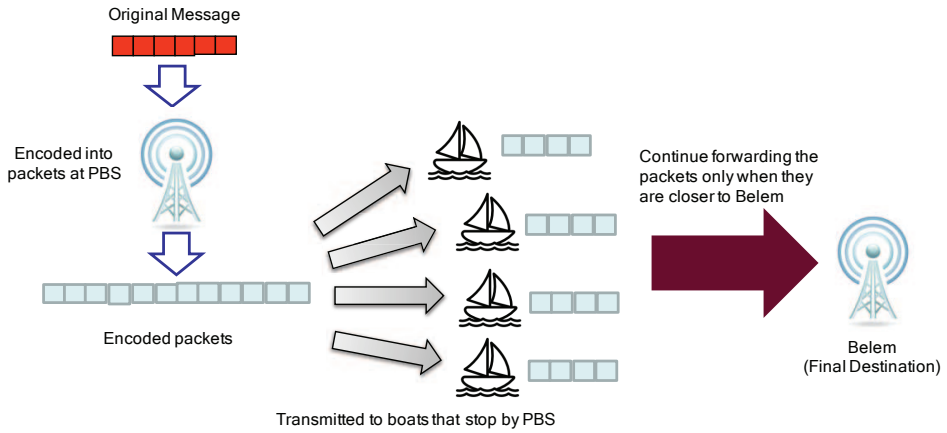

Figure 4.2: Overall Robust Routing process

transfer occurs only when the opportunistic routing rule is satisfied. When enough encoded packets are received by the healthcare facility in Belém, the original data file can be recovered.

4.6 Experiments

Boats are equipped with Wi-Fi (IEEE 802.11) devices and they follow predefined routes throughout the simulation area. Each boat has scheduled stopovers at various PBSs.

During sporadic government health programs in the Marajó Archipelago, physicians perform 100 ultrasound exams in one weekend, once a month. In other words, working at maximum capacity, they perform 50 exams per day. If we consider the size of each ultrasound file in MPEG format, and we add the high quality video of a health agent performing the examination (so the physician at Belém can see if the technician performed the exam correctly), the total size of an ultrasound examination record is around 75 Megabytes. All PBSs besides Belém, located at the extreme points of the

Marajó archipelago, were selected as sources of video transmissions, sending files to the main PBS in Belém. The average boat speed in the region is 15 Km/h (i.e., 9.32 mph).

We emulate the scenario above in our experiments by creating 50 files uniformly distributed across all PBSs except for Belém. These files are created during the first 20 hours of the total simulation time (of 24 hours), so that files generated towards the end of the simulation still have enough time to reach Belém. Here, a file travels from source to destination by hopping from boat to boat, boat to PBS or PBS to boat, following an opportunistic routing model. The simulation runs under two different modes: the *normal mode* and the *fountain code mode*. In the *normal mode*, we partition a file directly into 133 packets with all packets belonging to a file being generated at the same PBS. A file is transmitted successfully if all 133 packets are received at the final destination. Under the *fountain code mode*, we generate more than 133 packets per file (packets are of the same size as in normal mode, as well as the file size, and all packets belonging to the same file are generated at the same PBS), generating different levels of redundancy in the code. Namely, we consider three different codes, that generate 161, 175, and 205 packets per file respectively. For all three fountain codes considered, an arbitrary subset of only 144 fountain code packets from a file are required to be delivered to the final destination for the file to be successfully received, following [48]. Other simulation parameters are also described in Table 4.1

4.6.1 Transmission Affected by Rain

When a boat comes in contact with another boat or PBS, transmission speeds may be affected by heavy rain, which occurs frequently in the Amazon region. We

Table 4.1: Simulation parameters

Parameter	Value	Alt. Value	Alt. Value	Alt. Value
Number of files	50	100	200	400
Generation interval (seconds)	171	342	684	13680
File size (MB)	75	150	225	300
Transmission success probability	100%	70%	50%	-
Routing scheme	Epidemic	Opportunistic	Opportunistic W/ Partition	-
Number of fishing boats	5	10	20	-
Number of fountain codes partitions	161	175	205	-
Duration of boat staying at PBS (min)	5	20	-	-

Table 4.2: Fixed parameters

Parameter	Value
Number of PBSs	9
Number of original boats	13
Transmission speed (Mbps)	30
Transmission range (m)	40
Buffer Size	Infinity
Probability of losing packets when boat is delayed	50%

assign a rain probability of 50% at any point in time[2] and, if rain occurs, we assume that transmission speeds are reduced to $9Mbps$, 30% of the original value of $30Mbps$.

4.6.2 Transmission Loss

Transmission Loss is the loss probability associated with every data packet transmission between two nodes where each data packet transmission is treated as an independent event. We perform simulations with transmission loss probabilities of 0%, 2%, 5%, 10%, 20% and 30%. With an increase in the probability of transmission loss, there is a decrease in the number of data packets reaching the final destination, as would be expected, since many packets are lost on the way. As the count of successfully delivered packets decreases, the number of files successfully received at the final destination keeps on declining, as can be seen in Figure 4.3. It can be seen that only about 4.8% of files are received for normal mode with only 2% transmission loss probability. The percentage of successful files received is further reduced to 0%, with

[2]Variable rain probabilities had a negligible effect on the simulation and hence we decided to fix the rain probability at 50%

5% transmission loss probability. In fountain code mode, although the percentage of successfully received files decreases as the transmission loss probability increases, there is quite a bit of improvement compared to normal mode. As it appears in Figure 4.3, we are able to successfully transfer all 50 files when we use the fountain codes with a total of 175 or 205 packets with 2% transmission loss. We are still able to transfer about 95% of files successfully, even if we increase transmission loss probability to 5%. It may seem that dividing a file into more and more number of packets will increase the total number of successful files delivered to final destination, making them independent of transmission loss; at the same time, as the number of packets increase the network becomes more congested, leading to fall in percentage of successful files received at destination. This result may be more prominent in Figure 4.4, where the percentage of files successfully received increases with increase in transmission loss percentage, as an increase in transmission loss leads to reduced congestion in the network. This is further explained in the following section.

4.6.3 Boat Breakdown and Delay

A mechanical failure may stop the movement of a boat temporarily or permanently: such an event is termed as boat delay or boat breakdown respectively. In a boat breakdown the boat loses all its data packets, and suspends its movement until the end of the simulation, whereas in a boat-delay event, a boat stops for few hours (less than 24 hours) before resuming movement again on its path. A boat delay does not involve any loss of data but reduces the stoppage time (duration for which a boat stops at a PBS), for all successive PBSs, to 5 minutes from the usual 20 minutes. Every time a boat comes in contact with a PBS, we first check for a mechanical failure event, which occurs with 20% probability independently at each PBS. If a mechanical failure occurs, we then check if it corresponds to a boat-breakdown or a

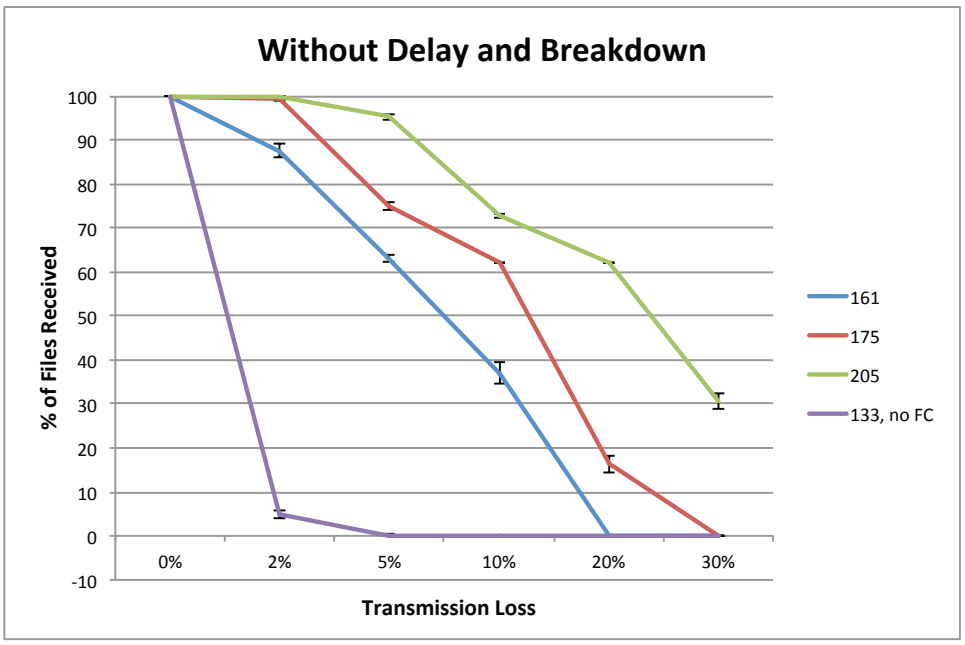

Figure 4.3: Percentage of files successfully received at final destination, for different number of generated packets with varying transmission loss probability, with 95% confidence intervals.

boat-delay event, which occur with 50% probability each. As expected for the same transmission loss probabilities, fewer number of files reach the final destination under the mechanical failure stochastic process (Figure 4.4), than with no chances of mechanical failure (Figure 4.3). An interesting observation can be made in Figure 4.4, where the percentage of successful number of files transferred increases initially as the transmission loss probability increases from zero to two percent; this might be due to the fact that the amount of data to be transfered is beyond the network capacity and in that case, since we are using fountain codes, randomly dropping packets can actually help.

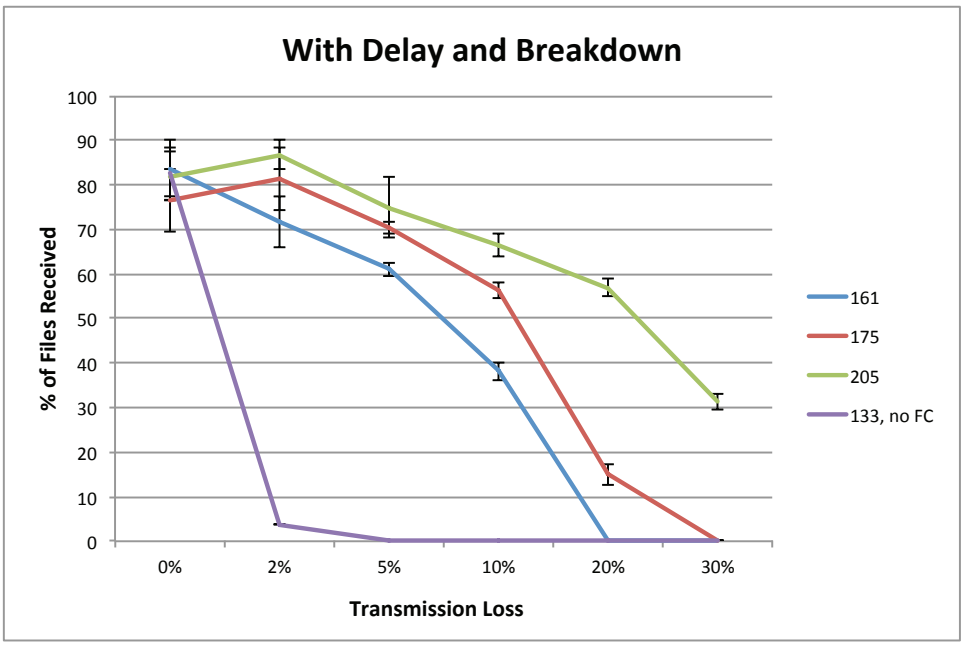

Figure 4.4: Percentage of files successfully received at final destination, for different number of generated packets with varying transmission loss probability and boats being delayed and broken, with 95% confidence intervals.

4.7 Conclusion

Given the significant challenge of providing healthcare to the remote and isolated communities in the Brazilian Amazon, we proposed a robust opportunistic routing algorithm, which takes advantage of using fountain codes to improve the delivery probability of the healthcare data in the presence of failures and unpredictable boat schedules. The proposed routing algorithm considers the probability of boats being delayed or breaking down, as well as the probability of loss in individual data packet transmissions due to the harsh environment in the region. Extensive experimental results show that the delivery probability of the healthcare data is significantly increased when using our robust opportunistic routing.

Furthermore, we could alternatively view this problem as a multicommodity flow

problem on the connection graph, where one would like to maximize the number of files (commodities) that are successfully transmitted from source to destination. In Chapter 5, we will further explore this "all-or-nothing" variation of multicommodity flow problem and present some further simulation results under this light in the Amazon scenario.

Chapter 5

A CONSTANT APPROXIMATION FOR MAXIMUM THROUGHPUT ROUTING

5.1 Introduction

The admission and allocation of multiple commodities in a graph is among the most fundamental and intensively studied algorithmic problems in networking. It is well-known that while for a single source-destination pair (s, t), the maximum possible flow from s to t is equal to the value of a minimum $s - t$ cut, this no longer holds for three or more commodities. While over the last decades, much progress has been made, some basic questions related to the maximization of the throughput achieved by multiple commodities in a network remain unresolved.

This work considers the following fundamental maximum throughput routing problem: given a set of k (splittable) multicommodity flows, select and route a subset of flows (all-or-nothing) such that throughput (the total number of satisfied commodities) in the n-node network is maximized. The main contribution of this work is a constant approximation algorithm for this problem (independent of k and n), based on randomized rounding. We also discuss two case studies, which indicate that the proposed algorithm can perform well in practice.

5.2 Contributions

In [46], we present a polynomial $(\alpha, \beta) - approximation$ algorithm for the ANF problem, which achieves a *constant* approximation ratio α for network throughput, and β is bounded by $O(\sqrt{(log(|V|) \cdot k)})$. Unlike the results in Chekuri et al. [15], where they present an approximation algorithm for the ANF problem with constant

β and α in $\Omega(\frac{1}{log^3|V|loglog|V|})$, our approach keeps α constant and allows β not to be constant. The Chekuri et al. approach [15] does not trivially imply the $(\alpha, \beta) -$ *approximation* trade off: It is not clear how one would modify their approach to ensure a constant approximation on the throughput while only violating edge capacity constraints by a constant.

Another main contribution of this work is to validate the proposed algorithm through extensive simulations on two case studies: First, we apply the algorithm through a Delay-tolerant network (DTN) that can be established via the actual passenger boats that serve the Amazon delta riverine scenario in Brazil; Second, we apply the proposed algorithm to the synthetic German50 benchmark network [53]. The simulation results show that, in practice our algorithms perform significantly better than the worst-case bounds suggest for the edge capacity violations, and only violate the edge capacities of the networks considered by less than a factor of 2 on average and less than a factor of $O(\sqrt{(log(|V|) \cdot k)})$ in the worst-case.

5.3 The All-or-Nothing (Splittable) Multicommodity Flow Problem

We model a flow network as a capacitated directed graph $G(V, E)$, where V is the set of nodes and E is the set of edges, and where each edge e has a given capacity $c(e) > 0$. We are also given a set of k commodities $\mathcal{F} = \{F_1, ..., F_k\}$, each with equal demand d. Each commodity $F_i \in \mathcal{F}$ is denoted by a pair (s_i, t_i) where s_i, t_i denote the source and destination for that commodity. Commodity F_i is satisfied if d units of this commodity can be successfully routed in the network. In the *All-or-Nothing (Splittable) Multicommodity Flow (ANF) problem*, one would like to maximize the total throughput in the network, where the throughput is measured in terms of the total number of commodities that are concurrently satisfied in a valid multicommodity flow. Note that we do not insist that the flow for a commodity F_i be non-splittable

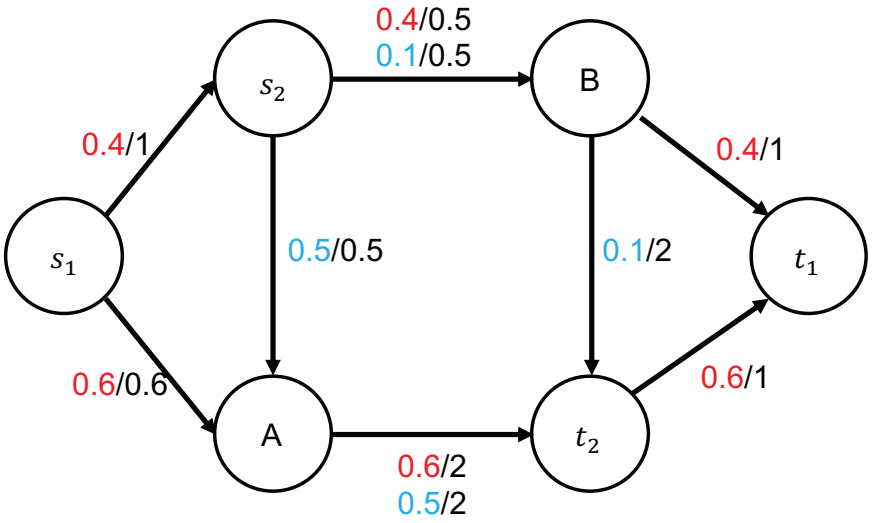

Figure 5.1: Example of The All-or-Nothing (Splittable) Multicommodity Flow Problem

or even that the flow on each edge be integral. The only assumption we make is that, for any i, the network has enough capacity to route d units of flow from s_i to t_i, if we were to consider commodity F_i alone in the network. The load of an edge e is equal to $\sum_i f_{i,e}$, where $f_{i,e}$ is the flow for commodity i on edge e.

