

BOSTON UNIVERSITY
GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

FORWARDING IN MOBILE OPPORTUNISTIC NETWORKS

by

VIJAY ERRAMILI

B.E., University of Delhi, 2001
M.S., University of Texas at Dallas, 2003

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2009

© Copyright by
VIJAY ERRAMILI
2009

Approved by

First Reader

Mark E. Crovella, Ph.D.
Professor of Computer Science,
Boston University

Second Reader

Christophe Diot, Ph.D.
General Manager and Director
Thomson Paris

Third Reader

John W. Byers, Ph.D.
Assoc. Professor of Computer Science
Boston University

DEDICATION

To Amma and Nannagaru, who sacrificed their dreams, so that I could live mine.

Acknowledgments

There is an African proverb that goes - “It takes a village to raise a child”. In my context, this can be suitably modified to : “It takes a city to assist Vijay get his Ph.D. degree”. This is a humble attempt to gratefully acknowledge all the denizens of this proverbial city.

First and foremost, my advisor Mark Crovella. If there is a person who can successfully advise the 43rd President of the United States towards a Ph.D. in Computer Science, it is him. Mark took me in when I wasn’t sure what I wanted to do with myself, and over the years he has taught me everything about research; starting from thinking about problems, asking the right questions, formalizing the questions, using sound techniques to answer these questions and presenting results in the best way possible. He has taught me the virtues of patience, the need to think deep, and at the same time not drown in myriad details. My research philosophy is a direct derivative of his and I’m proud to be a member of his academic family. In addition to research, we would also discuss the travails of the Red Sox, debate the awesomeness of Radiohead and the best cafes in Paris. I couldn’t wish for a better advisor.

I am very grateful to Christophe Diot and John Byers for serving on my dissertation committee. Working with Christophe is like a good roller-coaster ride. There are ups and downs but the experience is intense, and highly enjoyable. He has taught me to never leave a stone unturned while conducting research. His zeal and energy have contributed much to this thesis although I’m still waiting for the iPhone he promised me. John has given me great advice over the years, and I’ve learned a lot about teaching and oratorical skills from him.

Azer Bestavros has always encouraged me from day one. His enthusiasm for all things

science, his philosophy of “Build-it-and-they-will-come!” and his openness to discuss problems any time have helped me tremendously. I would like to thank Ibrahim Matta for being my adviser during my first year. He got me started on research and provided the very necessary initial encouragement and impetus. While I have not collaborated with Leo Reyzin, he has shown me that teaching can indeed be raised to an art form, first while I took his Crypto class and then later when I was his teaching fellow. I thank him for acting as the chair of my thesis committee. I would also like to thank George Kollios, Eric Kolaczyk and Shang-Hua Teng for helpful conversations over the years.

I am indeed very fortunate to have collaborated with Nina Taft and Augustin Chaintré. Nina was a joy to work with, for she would immediately put me at ease, that enabled me to work more efficiently. She taught me the importance of details in research. Augustin has had a profound influence on my research, both in terms of direct contribution to this thesis, as well as shaping my thinking. His passion for high quality research and his brilliance is only superseded by his general enthusiasm about everything in life. He has given me useful advice about all aspects of my research as well as the best restaurants in Paris. It has been professionally and personally rewarding to work with both of them.

A summer internship at Thomson Research, Paris gave me the opportunity to kick start research on my thesis topic. I was fortunate to have worked alongside Mostafa Ammar, Constantine Dovrolis, Jim Kurose and Jennifer Rexford while I was there, and they were kind enough to listen to my half-baked research ideas as well as offer generous encouragement. Fellow interns (Amogh, Ben, Haakon, Jiayue, Fernando, Vivek) as well as lab members (Augustin, Augustin Soule, Christophe, Hendrik, Vincent) made my months in Paris highly memorable, and amongst the best in my graduate life.

Among my colleagues in the WING group, I would like to specially thank Nahur Fonseca, who is my academic brother and my office mate. Nahur has over the years been my sounding board, always game to listen to my crazy research ideas with infinite patience and giving me valuable feedback and encouragement about many different aspects of my work. During the earlier years, Anukool Lakhina has generously given me his valuable time as well as

sound counsel. He got me enthusiastic about research in general by sheer osmosis; just being around him made me more productive. I would also like to thank Jef Considine, Karim Mattar, Mina Guirguis, Georgios Smaragdakis and Gabe Parmer for providing great feedback on my work during different stages of my graduate life.

My Big Fat Greek Family: What is the probability that a kid from New Delhi, India, a country of over a billion people, travels all the way to the US, enrolls in a PhD program, and gets adopted and subsumed by an amazing yet wholly different culture and their people all from a country of roughly 11 million? Fairly insignificant by any measure. Yet that is what happened in my case. My Greek family has adopted me as one of their own and over the years let me inhabit their joyous little world where I found support, encouragement and happiness in equal measure. I'll try to do justice to some key figures in my Greek family, in no particular order. To start with, there is Georgios Smaragdakis, who enrolled with me at the same time, and is graduating with me at the same time. Both of us have gone thru the PhD process together, handling the rough with the smooth. We