Figure 5.1 shows an example of such scenario. Here, there are two pairs of source and destination, namely (s_1, t_1) and (s_2, t_2). Each pair has a unit demand, and each edge has a capacity noted by the number after forward slash. The number before the capacity is the commodity flow that passes through the edge. We use red to denote the flow for pair (s_1, t_1) and blue for (s_2, t_2). We can see that commodity 1 is fully delivered since t_1 received the full demand, however, t_2 only receives a fraction of 0.6 of the demand, which is considered a delivery failure.

We seek to find a polynomial time $(\alpha, \beta) - approximation$ algorithm for the ANF problem, for constants $0 \leq \alpha < 1$ and $\beta \geq 1$: Namely, we seek to find a solution for the ANF problem such that the throughput is at least an α fraction of the maximum

throughput, and the maximum load on any edge is at most β times the edge capacity, with high probability[1].

5.4 Randomized Rounding and Analysis

This section describes and analyzes our approximation algorithm as detailed in Algorithm 1 for maximizing the total number of commodities that can be successfully delivered. The algorithm is based on randomized rounding,

We use a variable f_i to represent whether or not F_i is successfully delivered. For the sake of simplicity, we rescale each commodity size and edge capacity in the network by s, so that this problem becomes a unit flow multi-commodity problem.

Inputs: directed graph $G(V, E)$

commodities $\mathcal{F} = \{F_1, ..., F_k\}$

source-sink pairs $(s_i, t_i) \forall F_i \in \mathcal{F}$

capacities $c(u, v) \forall (u, v) \in E$

flow variables $f_{i,(u,v)} \forall F_i \in \mathcal{F}, (u, v) \in E, and$

$f_i, \forall F_i \in \mathcal{F}$

[1]With probability at least $1 - \frac{1}{n^c}$, where $c > 0$ is a constant.

$$\text{Maximize } \sum_{i=1}^{k} f_i$$

Subject to

$$\sum_{(s_i,v)\in E} f_{i,(s_i,v)} = f_i, \qquad \forall F_i \in \mathcal{F}$$

$$\sum_{(u,v)\in E} f_{i,(u,v)} = \sum_{(v,u)\in E} f_{i,(v,u)}, \quad \forall F_i \in \mathcal{F}, \forall v \in V - \{s_i, t_i\}$$

$$\sum_{i=1}^{k} f_{i,(u,v)} \leq c_{(u,v)}, \qquad \forall (u,v) \in E$$

$$f_{i,(u,v)} \geq 0, \qquad \forall F_i \in \mathcal{F}, \forall (u,v) \in E$$

$$f_{i,(u,v)} \leq f_i \cdot c_{(u,v)}, \qquad \forall F_i \in \mathcal{F}, \forall (u,v) \in E$$

$$f_i \in \{0,1\}, \qquad \forall F_i \in \mathcal{F}$$

Let OPT_i be the optimal solution of the integer formulation, and let OPT_f be the total amount of commodities delivered by solving the linear relaxation LP where the variables f_i are relaxed to assume any value in $[0,1]$. It is obvious that $OPT_f = \sum \tilde{f}_i$. Define the solution from the above Algorithm 1 as ALG, and the total amount of commodities delivered by ALG as OPT_{ALG}.

We use Chernoff over continuous random variables to bound the probability of achieving a fraction of the optimal solution.

Fact 5.4.1 (Chernoff Bound) *[50] Let $X = \sum_{i=1}^{n} X_i$ be a sum of n independent random variables $X_i \in [0,1], 1 \leq i \leq n$. Then $P(X < (1-\epsilon) \cdot E(X)) \leq exp(-\epsilon^2 \cdot E(X)/2)$ holds for $0 < \epsilon < 1$.*

Since we have already scaled down the flow by s, the variables are between 0 and 1, and we can apply the above Chernoff bound.

Algorithm 1 Randomized Rounding Algorithm for Formulation 1

Input:

Directed Connection Graph $G(V, E)$;

Commodities $\mathcal{F} = \{F_1, ..., F_k\}$;

Source-sink pair (s_i, t_i) for each $F_i \in \mathcal{F}$;

Capacity $c(u, v),\ \forall (u, v) \in E$;

Output: OPT_{ALG}

1: Change the last constraint to be $0 \leq f_i \leq 1$.

2: Relaxation to LP, solve this LP, obtain optimal solution $\tilde{f}_i$.

3: With probability $\tilde{f}_i$, set $f_i = 1$, otherwise set it to 0.

4: Scale up the fractional flow $\tilde{f_{i,e}}$ from the LP solution on edge e for commodity i by $\frac{1}{\tilde{f}_i}$, i.e., $f_{i,e} = \tilde{f_{i,e}} \times \frac{1}{\tilde{f}_i}$, for i s.t. $f_i = 1$.

5: If the solution is within a certain fraction of the optimal solution, return this solution; otherwise, repeat step 3 and 4, at most $\theta(\log |V|)$ times.

Claim 5.4.2 $Pr[OPT_{ALG} < (1 - \epsilon) \cdot OPT_f] \leq e^{-\epsilon^2 \cdot OPT_f/2}$

Proof For each commodity i, the expectation of f_i is $E(f_i) = 1 \cdot \tilde{f}_i + 0 \cdot (1 - \tilde{f}_i) = \tilde{f}_i$. Recall that $OPT_f = \sum \tilde{f}_i$, and let $OPT_{ALG} = \sum f_i$.

$$\begin{aligned} & Pr[OPT_{ALG} < (1 - \epsilon) \cdot OPT_f] \qquad (5.1) \\ & = Pr[\sum f_i < (1 - \epsilon) \cdot OPT_f] \\ & \leq e^{-\epsilon^2 \cdot OPT_f/2} \qquad \square \end{aligned}$$

Since we also assume that all commodities can be fully fractionally delivered alone, we have:

$$OPT_f \geq 1 \qquad (5.2)$$

Since we have proved that the optimal fractional solution is an upper bound on optimal solution for the ILP problem, by using $\epsilon = 2/3$ in the Chernoff bound, we get:

Theorem 5.4.3 *The probability of achieving less than 1/3 of the profit of an optimal solution is upper bounded by* $e^{-2/9} \approx 0.8007$.

Proof By taking $\epsilon = 2/3$, Equation 5.1 becomes:

$$Pr[OPT_{ALG} < \frac{1}{3} \cdot OPT_f] \leq e^{-(\frac{2}{3})^2 \cdot OPT_f/2} \quad (5.3)$$

$$= e^{-2 \cdot OPT_f/9} \quad (5.4)$$

By Equation 5.2, we know that the minimum value of OPT_f is 1. And by taking $OPT_f = 1$, we get the upper bound of $e^{-\epsilon^2 \cdot OPT_f/2}$. Therefore:

$$Pr[OPT_{ALG} < \frac{1}{3} \cdot OPT_f] \leq e^{-2/9} \quad (5.5)$$

Since $OPT_i \leq OPT_f$, thus we get,

$$Pr[OPT_{ALG} < \frac{1}{3} \cdot OPT_i] \leq e^{-2/9} \square$$

Fact 5.4.4 (Hoeffding's Inequality) *[24] Let* $\{X_i\}$ *be independent random variables, s.t.* $X_i \in [a_i, b_i]$, *then* $Pr(\sum_i X_i - E(\sum_i X_i) \geq t) \leq exp(-2t^2/\sum_i (b_i - a_i)^2)$ *holds.*

Theorem 5.4.5 *Given a single edge e with capacity* c_e. *Let* $\Delta_{F,e}$ *be the commodities going through edge e, thus* $|\Delta_{F,e}| \leq k$. *For all commodities* $i \in \Delta_{F,e}$, *choose* ϵ' *such that* $\max \frac{\tilde{f_{i,e}}}{\tilde{f_i}} \leq \epsilon' \cdot c_e$. *The probability that ALG exceeds the edge capacity constraint by a factor of* $\gamma = (1 + \epsilon' \cdot \sqrt{2 \log |V||\Delta_{F,e}|})$ *is bounded by* $|V|^{-4}$.

We analyze the probability that our algorithm violates capacity constraints by a certain factor, by employing Hoeffding's Inequality.

Proof Fix an edge $e \in E$, for commodity i. With probability $1 - \tilde{f}_i$, the flow on edge e for commodity i is set to 0, i.e., $f_{i,e} = 0$. With probability $\tilde{f}_i$, the flow on edge e for commodity i is set to $\tilde{f_{i,e}} \cdot \frac{1}{\tilde{f}_i}$.

Then the expectation of $f_{i,e}$ is

$$E(f_{i,e}) = \tilde{f_{i,e}} \cdot \frac{1}{\tilde{f}_i} \cdot \tilde{f}_i + 0 \cdot (1 - \tilde{f}_i) = \tilde{f_{i,e}} \tag{5.6}$$

Let F_e denote the flow on edge e induced by ALG. Then $F_e = \sum_{i, f_{i,e} \neq 0} f_{i,e}$ and the expectation of F_e is

$$E[F_e] = \sum_{i, f_{i,e} \neq 0} \tilde{f_{i,e}} \cdot \frac{1}{\tilde{f}_i} \cdot \tilde{f}_i = \sum_{i, f_{i,e} \neq 0} \tilde{f_{i,e}} \tag{5.7}$$

Since we relax the LP, and a feasible solution must obey edge capacity constraints, the cumulative load on edge e is equal or less than the capacity of e:

$$\sum_{i, f_{i,e} \neq 0} \tilde{f_{i,e}} \leq c_e \tag{5.8}$$

Therefore:

$$E[F_e] \leq c_e \tag{5.9}$$

Let $t = \epsilon' \cdot \sqrt{2 \log |V| |\Delta_{F,e}|} \cdot c_e$. By applying Hoeffding's Inequality and under the assumption of $\max \frac{\tilde{f_{i,e}}}{\tilde{f}_i} \leq \epsilon' \cdot c_e$, we get:

$$\begin{aligned} Pr[F_e - E(F_e) \geq t] &\leq exp(\frac{-2t^2}{\sum_i (\frac{\tilde{f_{i,e}}}{\tilde{f}_i})^2}) \\ &\leq exp(\frac{-2 \cdot \epsilon'^2 \cdot 2 \log |V| \cdot |\Delta_{F,e}| \cdot c_e^2}{\epsilon'^2 \cdot c_e^2 \cdot |\Delta_{F,e}|}) \\ &= |V|^{-4} \end{aligned} \tag{5.10}$$

Given Equation 5.9, let $\gamma = (1 + \epsilon' \cdot \sqrt{2 \log |V||\Delta_{F,e}|})$.

$$\begin{aligned} & Pr[F_e - c_e \geq \epsilon' \cdot \sqrt{2 \log |V||\Delta_{F,e}|}] \\ = \; & Pr[F_e \geq (1 + \epsilon' \cdot \sqrt{2 \log |V||\Delta_{F,e}|}) \cdot c_e] \\ = \; & Pr[F_e \geq \gamma \cdot c_e] \\ \leq \; & |V|^{-4} \end{aligned}$$ □

Since there are at most $|V|^2$ edges, and we know $|\Delta_{F,e}| \leq k$, by applying a union bound over all edges using Theorem 5.4.5 we obtain the following corollary.

Corollary 5.4.6 *The probability that ALG exceeds any of the edge capacity constraints by a factor of* $1 + \epsilon' \cdot \sqrt{2 \log |V| \cdot k}$ *is upper bounded by* $|V|^{-2}$.

Based on Theorem 5.4.3 and Corollary 5.4.6, if $|V| \geq 3$ holds, the probability of not finding a suitable solution, satisfying the objective and the capacity constraint, within a single round, is therefore upper bounded by $exp(-2/9) + 1/9 \leq 11/12$. The probability to find a suitable solution within $c \log |V|$ many rounds, where c is a constant, is then lower bounded by $1 - (11/12)^{c \log |V|} = 1 - \frac{1}{|V|^b}$, where b is a constant, for $|V| \geq 3$ and hence the randomized rounding scheme yields a solution with high probability.

Theorem 5.4.7 *The randomized rounding scheme yields a suitable* $(\alpha, \beta)-$*approximation solution within* $c \log |V|$ $(|V| \geq 3)$ *many runs where* c *is a constant and* $\alpha = \frac{1}{3}$, $\beta = 1 + \epsilon' \cdot \sqrt{2 \log |V| \cdot k}$ *with high probability.*

5.5 Application in Delay Tolerant Networks

A Delay Tolerant Network (DTN) is widely seen in many applications, such as [43], that shows the feasibility of leveraging boats as data mule nodes to carry medical

ultrasound videos from remote and isolated communities in the Amazon region in Brazil, to the main city of that area. Such videos are needed by physicians to perform remote analysis and follow-up routine of prenatal examinations of pregnant women. Other work can also be found in the context of IoT [9, 3] or mobile social networks [32, 49, 51].

However, since DTNs are dynamic and time evolving in nature, it is necessary to transform it into a static format that enables us to apply our proposed approximation method.

5.6 Case Study 1: Riverine Amazon Region

5.6.1 Scenario

The network described in Chapter 3 can be used an application scenario to evaluate our algorithm. Such network is time evolving and hence does not fit our model directly. However, we can convert it into a static graph using the connection graph model presented in Section 4.4, preserving the time precedence relations of the connections, as discussed in Section 4.4.

In the simulation setup, we use real data schedules for the regular passenger boats serving the Amazon Delta Region in the state of Pará in Brazil 43. We randomly generate 50 files, with a time interval of 60 seconds in between files, hence each file comes with a generation timestamp. In the connection graph, the commodity nodes are the files in $\mathcal{F}$, and connection nodes are the nodes in $\mathcal{C}$. For each commodity, we create a super sink, and connect it to all the connection nodes, that could potentially deliver the commodity to its destination. The connection graph has 449 connection nodes, 50 commodity nodes with 50 super sink nodes. Total number of edges is around 16000 and the edge capacity varies from 200 to 400000 due to different time

Commodity Size	RR	LP
1000	50	50
3000	50	49.96
5000	47.5	47.51
7000	42	42.13
9000	38.2	38.31

Table 5.1: Average number of commodities received for different commodity sizes

schedules.

5.6.2 Experimental Results

We first solve the LP formulation and then use the randomized rounding to select the commodities to deliver and then check the edge constraints to see if any edge constraint is violated. We repeat the randomized rounding process 100 times, and report the following statistics: the $(\alpha, \beta)-$distribution and the $(\alpha, \beta)-$distance as we explain later. Note that according to the theoretical bounds given by Corollary 5.4.6, the violation ratio can be proportional to $\sqrt{2 \cdot \log |V| \cdot k}$, which is roughly 25.1.

The default setting for commodity size we consider is 7000MB. We also test with different commodity sizes to find out how the proposed algorithm performs under various settings. Each setting is executed for 10 times and the average commodity delivery is reported. Figure 5.6 and Table 5.1 shows the comparison between the number of commodities delivered using our proposed randomized rounding algorithm (RR) and Linear Programming (LP). As we can see, even though different commodity sizes result in different numbers of commodities received, the ratio stays around 1 which means that the RR solution is basically as competitive as the LP solution.

Figure 5.2 shows the edge loads when commodity size is 7000MB. As we can see

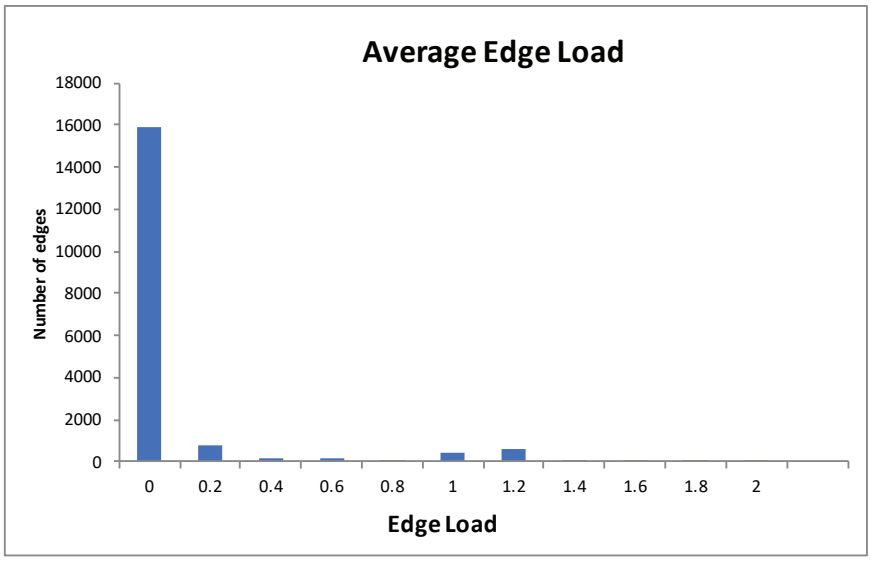

Figure 5.2: Edge Load for Commodity Size 7000MB in Amazon Scenario

from the figure, most of the edges have only a small load, and few edges have moderate load, are fully saturated, or violate their capacities (see Figure 5.2). We observe that some connections are not used for delivering commodities and some connections on the main path are used for delivery of many commodities.

Figure 5.3 shows the edge capacity violations when the commodity size is 7000MB. We see that among the edges that violate their capacities, most slightly violate the capacity constraints by at most 10%, while only few have higher violations, but none by more than 30% of their capacities.

Figure 5.4 shows the edge load information for the run with the lowest edge capacity violation ratios among our 100 simulation runs for a commodity size of 7000MB. Recall that our algorithm always picks the best out of $\theta(\log |V|)$ runs. As we can see, only a few edges have a violation at 1.2x and most of edges are either not used or not fully loaded.

Figure 5.5 shows the histogram of number of files delivered for the 100 runs of

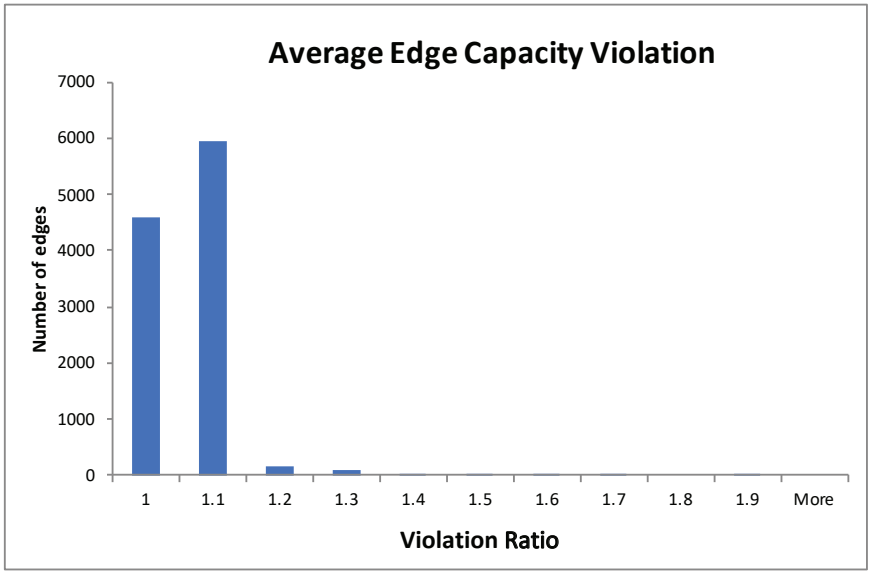

Figure 5.3: Edge Capacity Violations for Commodity Size 7000MB in Amazon Scenario

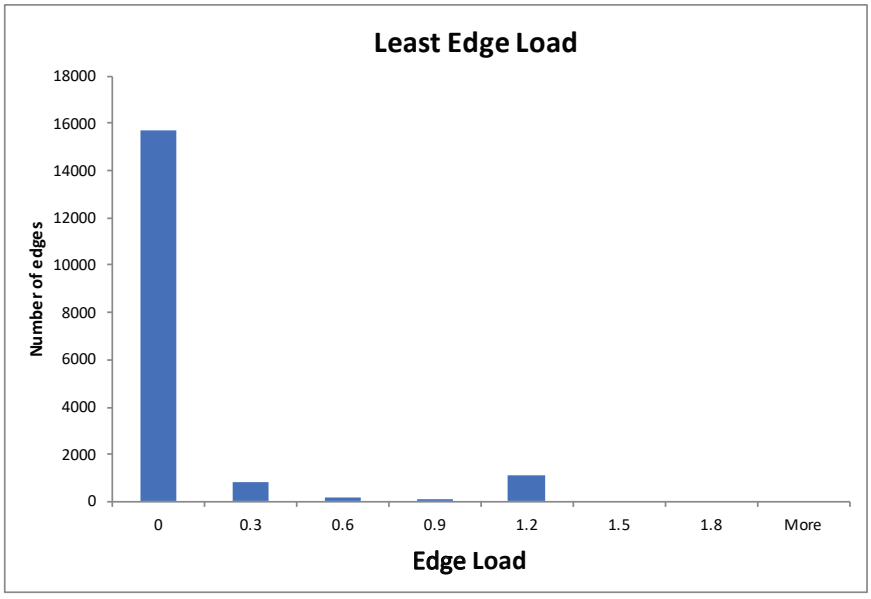

Figure 5.4: Least Edge Load for Commodity Size 7000MB in Amazon Scenario

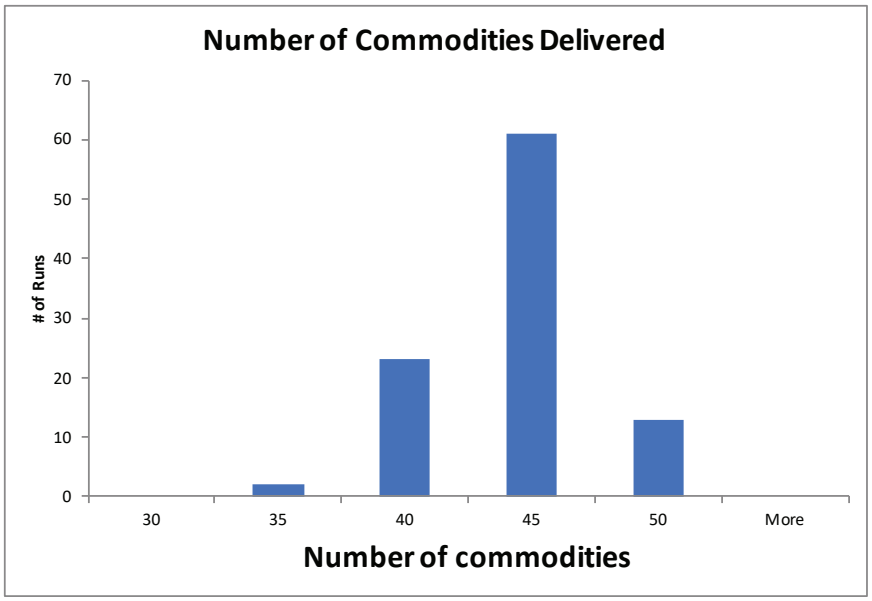

Figure 5.5: Number of Files Delivered after Randomized Rounding for Commodity Size 7000MB in Amazon Scenario

Amazon scenario. The solution that the LP returns is 48.4. Since we allow edge capacity violations and round up the solutions during the randomized rounding process, for some cases we will observe that the objective function value will be higher than the LP solution.

Figure 5.6 shows the comparison between the number of commodities delivered using our proposed randomized rounding algorithm (RR) and Linear Programming (LP), incorporating the results in Table 5.1. As we can see, even though different commodity sizes result in different numbers of commodities received, the ratio stays around 1 which means that the RR solution is basically as competitive as the LP solution.

Figure 5.7 shows the $(\alpha, \beta)-$distribution of the 100 runs for Amazon Scenario. As we can see, most runs have a competitive throughput ratio α at least 0.8. In terms of edge capacity violation ratio β, most runs have a ratio between 1 and 2.

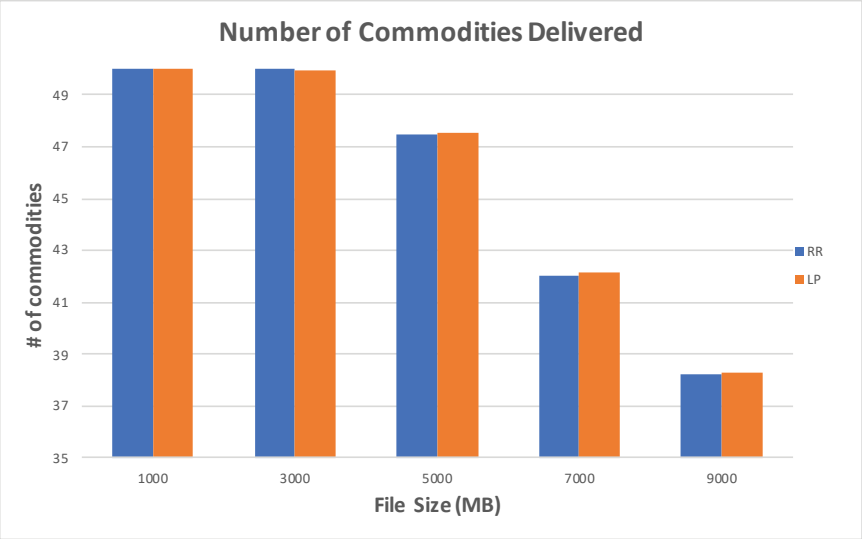

Figure 5.6: RR and LP comparison in terms of commodity delivery in Amazon Scenario

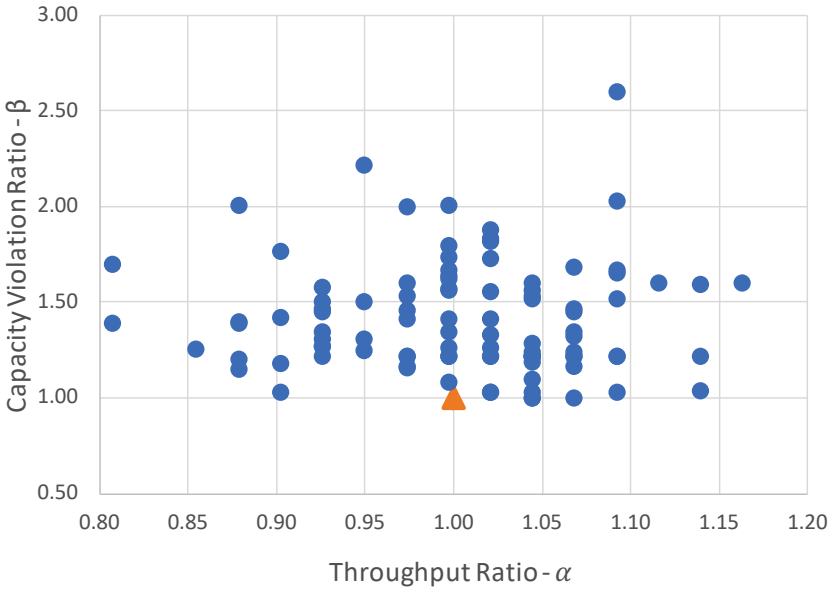

Figure 5.7: $(\alpha, \beta)-$distribution for Commodity Size 7000MB in Amazon Scenario

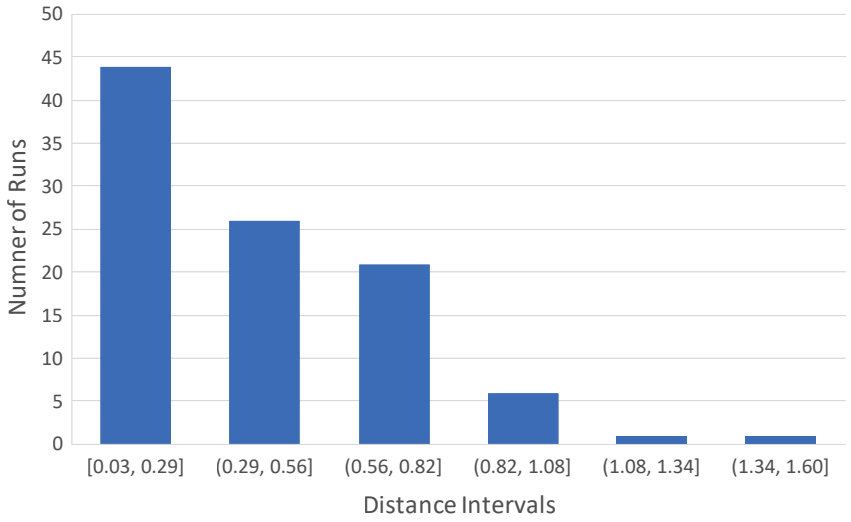

Figure 5.8: $(\alpha, \beta)-$distances for Commodity Size 7000MB in Amazon Scenario

Given the $(\alpha, \beta)-$distribution shown in Figure 5.7, we calculate the Euclidean distances among each (α, β) point with respect to the optimal point, which is given by $\alpha = 1, \beta = 1$, and plot the distribution of these distances in Figure 5.8. As we can see, most runs are within a small distance (0.82) to the optimal solution.

While in some applications, a capacity violation for an edge e may mean that you need to pay a penalty or additional charges for using a higher bandwidth than what was reserved with your network provider for the edge e, in some applications of the ANF problem, the capacity of an edge may pose a strict requirement (e.g., the cross section of a pipe, or the capacity of a highway) and cannot be violated. Hence we run some additional simulations where we scale down the capacity of each edge in the Amazon network by a factor of 2.5, since the original experiments (Figure 5.7) show that 95% of the runs have edge capacity violation ratios lower than 2.5.

Figure 5.9 shows the $(\alpha, \beta)-$distribution of the 100 runs for Amazon Scenario after scaling down the edge capacities. The best run achieves throughput ratio $\alpha = 0.95$ and edge violation ratio $\beta = 0.72$, which means the proposed randomized rounding

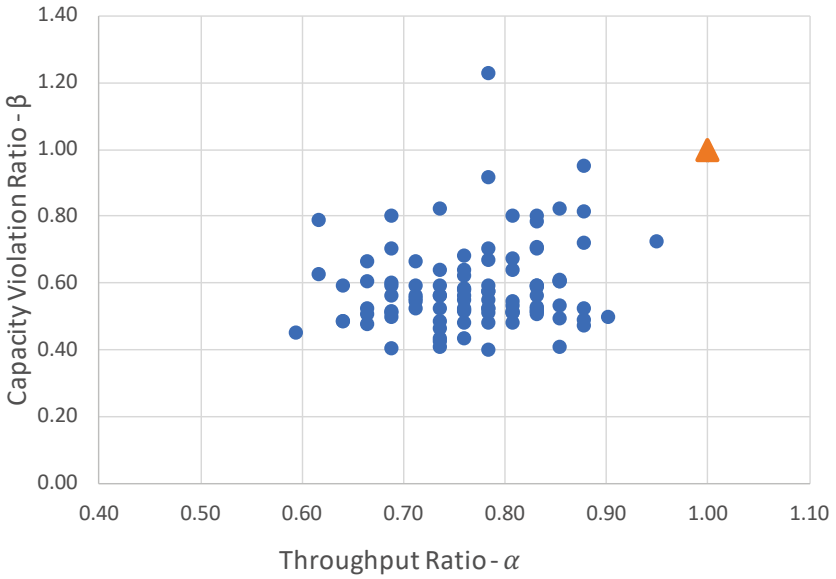

Figure 5.9: $(\alpha, \beta)-$distribution for Commodity Size 7000MB in Amazon Scenario after scaling down edge capacities

algorithm finds a feasible solution that is competitive with optimal solution found by the LP without incurring any edge capacity violations.

Figure 5.10 shows the $(\alpha, \beta)-$distances given the $(\alpha, \beta)-$distribution shown in Figure 5.9, we calculate the Euclidean distances among each (α, β) point with respect to the optimal point, which is $\alpha = 1, \beta = 1$, and plot the distribution of these distances. As we can see, most runs are within a small distance (0.59) to the optimal solution.

5.7 Case Study 2: German50 Network

In this section, we will discuss how the proposed method performs on the synthetic data obtained from [53].

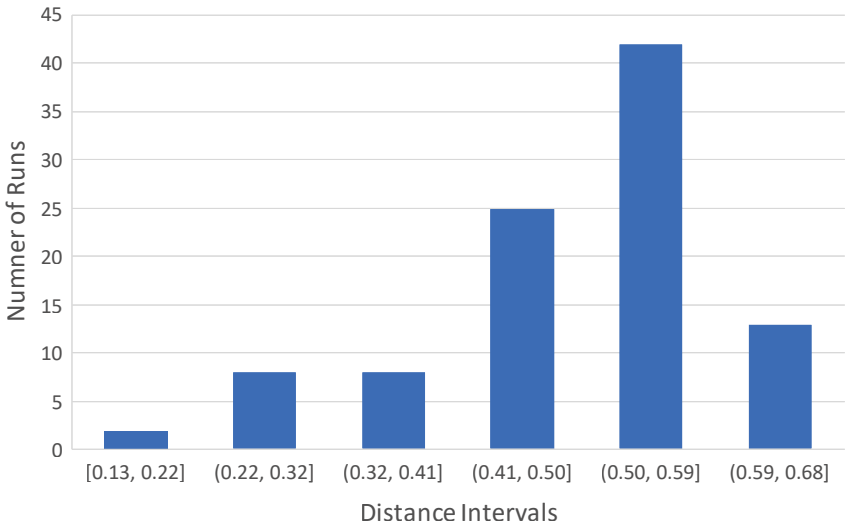

Figure 5.10: $(\alpha, \beta)-$distances for Commodity Size 7000MB in Amazon Scenario after scaling down edge capacities

5.7.1 Scenario

We use the testbed Germany50 network from SNDlib [53]: In this network, there are 50 nodes and 88 edges, and number of commodities is 662. The average node degree is 3.52. The edge capacity of each edge is 40, and in order to test the proposed method, we set the commodity size to be 50.

5.7.2 Experimental Results

Again, we repeat the randomized rounding process 100 times, and report the following statistics: the $(\alpha, \beta)-$distribution and the $(\alpha, \beta)-$distance. Note that according to the theoretical bounds given by Corollary 5.4.6, the violation ratio can be proportional to $\sqrt{2 \cdot \log |V| \cdot k}$, which is roughly 71.9.

Figure 5.11 shows the histogram of average edge capacity violation for 100 runs. The x axis shows that the number of times the capacity is violated. As we can see from the histogram, most of the violations are under 1.5 and very few are above 2.

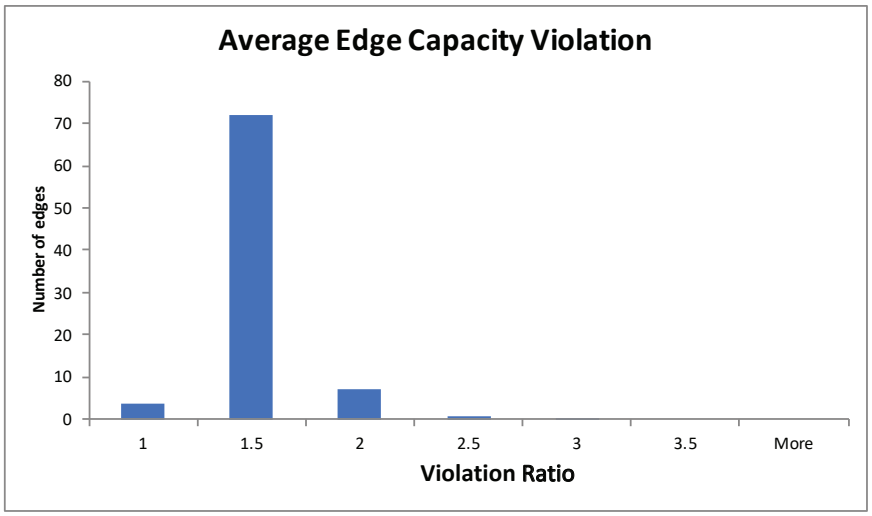

Figure 5.11: Average Edge Capacity Violation in German 50 Scenario

This means that even though some of the edges did violate the capacity constraints, but it is much smaller compared to the assumed violation ratio.

Figure 5.12 shows the histogram of the number of files that are delivered after the randomized rounding process, the optimal solution returned by the LP solver is 70.4, and we can see here roughly half of the runs, the randomized rounded solution is better than LP optimal solution with edge capacity violation.

Figure 5.13 shows the lowest edge violation ratio among these 100 runs, The number of files delivered after randomized rounding process is 53, which is 17 less compared to the LP solution, which only has 53 violations and most of the violations happen at 1.1 and 1.3 times.

Note that the number of commodities is fixed in the German50 network, and hence, we cannot evaluate the impact of varying the number of commodities on the algorithm's performance, as we did for the Amazon case study.

Figure 5.14 shows the $(\alpha, \beta)-$distribution of the 100 runs for German 50 Scenario before scaling down the edge capacities. As we can see that all runs have competitive

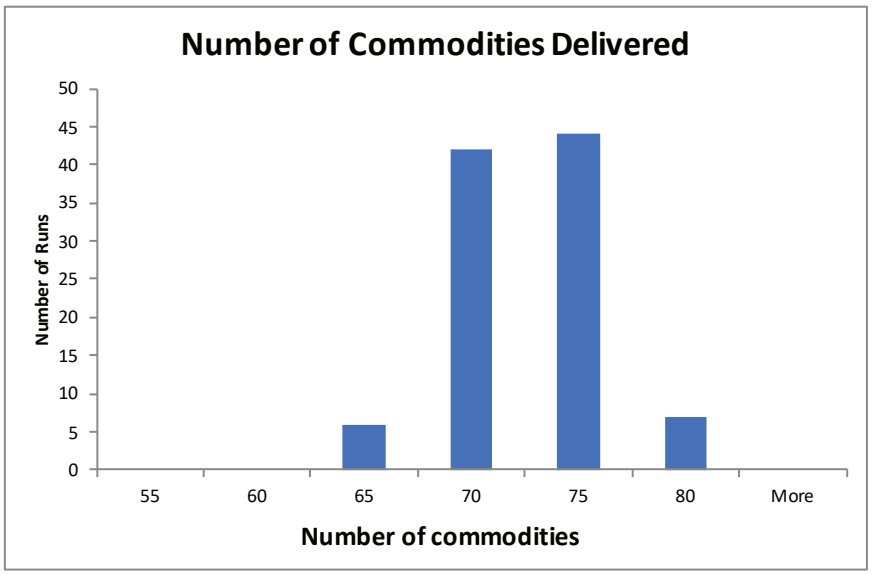

Figure 5.12: Number of Files Delivered after Randomized Rounding in German 50 Scenario

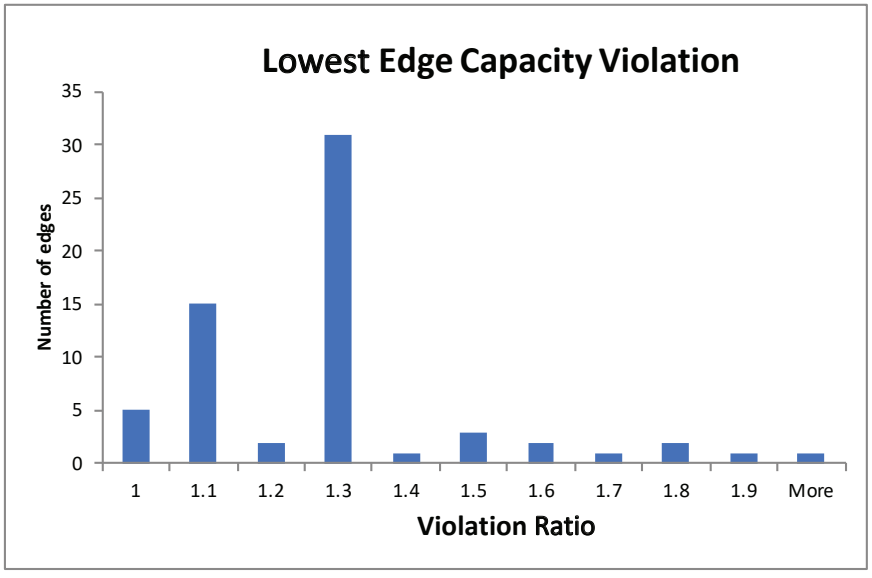

Figure 5.13: Least number of Edge Capacity Violation in German 50 Scenario

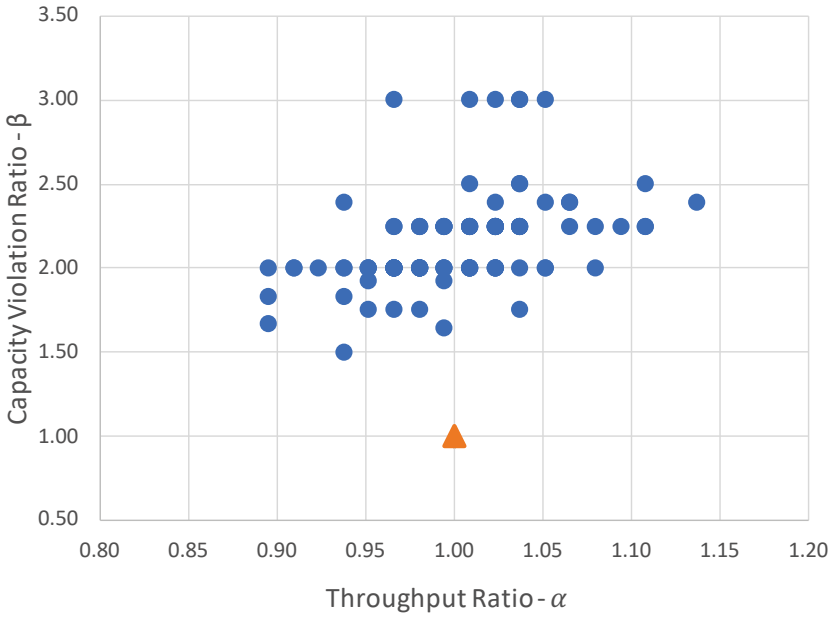

Figure 5.14: $(\alpha, \beta)-$distribution for German 50 Scenario

throughput ratio α at least 0.9, and most runs have an edge capacity violation ratio between 1.5 and 2.5, but no more than 3.

Figure 5.15 shows the $(\alpha, \beta)-$distances given the $(\alpha, \beta)-$distribution shown in Figure 5.14. As we can see, most runs are within a small distance 1.5 to the optimal solution.

As we did in Section 5.6.2, we also run experiments with scaled down edge capacities (by a factor of 2.5, since also here 90% of the original runs in Figure 5.14 have edge capacity violation ratios lower than 2.5) for the German50 network in order to obtain solutions with no edge capacity violations.

Figure 5.16 shows the $(\alpha, \beta)-$distribution of the 100 runs for German 50 Scenario after scaling down edge capacities. As we can see, the best run achieves throughput ratio $\alpha = 0.37$ and edge violation ratio $\beta = 0.89$. This is also aligned with our analysis where the randomized rounding algorithm is able to find a feasible solution with at least $\frac{1}{3}$ of the throughput of optimal LP solution without any edge capacity

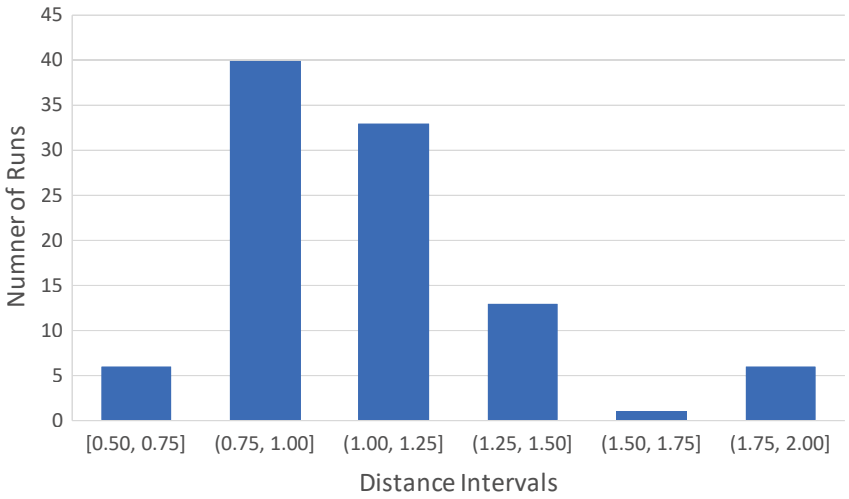

Figure 5.15: $(\alpha, \beta)-$distances for German 50 Scenario

violations.

Figure 5.17 shows the $(\alpha, \beta)-$distances given the $(\alpha, \beta)-$distribution shown in Figure 5.16. As we can see, most (α, β) points are within a small distance (0.81) to the optimal point.

5.8 Conclusions

We presented the first constant-approximation algorithm for the fundamental problem of maximizing throughput for all-or-nothing splittable commodities. We also showed, using simulations, that our algorithm features interesting properties in different case studies and when applied to scheduling and delay-tolerant networks where nodes have limited buffer size and short contact times. We believe that this work provides interesting avenues for future research. In particular, it would be interesting to further improve the trade off between approximation quality and augmentation and investigate lower bounds.

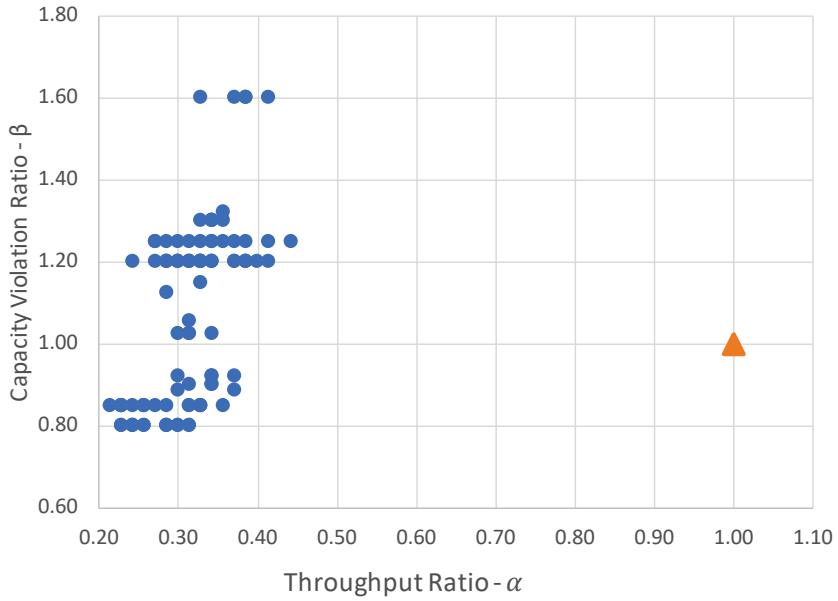

Figure 5.16: $(\alpha, \beta)-$distribution for German 50 Scenario after scaling down edge capacities

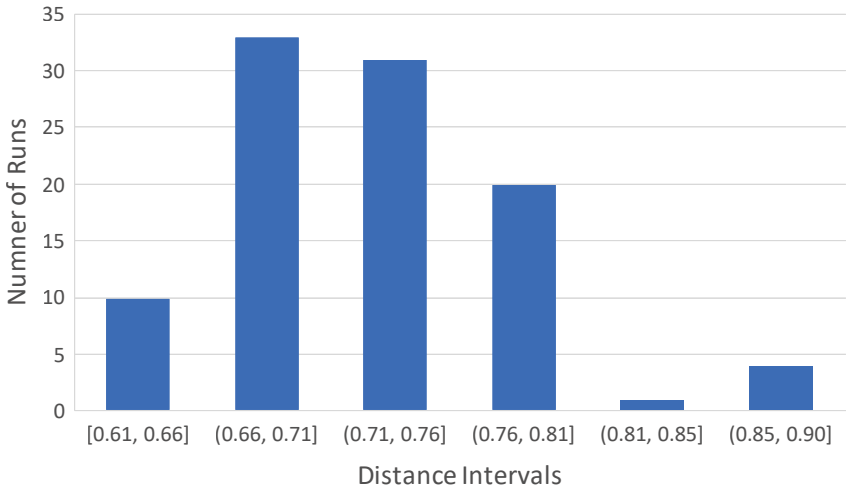

Figure 5.17: $(\alpha, \beta)-$distances for German 50 Scenario after scaling down edge capacities

Chapter 6

INTEREST- AND CONTENT-BASED DATA DISSEMINATION IN MOBILE SOCIAL NETWORKS

6.1 Introduction

With the increasing popularity of hand-held mobile devices such as smart phones and smart watches, people are connected more than ever, which enables the information to be created, forwarded, and exchanged at levels that one could not envision just a few years ago. Mobile social networks (MSNs) thrive with the popularity of mobile smart devices and exhibit the properties of social networks. To efficiently disseminate the information within MSNs, there have been research efforts on content-based routing schemes which rely on the network structure and user interest profiles. However, none of these prior works considered the impact of data content during data dissemination in MSNs. However such content is closely related to users' preferences, which have a significant influence on the result of the dissemination. To fill this void, we propose an interest- and content- based dissemination scheme in MSNs, where the contents of the messages, along with the network structural information are taken into consideration. In the proposed scheme, each user is associated with an interest profile and each message is associated with a message content profile. Similarities are measured between the users and the messages given their profiles. We also use PageRank to measure the importance of each user when the network evolves over time. Each piece of information is propagated based on the similarity scores and PageRank selection of relay users. We experimentally show that the proposed scheme achieves higher delivery performance compared to the existing schemes while

remaining cost-effective.

6.2 Contributions

Content-based routing schemes which rely on the network structure and user interest profiles have been studied in the literature [9, 22, 17, 32, 49, 51, 65, 68]. There are also various works studying the impact on personalized content [42]. However, these works did not consider the impact of message content during data dissemination. Since users' preferences on message content can have a significant influence on the result of the dissemination, we propose an interest- and content-based data dissemination scheme, that takes the message content as well as the network structure and user interests into consideration.

The main contributions of the proposed content-based routing scheme are:

- To the best of our knowledge, we are the first to consider integrating message content with the evolving network structure, user interest profiles, and message content profiles for data dissemination in MSNs.
- We propose a data dissemination scheme which combines the PageRank results with the similarity score, where the PageRank algorithm is run over the evolving structure of the MSN and the similarity score integrates the message content profile and user interest profile.

6.3 System Model and Problem Formulation

In this section, we present my system model for MSNs and formulate the problem.

6.3.1 System Model of MSN

Due to its evolving nature, the structure of an MSN is time-sensitive. We divide the whole time T into c non-overlapping time slots, i.e., $T = [t_1, t_2, ...t_c]$. For each

time $t \in T$, we use an undirected graph $G^t = (V^t, E^t)$ to represent the MSN at time t, where each node $v_i \in V^t$ represents one distinct mobile device user. An edge $e_{ij}^t = (v_i, v_j) \in E^t$ indicates the direct communication contact between two users $v_i \in V^t$ and $v_j \in V^t$ at time t. We define $n^t = |V^t|$ and $m^t = |E^t|$. Each user v_i is associated with a *user interest profile* I_i over some selected items or topics (*feature set*) to depict the preferences of v_i. The concepts of user interest profile and feature set will be explained later in this subsection.

There are d messages $\mathfrak{M} = \{M_1, ..., M_d\}$ generated in the MSN. Each message $M_k = (M_{k,1}, M_{k,2}, ...)$ can be written as the combination of several smaller messages, $M_{k,i}$, also called *items*. Each message M_k is generated at time t_{λ_k} by node v_{δ_k}, and should be propagated to as many nodes interested in some of the message contents as possible. We associate each message M_k with a *message content profile*, and compute the *content similarities* among the profiles (these two concepts will also be introduced later in the subsection). We use $R_k^{\leq t} \subseteq V$ to denote the set of nodes who have received the message M_k up to time t. $R_k^{\leq t}$ will then propagate M_k to a subset of their neighbors in the MSN at time $t+1$. At time t_{λ_k}, $R_k^{\leq t_{\lambda_k}} = \{v_{\delta_k}\}$. For each M_k, its *time to live* (TTL) is defined as the maximum time duration that M_k disseminates in the MSN. We use t_{ι_k} to denote the TTL of M_k. At time $t_{\lambda_k} + t_{\iota_k}$, the dissemination of M_k stops, and $R_k^{\leq t_{\lambda_k} + t_{\iota_k}}$ is the set of nodes that M_k has reached. we now introduce four key concepts: *feature set, user interest profile, message content profile*, and *content similarity*.

Feature Set: A Feature set is a set of features selected to depict the user interest and message content profiles. Let $\mathcal{F} = \{f_1, ..., f_p\}$ be a feature set, where each f_i denotes one feature that can be rated. Each feature can be materialized as a real item such as "fruit" or "football". It can also be a topic such as "Ad Hoc Mobile Networks" or "Data Mining".

User Interest Profile: For each node v_i, we create a *user interest profile* based on v_i's rating over the feature set $\mathcal{F}$. The user interest profile I_i is represented by a vector of ratings $(I_{i1}, ..., I_{ip})$, where $I_{ij} \in [0, 1]$ is user v_i's rating on feature f_j. We assume that each I_i is normalized, i.e., $\sum_{j=1}^{p} I_{ij} = 1$ for each $v_i \in V$. Table 6.1 provides an example of several user interest profiles. In a user interest profile, a larger score implies a higher preference towards the corresponding feature, while a low score indicates a dislike towards the feature.

Table 6.1: User Interest Profile for Each User

User ID	f_1	f_2	f_3	**...**
v_1	0.25	0.15	0.45	...
v_2	0.1	0.65	0.05	...
v_3	0.45	0.15	0.15	...
...	...	...	...	...

Message Content Profile: The message content profile depicts the relationship between a message and the feature set. We use $\mathcal{M}_{k,l}^{j} \in \{0, 1\}$ to represent the relevance between item $M_{k,l}$ and feature f_j. If they are related, $\mathcal{M}_{k,l}^{j} = 1$; and $\mathcal{M}_{k,l}^{j} = 0$ otherwise. The message content profile $\mathcal{M}_k$ for message M_k is defined as the matrix $(\mathcal{M}_{k,1}^{1}, \mathcal{M}_{k,1}^{2}, \cdots, \mathcal{M}_{k,1}^{p}; \mathcal{M}_{k,2}^{1}, \mathcal{M}_{k,2}^{2}, \cdots, \mathcal{M}_{k,2}^{p}; \cdots)$.

Content Similarity: The content similarity between a node v_i and a message M_k is defined as follows:

$$\begin{aligned} S_{ik} &= Sim(v_i, M_k) \\ &= cosine(I_i, g(\mathcal{M}_k)) \end{aligned} \tag{6.1}$$

where $cosine(\cdot, \cdot)$ is the cosine similarity metric function and $g(\cdot)$ is a linear transformation function that extracts row representatives from $\mathcal{M}_j$. We will discuss $g(\cdot)$ in the following paragraphs and present its computation in Section 6.4.1.

The value S_{ik} represents the extent of how much v_i is interested in message M_k. The higher the score, the more likely that v_i is interested in M_k. We define a *similarity threshold* α, such that if the similarity score is no less than α, v_i is considered to be interested in M_k; otherwise, v_i is not interested. However, we do not select the relay node simply based on this similarity score, since the nodes with high similarity score do not always consist of the most efficient relay nodes in the network. We will present the detailed relay node selection scheme in Section 6.4.

Since each message M_k has multiple contents, we need a function $g(\cdot)$ that captures the essence of the message so that we can then compare the cosine similarity between I_i and $g(\mathcal{M}_k)$. Therefore, we use the Singular Value Decomposition (SVD) [59] method to find the row representative of M_k. In SVD, a data feature matrix A can be decomposed into the form of $A = U \times S \times W^T$, where there are r orthogonal column vectors in U that can be used to create a r-dimensional basis to describe the n data objects. These r dimensions are usually referred to as the latent variables or the latent semantics of the data. Intuitively, the columns of U can be regarded as the eigen-objects of the data, each corresponding to one independent concept/cluster. Therefore, in order to get a single scalar value that best describes the l items in the data file, we use the first column of U, which is also called the first left singular vector, and project the original data file content onto it, i.e., $Proj_{\mathcal{M}_j} = (\mathcal{M}_j^T \times U_1)^T$. An example of the message content profile is illustrated in Table 6.2. We will further describe this process in detail in Section 6.4.1.

6.3.2 Problem Formulation

We aim to disseminate the messages M_k to the nodes who are interested in the message over the time-sensitive MSN. We want M_k to reach as many nodes that are interested in M_k as possible. Based on the previous description of the MSN model,

Table 6.2: Content Representative for messages

Message ID	f_1	f_2	f_3	**...**
M_1	0.132	0.695	0.05	...
M_2	0.882	0.095	0.11	...
M_3	0.035	0.195	0.65	...
...	...	...	...	...

we present the following optimization problem.

$$\max \sum_{k=1}^{d} \sum_{v_i \in R_k^{\leq t_{\lambda_k} + t_{\iota_k}}} X_{ik}$$
$$\text{where } X_{ik} = \begin{cases} 1, & \text{if } S_{ik} > \alpha, \\ 0, & \text{otherwise.} \end{cases} \tag{6.2}$$

The objective function in this optimization represents the total dissemination outcome for all the messages. The sum of X_{ik} indicates the number of interested nodes that M_k reaches, where X_{ik} is the indicator of whether v_i is interested M_k, the condition of which is discussed in Section 6.3.1.

6.4 Interest- and Content-Based Dissemination

In this section, we present my proposed approach for maximizing the dissemination of the messages to interested nodes. The approach is divided into three major phases: *data file analysis, network structure analysis, and relay selection.*

In the data file analysis phase, we analyze each context and use the SVD [59] technique to transform the message content profile into a vector, in order to compute the content similarities to the user interest profiles. In the network structure analysis phase, we analyze the dynamic structure of the MSN and use the PageRank [70]

algorithm to compute a score that is interpreted as the centrality of each node, which measures how important a node is in the MSN. In the relay selection phase, we combine the content similarity with the centrality score to select nodes for relaying each M_k. We now describe each phase in detail.

6.4.1 Data File Analysis

Recall that, the user interest profiles are in vector format, whereas the message content profiles are in a 2-dimensional matrix format (if a message contains more than one content item). This makes it difficult to directly measure the cosine similarity between user interest and message content profiles. Using the *Feature Set* $\mathcal{F}$, we transform the message contents into the message content profile as described in Section 6.3.1. We use SVD [59] to find the first singular vector and project the message content profile onto it to get the best representation of the whole message content using latent semantics. SVD can be computed incrementally [16] when the data is time evolving.

To be more specific, given a message content profile $\mathcal{M}_k$, we first obtain its SVD, $\mathcal{M}_k = U_k \times S_k \times W_k^T$. We then take the first singular vector from U_k and denote it as U_k^1. Projecting the original content profile $\mathcal{M}_k$ onto the first singular vector gives us the row mixture, denoted by $\mathcal{M}_k^\rho$, i.e. $\mathcal{M}_k^\rho = ({\mathcal{M}_k}^T \times U_k^1)^T$. Let $g(\mathcal{M}_k)$, as it appears in Equation 6.1, be equal to $\mathcal{M}_k^\rho$. This process is illustrated in Figure 6.1.

Now that we can compute the content similarity metric between each pair of user v_i and message M_k by performing the cosine similarity between I_i and $\mathcal{M}_k^\rho$, i.e., $S_{ik} = cosine(I_i, \mathcal{M}_k^\rho)$ in Equation (6.1).

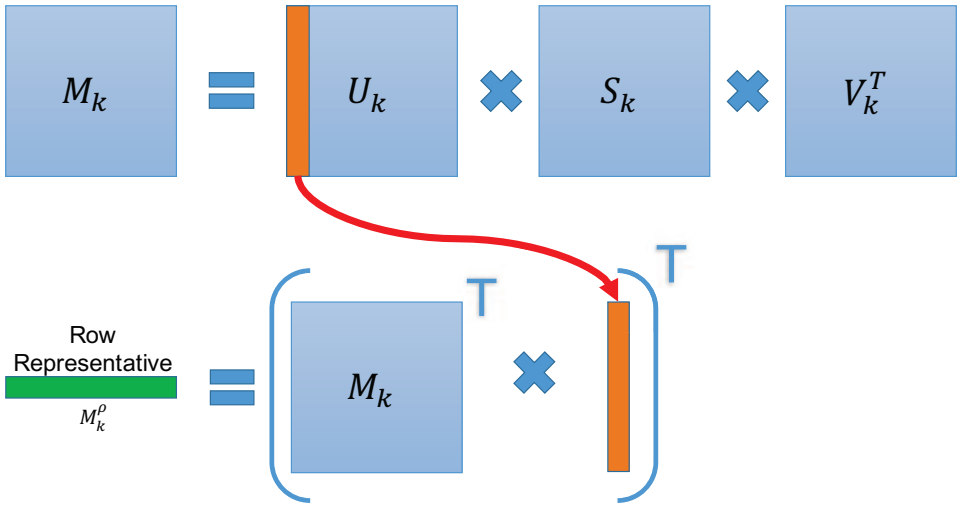

Figure 6.1: Using SVD to find row representatives for message content profile

6.4.2 Network Structure Analysis

We adopt the PageRank algorithm [70] to compute the pagerank score of each node on the MSN. PageRank is a method used by Google to rank websites in their search engine results, the underlying assumption is that more important websites are likely to receive more links from other websites [70]. We use this score as a measurement on how important the node is to help disseminate the message. The higher the score, the more important a node is, and it is more likely to be chosen as a relay node.

Let $G = (V, E)$ be a general undirected simple graph, where $V = \{v_1, v_2, ..., v_n\}$ is the set of n nodes, and E is the edge set over V with m edges. The transition matrix P is defined as:

$$P_{ij} = \begin{cases} \frac{1}{h_j}, & (j,i) \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{6.3}$$

where h_j is the degree of node v_j. The PageRank score vector $\vec{x}$ for all nodes in the

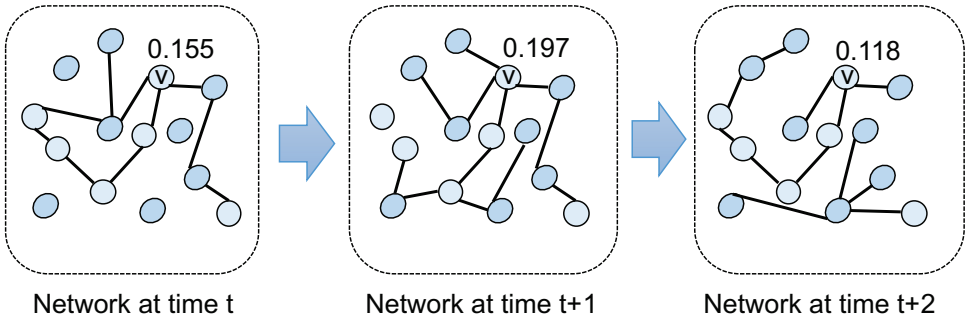

Figure 6.2: PageRank score for node v on dynamic networks

graph G is formulated as:

$$\vec{x} = \eta P\vec{x} + (1-\eta)\frac{\vec{1}}{n}, \tag{6.4}$$

where η is the *damping factor* which is usually set to be 0.85 [70].

Since the MSN structure evolves over time, we need to evaluate the effectiveness of applying the PageRank algorithm over dynamic graphs. Implementations of the incremental PageRank algorithms on dynamic graphs were proposed in [39, 7]. In practice, each node cannot access the global network structure at all times and computing PageRank in a distributed manner seems intractable, however, PageRank scores can still be approximated using local subgraphs [8]. To demonstrate the effectiveness of the PageRank algorithm used in our scheme, we use Figure 6.2 as an illustration, where at each snapshot of the dynamic network, we estimate the PageRank scores for each user node.

From Figure 6.2, we observe that the PageRank score for each node evolves with the network structure, reflecting the changes of node importance over time. The PageRank scores may need to be updated every time there is a change in the MSN structure.

6.4.3 Relay Selection

In order to appropriately select relay nodes to maximize the objective function proposed in Section 6.3.2, we need to answer the following two questions at the time when we consider relaying message M_k to node v_i: 1) Is v_i interested in M_k? 2) Can v_i be an important node that can help forward M_k to many other nodes who are interested in M_k?

We use the following scheme to integrate the content similarity and PageRank score. We define the utility score between node v_i and message M_k as follows:

$$Util(v_i, M_k) = \gamma Sim(v_i, M_k) + \mu PR(v_i, t) \tag{6.5}$$

where $PR(v_i, t)$ is the PageRank score for node v_i on graph G^t, and γ and μ are positive weighting factors for balancing content similarity and PageRank scores.

We define a *relay selection threshold* β, such that if the utility score is no less than β, v_i is selected as the relay node; otherwise, v_i is not selected at time t to receive message M_k (note that this does not necessarily mean that v_i will not be selected in the future for M_k nor that v_i cannot be selected to propagate other messages at time t). Moreover, if the similarity score between v_i and M_k is no less than α, v_i is considered to be interested in the message. On the other hand, if the similarity score is less than α, then node v_i may be considered as a data carrier and not an interested individual. The complete dissemination scheme is shown in Algorithm 2.

6.5 Evaluation Results and Analysis

In this section, we evaluate the proposed interest- and content-based data dissemination scheme under various MSN configurations. In particular, we use the dataset of a real data trace [60] to mimic the dissemination in the MSN. We implemented

Algorithm 2 Interest and Content Based Relay Scheme

Input:

Network time slices $T = [t_1, t_2, ...t_c]$;

Similarity score threshold for identifying interested nodes α;

Utility score threshold for selecting individuals as relay nodes β;

Unique feature set $\mathcal{F}$;

Output:

Number of interested individuals for all messages;

for $t \leftarrow t_1$ **to** t_c **do**
  Construct time sensitive network graph $G^t = (V^t, E^t)$, and calculate each node v_i's PageRank score as $PR(v_i, t)$;
  Construct each user's interest profile I_i;
  Define the set of neighbors of v_i in G^t as $N^t(v_i)$;
  for $k \leftarrow 1$ **to** d **do**
    for Each $v_{i'} \in R_k^{\leq t}$ **do**
      for Each $v_i \in N^t(v_{i'})$ **do**
        Calculate utility score $Util(v_i, M_k)$ using Equation (6.5);
        if $Util(v_i, M_k) \geq \beta$ **then**
          Select v_i as a relay node;
          $R_k^{\leq t} \leftarrow R_k^{\leq t} \cup \{v_i\}$;
          if $S_{ik} \geq \alpha$ **then**
            v_i is interested in message M_k and $X_{ik} \leftarrow 1$;
          end if
        end if
      end for
    end for
  end for
end for

our dissemination scheme and compared its performance with some of the existing algorithms under different metrics.

6.5.1 Evaluation Setup

We first introduce the dataset that we use to represent the MSN. The data traces allow us to follow the dissemination process in a social network and help us evaluate the performance of the proposed dissemination algorithm in a real world scenario. To test the dissemination results of the proposed algorithm on the data trace, we used the ONE simulator [38] and implemented our interest- and content-based dissemination scheme. We also used the Epidemic built-in router in the simulator as a baseline for comparison. We set the buffer size of each device to $100MB$ and the TTL of each message to 300 minutes. Messages are created every 120 seconds. Each time we uniformly randomly select a user and make him/her the source of the message. When a message is propagated to a node whose buffer does not have enough space to hold it, the oldest stored message will be deleted from the node and replaced by the new one.

6.5.2 Real Trace: INFOCOM06 Trace

INFOCOM06 data trace [60] spans almost 4 days. In the trace, there are 79 students and researchers as participants attending the student workshops. Each student and researcher carried an iMote device which has short range wireless communication capability. Furthermore, each participant filled out a survey, the last question of which specified 35 topics in computer science area and the participants marked their interests from these topics. We use these 35 topics as the feature set and use the marks of each participant as the user interest profile.

Parameter Settings for INFOCOM06 Trace

The weighting factors in Equation 6.5 are set to $\gamma = 1$ and $\mu = 10$. Notice that, since PageRank scores are amortized over all nodes in the network, in order to make PageRank more effective, a larger μ value is preferred. Then, the relay selection threshold β is set to 0.5. The similarity threshold α is set to 0.6 to determine if a node can be an interested individual.

6.5.3 Real Trace: MIT Reality Trace

MIT Reality Trace was collected in 2004 by [25], this dataset is generated by 75 students and faculty in the MIT Media Laboratory, and 25 incoming students at the MIT Sloan business school near the Media Laboratory, using Nokia 6600 smart phones with some tracking software pre-installed. The data contains a lot of information, such as movement traces of the students and their friendship and a survey which captures their interests. The total length of the data is around $500,000$ hours. In order to simplify the testing, we randomly get a subset of 5000 hours and map it into one day scale by its timestamps.

Parameter Settings for MIT Reality Trace

The weighting factors in Equation 6.5 are set to $\gamma = 1$ and $\mu = 20$, since MIT Reality Trace has more nodes than INFOCOM06 Trace and thus the amortized pagerank score is smaller, which an even larger weight is desired. Then, the relay selection threshold β is set to 0.6. The similarity threshold α is set to 0.6 to determine if a node can be an interested individual.

6.5.4 Evaluation Metrics

We compared our proposed dissemination scheme with Epidemic [68], where a message is disseminated whenever two nodes meet each other. We also compared our algorithm with the multi-cast version of SANE [49], where only the user interest profiles are taken into consideration. Furthermore, we also compared our results with the opportunistic-based strategy BinarySW [65]. Both SANE and BinarySW have been explained in more detail in Section 2.5. We analyze the performance of the dissemination protocols using the following two metrics:

- *Number of Interested Recipients*: the number of recipients who are interested in the messages being disseminated. This can be used to measure the delivery performance of the dissemination schemes.
- *Number of Messages Relayed*: the total number of times that messages are forwarded in the network over time. This essentially reflects the cost of the dissemination, where more data transfers among mobile devices indicates higher power and network resource consumption.

6.5.5 Results for INFOCOM06 Trace

We denote our proposed interest- and content- based dissemination as ICB.

Delivery Performance

We evaluate the delivery performance on various buffer size and message TTL settings. The results are an average of 10 runs, since all the results have relatively small standard deviation (in the 10 – 40 range, for number of messages delivered in the range of 1000 – 1700), we omit the confidence interval on the charts.

As we see in Figure 6.3, Epidemic achieves the highest dissemination outcome, because it utilizes every possible opportunity to flood the messages. My proposed ICB dissemination scheme is the second best and consistently improves its delivery performance over SANE and BinarySW. Note that when the buffer size is 10MB, ICB dissemination performs the best among all. This is because Epidemic creates many replicated copies of a message, quickly exhausting the overall network storage capacity when buffer sizes are small.

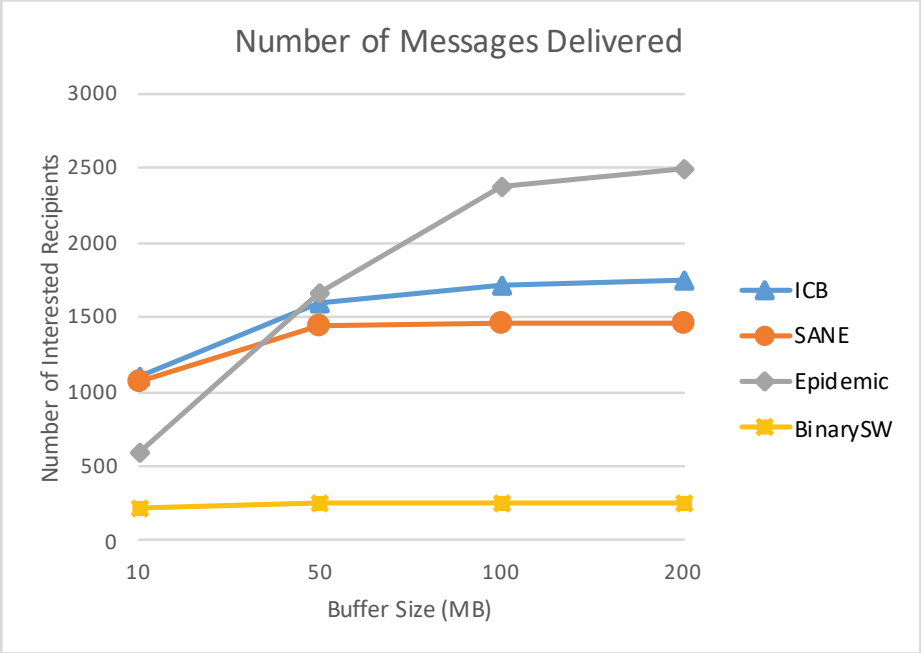

Figure 6.3: Delivery performance comparison under different buffer size settings, and default TTL = 300 mins

Figure 6.4 shows the delivery performance of the four algorithms under different message TTL settings. The observation is similar to Figure 6.3; ICB delivers messages to up to 25% more interested recipients than SANE. However, Epidemic delivery performance drops from TTL=300 to 500. This happens because the longer a message stays in the network, the more opportunities it has to be relayed, again exhausting

the overall network storage capacity.

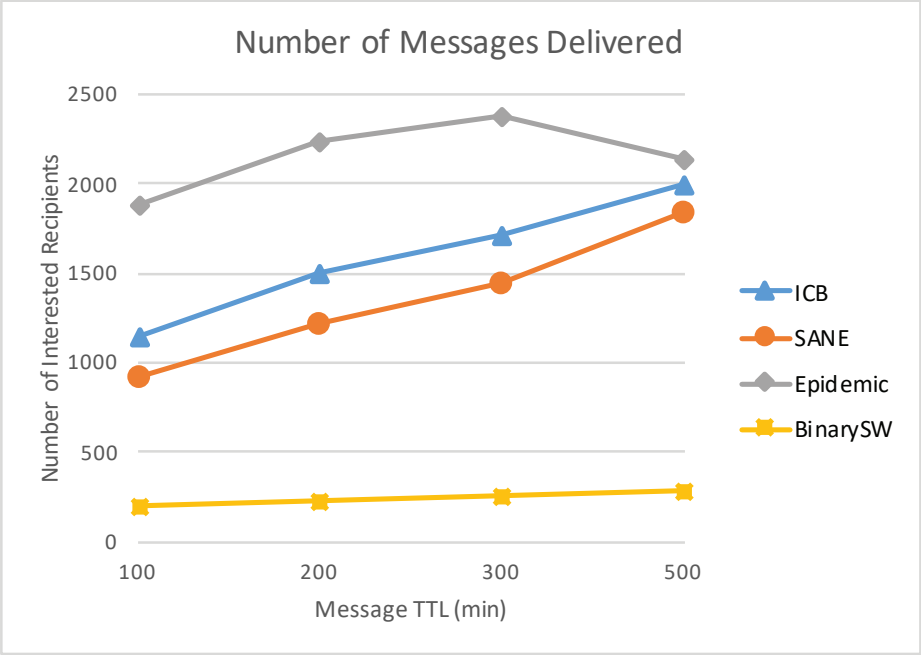

Figure 6.4: Delivery performance comparison under different message TTL settings, and default buffer size = 100MB

Delivery Cost

We also evaluate the delivery costs on the same buffer size and message TTL settings. As we see in Figure 6.5, Epidemic has the highest cost because it replicates the messages whenever there is a connection. My proposed ICB dissemination has slightly higher cost than SANE and BinarySW due to the fact that we utilize network structural important nodes as message carriers to improve delivery performance as confirmed in Figure 6.3.

Figure 6.6 shows the delivery costs of the four algorithms under different message TTL settings. When message TTL increases, the costs for all algorithms become higher since the longer a message stays in the network, the more opportunities it

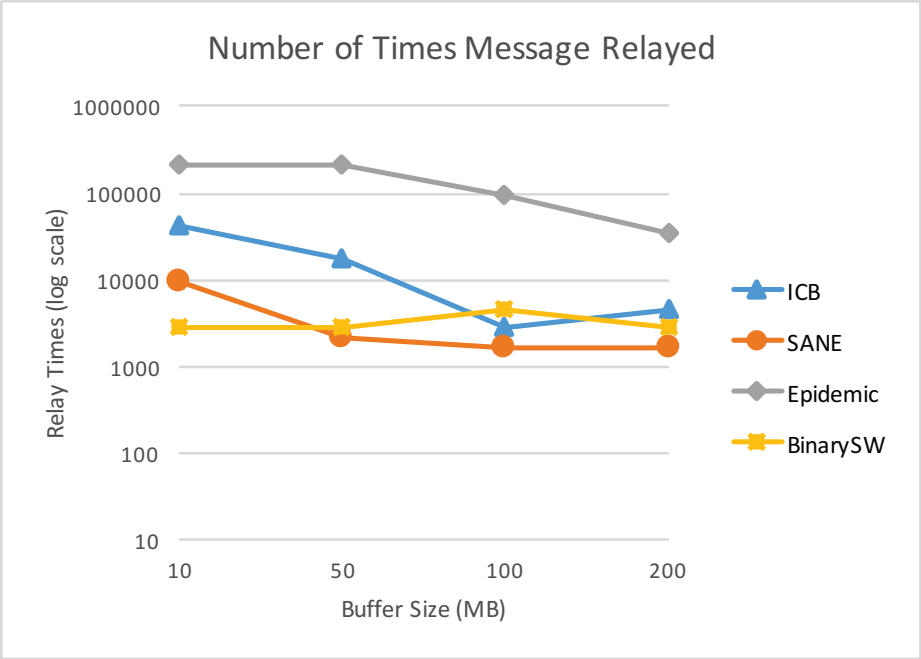

Figure 6.5: Delivery costs comparison under different buffer size settings, and default TTL = 300 mins

has to be relayed, and thus increase the delivery costs. However, our proposed ICB dissemination scheme keeps a smoother and slower increase than other algorithms.

6.5.6 Results for MIT Reality Trace

Delivery Performance

We also evaluate the delivery performance on various buffer size and message TTL settings for the MIT Reality Trace. Since again, the standard deviation of the the results are relatively small, and thus we omit the confidence interval on the charts

As we see in Figure 6.7, Epidemic achieves the highest dissemination outcome while the buffer size is large, because it utilizes every possible opportunity to flood the messages. My proposed ICB dissemination scheme is the second best and consistently improves its delivery performance over SANE and BinarySW except only for the case

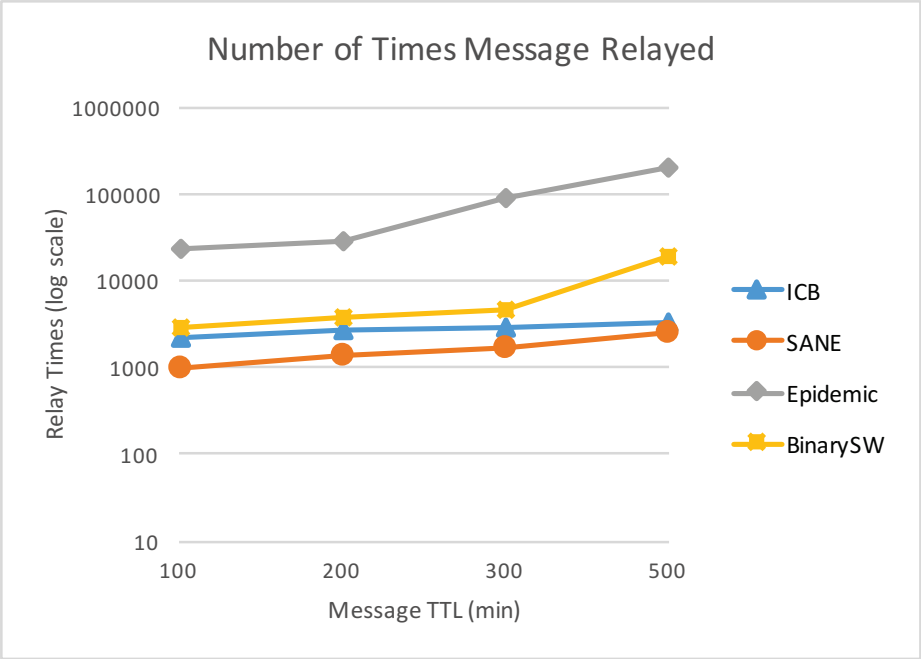

Figure 6.6: Delivery costs comparison under different message TTL settings, and default buffer size = 100MB

when buffer size is extremely small.,This is due to the fact when buffer is small, it is harder to utilize network structural important node to relay the message, since it has little room to keep these messages. And when it is small, other dissemination scheme also works poorly, such as Epidemic. This is because Epidemic creates many replicated copies of a message, quickly exhausting the overall network storage capacity when buffer sizes are small. On the other hand, SANE only relays messages to those who have interest in them so nodes keep fewer messages in their buffer and thus the final outcome is better.

Figure 6.8 shows the delivery performance of the four algorithms under different message TTL settings. Similarly to Figure 6.7, the proposed ICB dissemination scheme is able to constantly outperform SANE in various settings, Epidemic dissemination has the best delivery performance except for the case when TTL is 500

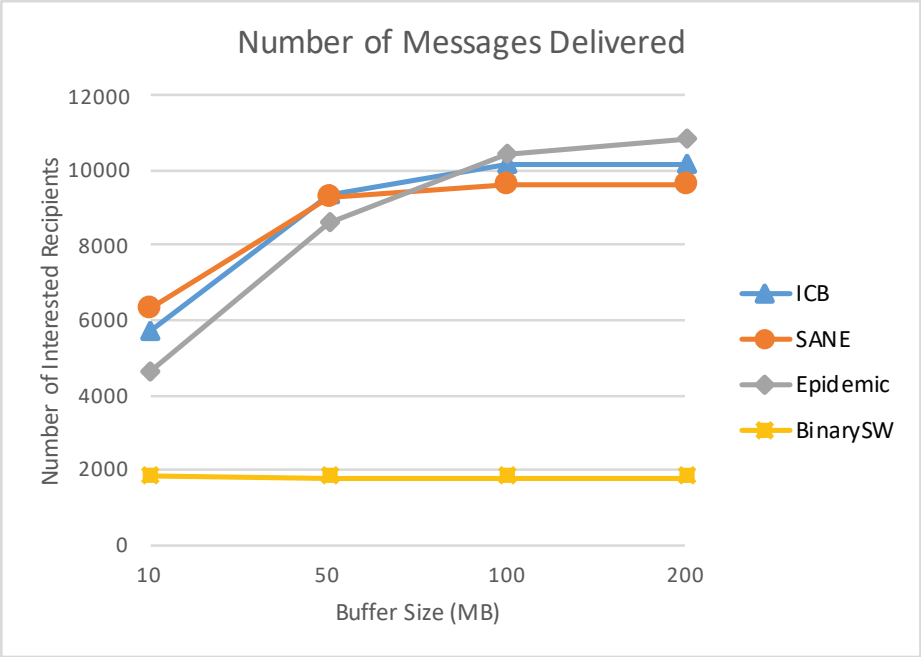

Figure 6.7: Delivery performance comparison under different buffer size settings, and default TTL = 300 mins

minutes. This is because when the TTL is large, more copies are made and thus the old data is forced to be dropped to make room for the new data, exhausting the overall network storage capacity and this results in a loss of deliveries.

Delivery Cost

We also evaluate the delivery costs on the same buffer size and message TTL settings.

As we see in Figure 6.9, Epidemic has the highest cost because it replicates the messages whenever there is a connection. My proposed ICB dissemination has slightly higher cost than SANE and when the buffer size is large, the number of relays are almost the same. BinarySW has the fewest number of relays due to the reason that it rarely relays messages and the number of deliveries is poor as we can confirm in Figure 6.7.

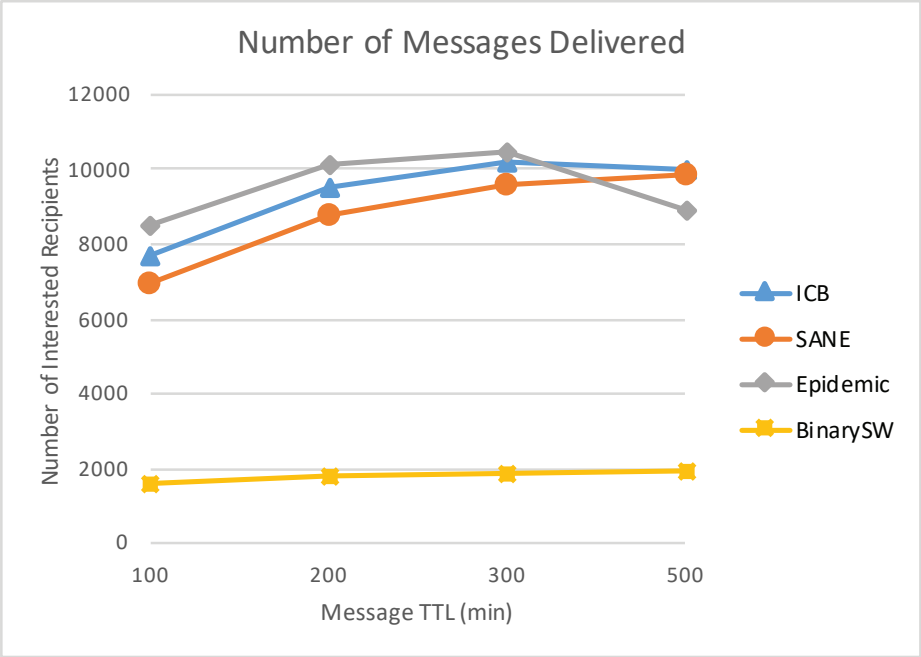

Figure 6.8: Delivery performance comparison under different message TTL settings, and default buffer size = 100MB

Figure 6.10 shows the delivery costs of the four algorithms under different message TTL settings. As we can see that when TTL is increased, the data gets more chances of being replicated and relayed, so that all four algorithms observe an increase, and for smaller TTL, the proposed ICB has almost the same relays as SANE, and when TTL becomes larger, the data gets more chances to be stored at the structural important nodes and thus the relays also increase.

6.6 Conclusion

Realizing the challenges in the data dissemination tasks in mobile social networks, we propose an efficient interest- and content-based dissemination scheme, where both message content and user interest are taken into consideration for relay selection. We also incorporate the network structure analysis technique PageRank in order to fur-

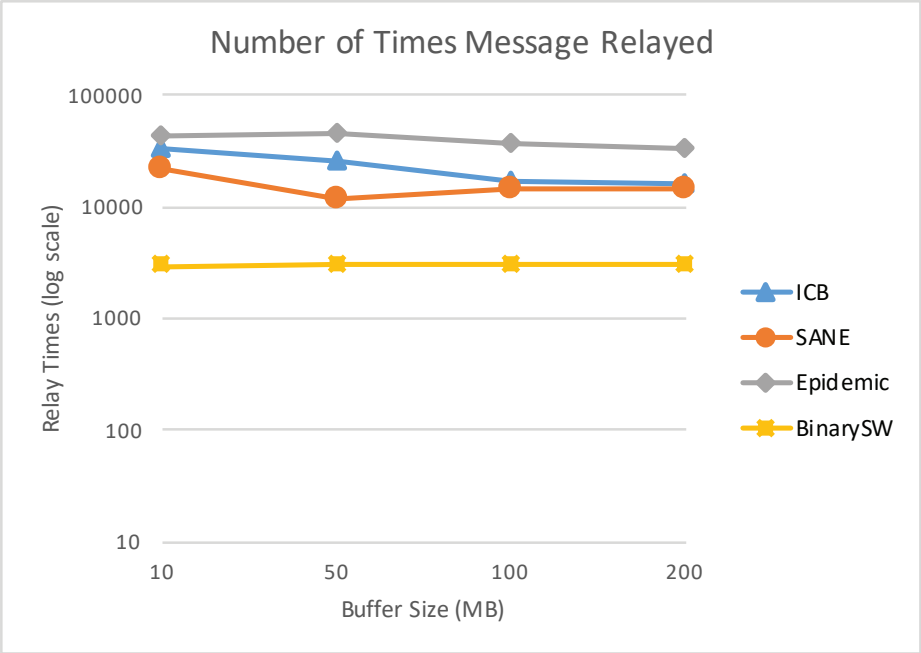

Figure 6.9: Delivery costs comparison under different buffer size settings, and default TTL = 300 mins

ther improve our relay decisions. The proposed dissemination scheme achieves higher message delivery performance compared to existing algorithms such as SANE and BinarySW algorithms, while keeping the overall costs comparable to the costs of these two algorithms and much lower than the overall cost of the Epidemic dissemination scheme.

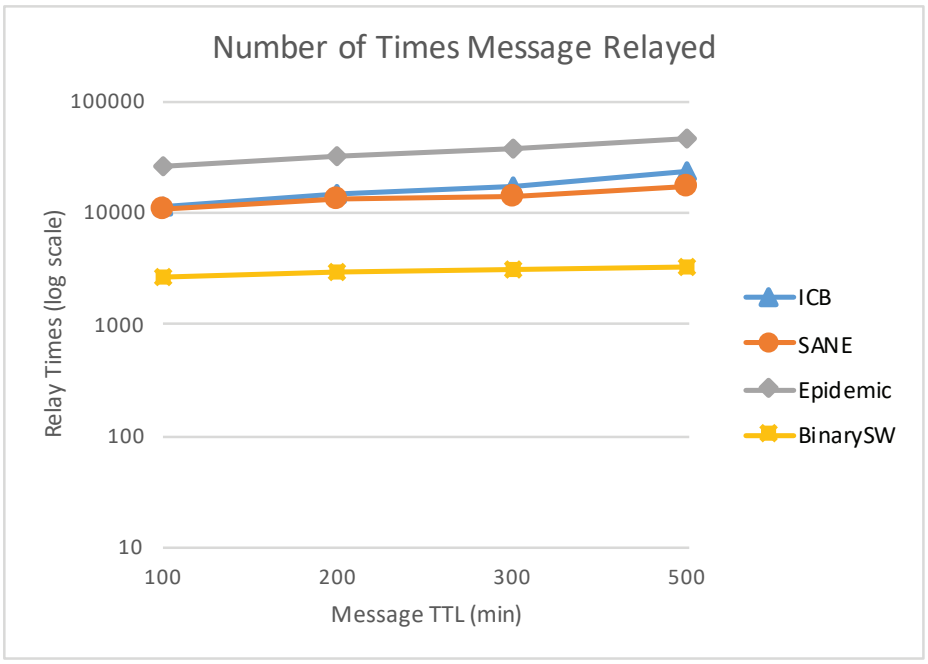

Figure 6.10: Delivery costs comparison under different message TTL settings, and default buffer size = 100MB

Chapter 7

CONCLUSIONS

In this dissertation, we first realize the challenges in delay tolerant networks and develop CoDPON architecture and utilize both the NS-2 simulator and ONE simulator to study the feasibility of using ferry boats as data mules in the Amazon Delta Region to deliver medical files from remote communities to the capital city Belém. Then we design an opportunistic routing scheme and coupled with fountain codes to help deliver the medical files, by partitioning medical files into small packets and encoding them with redundancy, and in the case of unpredictable delays and transmission loss, the extensive experiment results show that this opportunistic routing is able to recover more medical files by reconstructing from small packets that are delivered.

Furthermore, we extend the scenario of providing healthcare in Amazon Delta Region and explore the problem of Maximum Throughput Routing in ANF problem. We developed a polynomial $(\alpha, \beta) - approximation$ algorithm, which achieves a *constant* approximation ratio α for network throughput. We also validate the proposed algorithm through extensive simulations on two case studies, namely, for the boat data mule network in the Amazon Delta Region and the synthetic German50 benchmark.

We conclude by considering the data dissemination problem in Mobile Social Networks using people's interest profiles and network structures, and develop an interest and content based dissemination scheme to help broadcast messages. Experiment results on different data sets show that the proposed algorithm presented is efficient and highly-effective compared to existing routing methods.

7.1 Future Work

We would like to extend our work on data mule networks beyond the Amazon Delta region scenario. For example, we are currently investigating if it could be useful in the Tempe Public Transportation Orbits network. Orbits have five lines serving the city of Tempe in Arizona with a fixed displacement plan. All five lines have a common stop at the transportation center, and intersect with each other along their routes. Therefore, if a message is generated and needs to be transmitted to the transportation center, our routing algorithm can be applied by taking each bus as a mule to transmit data. Moreover, we are considering also testing and enhancing our protocol, when applied to random networks, such as an ER graph or a power law graph, to study the impact of topologies on file delivery probability.

Though we have incorporated a fountain code approach into robust opportunistic routing, there remains several open challenges, which lie in the future work of our research. For example, ultrasound files are equally split into k packets and treated equally. However, different snapshots of the ultrasound file may have vary in importance. An ultrasound file is a video recording of a series of images, where some frames are essential while errors in other frames are tolerable. Unequal Error Protection can be used to improve the fountain codes by generating different encoding scheme according to the importance of the different frames. However, due to the encoding mechanism of fountain codes, the importance of different snapshots and packets cannot be directly quantified. Thus, we need to design a feedback mechanism based on the missing packets that are needed for decoding the original file.

In the ANF problem, we come up with a randomized rounding algorithm which achieves constant throughput approximation, but how to turn this random algorithm into a predictable deterministic algorithm and eliminate or reduce the randomness

still remain an open challenge.

We could use the method of conditional expectations to derandomize the random rounding algorithm as described in Algorithm 1 in Section 5.4. Suppose $f_1, f_2, ..., f_i$ are already determined to be 0 or 1. Now we calculate the conditional expectation of the total flow in the network if we were to allow randomization from now on. We have:

$$\begin{aligned} & E[flow|f_1, ..., f_i \ fixed] = \\ & (1 - \tilde{f_{i+1}}) \cdot E[flow|f_1, ..., f_i \ fixed, and \ f_{i+1} = 0] \\ + \quad & \tilde{f_{i+1}} \cdot E[flow|f_1, ..., f_i \ fixed, and \ f_{i+1} = 1] \end{aligned}$$

From the equation above we can observe that for at least one of the two choices of either $f_{i+1} = 0$ or $f_{i+1} = 1$, the expected flow does not decrease. Therefore at least one of the following statement is true:

$$E[flow|f_1, ..., f_i \ fixed, and \ f_{i+1} = 0] \geq E[flow|f_1, ..., f_i \ fixed]$$
$$E[flow|f_1, ..., f_i \ fixed, and \ f_{i+1} = 1] \geq E[flow|f_1, ..., f_i \ fixed]$$

So at each step we can always check the expected value of the flow in both cases and deterministically assign f_{i+1} to be 0 or 1 leading to a greater expected flow. We already know that without fixing any f_i values, the expected flow is actually the solution of the LP, i.e., $E[flow] = OPT_{ALG}$ where each choice of f_i is randomized from the beginning. Therefore, if we use this derandomize strategy described above from the beginning, we can have:

$$
\begin{aligned}
OPT_{ALG} &= E[flow] \\
&\leq E[flow|f_1 \ fixed \ using \ Derandomization] \\
&\leq E[flow|f_1, f_2 \ fixed \ using \ Derandomization] \\
&... \\
&\leq E[flow|f_1, ..., f_n \ fixed \ using \ Derandomization]
\end{aligned}
$$

In the Interest- and Content-Based Data Dissemination problem, we have evaluated the algorithm on both INFOCOM06 Trace and MIT Reality Trace. In order to make it more robust and general, it is important to apply it on various other datasets, since different data may exhibit different network structures and the weighting factors in relay selection part described in Section 6.4.3 are user inputs, so they may be sensitive to different networks. Furthermore, there are some parameters that can potentially impact the algorithm, such as the size of message that is being disseminated. In practice, messages may have different sizes, a video promotion message is much larger than a text of news, which requires longer relay time and more battery resources. Therefore, these parameters are also important in the evaluation of the algorithm.

Besides using the PageRank scores and user interest profiles, there are other ways to improve the relay selection process. For example, we can learn the movement patterns of the users so that we can predict the message relay opportunities, this will help to resolve the shortcomings of using PageRank since PageRank only considers local networks and cannot predict the future events either.

REFERENCES

[1] "https://en.wikipedia.org/wiki/amazon_river".

[2] Abdelkader, T., K. Naik, A. Nayak, N. Goel and V. Srivastava, "SGBR: A routing protocol for delay tolerant networks using social grouping", IEEE Trans. Parallel Distrib. Syst. **24**, 12, 2472–2481, URL https://doi.org/10.1109/TPDS.2012.235 (2013).

[3] Acer, U. G., P. Giaccone, D. Hay, G. Neglia and S. Tarapiah, "Timely data delivery in a realistic bus network", IEEE Trans. Vehicular Technology **61**, 3, 1251–1265, URL https://doi.org/10.1109/TVT.2011.2179072 (2012).

[4] Ahmad, S., R. Hamzaoui and M. Al-Akaidi, "Unequal error protection using fountain codes with applications to video communication", IEEE Trans. Multimedia **13**, 1, 92–101, URL https://doi.org/10.1109/TMM.2010.2093511 (2011).

[5] Altman, E. and F. D. Pellegrini, "Forward correction and fountain codes in delay tolerant networks", in "Proceedings of the INFOCOM 2009. 28th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies", pp. 1899–1907 (2009), URL https://doi.org/10.1109/INFCOM.2009.5062111.

[6] Anastasi, G., M. Conti and M. D. Francesco, "Data collection in sensor networks with data mules: An integrated simulation analysis", in "Proceedings of the 13th IEEE Symposium on Computers and Communications", pp. 1096–1102 (2008), URL https://doi.org/10.1109/ISCC.2008.4625629.

[7] Bahmani, B., A. Chowdhury and A. Goel, "Fast incremental and personalized pagerank", vol. 4, pp. 173–184 (2010), URL http://www.vldb.org/pvldb/vol4/p173-bahmani.pdf.

[8] Bar-Yossef, Z. and L. Mashiach, "Local approximation of pagerank and reverse pagerank", in "Proceedings of the 17th ACM Conference on Information and Knowledge Management", pp. 279–288 (2008), URL http://doi.acm.org/10.1145/1458082.1458122.

[9] Burgess, J., B. Gallagher, D. D. Jensen and B. N. Levine, "Maxprop: Routing for vehicle-based disruption-tolerant networks", in "Proceedings of the INFOCOM. 25th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies", (2006), URL https://doi.org/10.1109/INFOCOM.2006.228.

[10] Cerf, V. G., S. C. Burleigh, A. Hooke, L. Torgerson, R. C. Durst, K. L. Scott, K. Fall and H. S. Weiss, "Delay-tolerant networking architecture", RFC **4838**, 1–35, URL https://doi.org/10.17487/RFC4838 (2007).

[11] Chekuri, C. and A. Ene, "The all-or-nothing flow problem in directed graphs with symmetric demand pairs", Math. Program. **154**, 1-2, 249–272, URL https://doi.org/10.1007/s10107-014-0856-z (2015).

[12] Chekuri, C., S. Kannan, A. Raja and P. Viswanath, "Multicommodity flows and cuts in polymatroidal networks", SIAM J. Comput. **44**, 4, 912–943, URL https://doi.org/10.1137/130906830 (2015).

[13] Chekuri, C., S. Khanna and F. B. Shepherd, "Multicommodity flow, well-linked terminals, and routing problems", in "Proceedings of the 37th Annual ACM Symposium on Theory of Computing", pp. 183–192 (2005), URL http://doi.acm.org/10.1145/1060590.1060618.

[14] Chekuri, C., S. Khanna and F. B. Shepherd, "Edge-disjoint paths in planar graphs with constant congestion", SIAM J. Comput. **39**, 1, 281–301, URL https://doi.org/10.1137/060674442 (2009).

[15] Chekuri, C., S. Khanna and F. B. Shepherd, "The all-or-nothing multicommodity flow problem", SIAM J. Comput. **42**, 4, 1467–1493, URL https://doi.org/10.1137/100796820 (2013).

[16] Chen, X. and K. S. Candan, "LWI-SVD: low-rank, windowed, incremental singular value decompositions on time-evolving data sets", in "Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 987–996 (2014), URL http://doi.acm.org/10.1145/2623330.2623671.

[17] Choudhury, M. D., H. Sundaram, A. John, D. D. Seligmann and A. Kelliher, "Birds of a feather: Does user homophily impact information diffusion in social media?", CoRR **abs/1006.1702**, URL http://arxiv.org/abs/1006.1702 (2010).

[18] Chuzhoy, J., "Routing in undirected graphs with constant congestion", SIAM J. Comput. **45**, 4, 1490–1532, URL https://doi.org/10.1137/130910464 (2016).

[19] Coutinho, M. M., "http://www.margalho.pro.br/codpon/", (2011).

[20] Coutinho, M. M., A. Efrat, T. Johnson, A. Richa and M. Liu, "Healthcare supported by data mule networks in remote communities of the amazon region", International Scholarly Research Notices (2014).

[21] Coutinho, M. M., T. Moreira, E. Silva, A. Efrat and T. Johnson, "A new proposal of data mule network focused on amazon riverine population", in "Proceedings of the 3rd Extreme Conference on Communication: The Amazon Expedition", p. 10 (ACM, 2011).

[22] Daly, E. M. and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets", in "Proceedings of the 8th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing", pp. 32–40 (2007), URL http://doi.acm.org/10.1145/1288107.1288113.

[23] Dong, Z., G. Song, K. Xie and J. Wang, "An experimental study of large-scale mobile social network", in "Proceedings of the 18th International Conference on World Wide Web", pp. 1175–1176 (2009), URL http://doi.acm.org/10.1145/1526709.1526915.

[24] Dubhashi, D. P. and A. Panconesi, *Concentration of Measure for the Analysis of Randomized Algorithms* (Cambridge University Press, 2009), URL http://www.cambridge.org/gb/knowledge/isbn/item2327542/.

[25] Eagle, N. and A. Pentland, "Reality mining: sensing complex social systems", Personal and Ubiquitous Computing **10**, 4, 255–268, URL https://doi.org/10.1007/s00779-005-0046-3 (2006).

[26] Edmonds, J. and R. M. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems", J. ACM **19**, 2, 248–264, URL http://doi.acm.org/10.1145/321694.321699 (1972).

[27] Erlebach, T. and K. Jansen, "The maximum edge-disjoint paths problem in bidirected trees", SIAM J. Discrete Math. **14**, 3, 326–355, URL https://doi.org/10.1137/S0895480199361259 (2001).

[28] Fabbri, F. and R. Verdone, "A sociability-based routing scheme for delay-tolerant networks", EURASIP J. Wireless Comm. and Networking **2011**, URL https://doi.org/10.1155/2011/251408 (2011).

[29] Fall, K. R., "A delay-tolerant network architecture for challenged internets", in "Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication", pp. 27–34 (2003), URL http://doi.acm.org/10.1145/863955.863960.

[30] Gao, W., Q. Li, B. Zhao and G. Cao, "Multicasting in delay tolerant networks: a social network perspective", in "Proceedings of the 10th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing", pp. 299–308 (2009), URL http://doi.acm.org/10.1145/1530748.1530790.

[31] Hui, P. and J. Crowcroft, "How small labels create big improvements", in "Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications - Workshops", pp. 65–70 (2007), URL https://doi.org/10.1109/PERCOMW.2007.55.

[32] Hui, P., J. Crowcroft and E. Yoneki, "BUBBLE rap: Social-based forwarding in delay-tolerant networks", IEEE Trans. Mob. Comput. **10**, 11, 1576–1589, URL https://doi.org/10.1109/TMC.2010.246 (2011).

[33] IBGE, "Ibge censo 2010", URL http://www.censo2010.ibge.gov.br/ (2010).

[34] Jones, E. P. C., L. Li, J. K. Schmidtke and P. A. S. Ward, "Practical routing in delay-tolerant networks", IEEE Trans. Mob. Comput. **6**, 8, 943–959, URL https://doi.org/10.1109/TMC.2007.1016 (2007).

[35] Kalyanpur, A., V. P. Neklesa, D. T. Pham, H. P. Forman, S. T. Stein and J. A. Brink, "Implementation of an international teleradiology staffing model", Radiology **232** (2004).

[36] Karsten, C. V., D. Pisinger and S. R. B. D. Brouer, "The time constrained multi-commodity network flow problem and its application to liner shipping network design", Transportation Research Part E **76**, 122–138 (2015).

[37] Kawarabayashi, K. and Y. Kobayashi, "All-or-nothing multicommodity flow problem with bounded fractionality in planar graphs", in "Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science", pp. 187–196 (2013), URL https://doi.org/10.1109/FOCS.2013.28.

[38] Keränen, A., J. Ott and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation", in "Proceedings of the 2nd International Conference on Simulation Tools and Techniques for Communications, Networks and Systems", p. 55 (2009).

[39] Kim, K. S. and Y. S. Choi, "Incremental iteration method for fast pagerank computation", in "Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication", pp. 80:1–80:5 (2015), URL http://doi.acm.org/10.1145/2701126.2701165.

[40] Laoutaris, N., G. Smaragdakis, P. Rodriguez and R. Sundaram, "Delay tolerant bulk data transfers on the internet", in "Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems, SIGMETRICS/Performance", pp. 229–238 (2009), URL http://doi.acm.org/10.1145/1555349.1555376.

[41] Li, Q., S. Zhu and G. Cao, "Routing in socially selfish delay tolerant networks", in "Proceedings of the INFOCOM. 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies", pp. 857–865 (2010), URL https://doi.org/10.1109/INFCOM.2010.5462138.

[42] Li, X., S. Huang, K. S. Candan and M. L. Sapino, "Focusing decomposition accuracy by personalizing tensor decomposition (PTD)", in "Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management", pp. 689–698 (2014), URL http://doi.acm.org/10.1145/2661829.2662051.

[43] Liu, M., T. Johnson, R. Agarwal, A. Efrat, A. Richa and M. M. Coutinho, "Robust data mule networks with remote healthcare applications in the amazon region: A fountain code approach", in "Proceedings of the 17th International Conference on E-health Networking, Application & Services", pp. 546–551 (2015).

[44] Liu, M. and A. W. Richa, "Interest- and content-based data dissemination in mobile social networks", in "Proceedings of the IEEE Global Communications Conference", pp. 1–6 (2017), URL https://doi.org/10.1109/GLOCOM.2017.8255085.

[45] Liu, M. and A. W. Richa, "Interest- and content-based data dissemination in mobile social networks", To be submitted to IEEE/ACM Transactions on Networking (2019).

[46] Liu, M., A. W. Richa, M. Rost and S. Schmid, "A constant approximation for maximum throughput routing", in "To be submitted to INFOCOM 2019", (2019).

[47] Luby, M., "LT codes", in "Proceedings of the 43rd Symposium on Foundations of Computer Science", p. 271 (2002), URL https://doi.org/10.1109/SFCS.2002.1181950.

[48] MacKay, D., "Fountain codes", IEE Proceedings-Communications **152**, 6, 1062 – 1068 (2005).

[49] Mei, A., G. Morabito, P. Santi and J. Stefa, "Social-aware stateless forwarding in pocket switched networks", in "Proceedings of the INFOCOM 2011. The 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies", pp. 251–255 (2011), URL https://doi.org/10.1109/INFCOM.2011.5935076.

[50] Mitzenmacher, M. and E. Upfal, *Probability and computing - randomized algorithms and probabilistic analysis* (Cambridge University Press, 2005).

[51] Moghadam, A. and H. Schulzrinne, "Interest-aware content distribution protocol for mobile disruption-tolerant networks", in "Proceedings of the 10th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks", pp. 1–7 (2009), URL https://doi.org/10.1109/WOWMOM.2009.5282479.

[52] Naidu, S., S. Chintada, M. Sen and S. Raghavan, "Challenges in deploying a delay tolerant network", in "Proceedings of the Third Workshop on Challenged Networks", pp. 65–72 (2008), URL http://doi.acm.org/10.1145/1409985.1409998.

[53] Orlowski, S., M. Pióro, A. Tomaszewski and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library", in "Proceedings of the 3rd International Network Optimization Conference", (2007), http://sndlib.zib.de, extended version accepted in Networks, 2009.

[54] Pentland, A., R. Fletcher and A. Hasson, "Daknet: Rethinking connectivity in developing nations", IEEE Computer **37**, 1, 78–83, URL https://doi.org/10.1109/MC.2004.1260729 (2004).

[55] Pujol, J. M., A. L. Toledo and P. Rodriguez, "Fair routing in delay tolerant networks", in "Proceedings of the 28th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies", pp. 837–845 (2009), URL https://doi.org/10.1109/INFCOM.2009.5061993.

[56] Rahnavard, N., B. N. Vellambi and F. Fekri, "Rateless codes with unequal error protection property", IEEE Trans. Information Theory **53**, 4, 1521–1532, URL https://doi.org/10.1109/TIT.2007.892814 (2007).

[57] Rost, M. and S. Schmid, "Virtual network embedding approximations: Leveraging randomized rounding", in "Proc. IFIP Networking", (2018).

[58] S. Merugu, M. A. and E. Zegura, "Routing in space and time in network with predictable mobility", Technical Report GIT-CC-04-7 (2004).

[59] Sarwar, B., G. Karypis, J. Konstan and J. Riedl, "Application of dimensionality reduction in recommender systems a case study", in "Proceedings of the ACM WebKDD'00", pp. 1–11 (2000).

[60] Scott, J., R. Gass, J. Crowcroft, P. Hui, C. Diot and A. Chaintreau, "Crawdad dataset cambridge/haggle", http://crawdad.org/cambridge/haggle/ (2009).

[61] Seguin-Charbonneau, L. and F. B. Shepherd, "Maximum edge-disjoint paths in planar graphs with congestion 2", in "Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science", pp. 200–209 (2011), URL https://doi.org/10.1109/FOCS.2011.30.

[62] Sejdinovic, D., D. Vukobratovic, A. Doufexi, V. Senk and R. J. Piechocki, "Expanding window fountain codes for unequal error protection", IEEE Trans. Communications **57**, 9, 2510–2516, URL https://doi.org/10.1109/TCOMM.2009.09.070616 (2009).

[63] Shah, R. C., S. Roy, S. Jain and W. Brunette, "Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks", Ad Hoc Networks **1**, 2-3, 215–233, URL https://doi.org/10.1016/S1570-8705(03)00003-9 (2003).

[64] Shokrollahi, A., "Raptor codes", IEEE Trans. Information Theory **52**, 6, 2551–2567, URL https://doi.org/10.1109/TIT.2006.874390 (2006).

[65] Spyropoulos, T., K. Psounis and C. S. Raghavendra, "Efficient routing in intermittently connected mobile networks: the single-copy case", IEEE/ACM Trans. Netw. **16**, 1, 63–76, URL http://doi.acm.org/10.1145/1373452.1373458 (2008).

[66] Su, Z., Q. Xu, H. Zhu and Y. Wang, "A novel design for content delivery over software defined mobile social networks", IEEE Network **29**, 4, 62–67, URL https://doi.org/10.1109/MNET.2015.7166192 (2015).

[67] Sundararaj, L. and P. Vellaiyan, "An overview of alagappa university delay tolerant water monitoring network", IJCSNS **10**, 5, 19 (2010).

[68] Vahdat, A. and D. Becker, "Epidemic routing for partially-connected ad hoc networks", Tech. rep., Duke University Technical Report (2000).

[69] Vellambi, B. N., N. Rahnavard and F. Fekri, "FTS: A distributed energy-efficient broadcasting scheme using fountain codes for multihop wireless networks", IEEE Trans. Communications **58**, 12, 3561–3572, URL https://doi.org/10.1109/TCOMM.2010.101210.070495 (2010).

[70] White, S. and P. Smyth, "Algorithms for estimating relative importance in networks", in "Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", pp. 266–275 (2003), URL http://doi.acm.org/10.1145/956750.956782.

[71] Xiao, M., J. Wu and L. Huang, "Community-aware opportunistic routing in mobile social networks", IEEE Trans. Computers **63**, 7, 1682–1695, URL https://doi.org/10.1109/TC.2013.55 (2014).

[72] Yue, J., Z. Lin and B. Vucetic, "Distributed fountain codes with adaptive unequal error protection in wireless relay networks", IEEE Trans. Wireless Communications **13**, 8, 4220–4231, URL https://doi.org/10.1109/TWC.2014.2314632 (2014).

[73] Zafarani, R., M. A. Abbasi and H. Liu, *Social Media Mining: An Introduction* (Cambridge University Press, 2014), URL https://doi.org/10.1017/CBO9781139088510.

[74] Zhang, X., J. Kurose, B. N. Levine, D. F. Towsley and H. Zhang, "Study of a bus-based disruption-tolerant network: mobility modeling and impact on routing", in "Proceedings of the 13th Annual International Conference on Mobile Computing and Networking", pp. 195–206 (2007), URL http://doi.acm.org/10.1145/1287853.1287876.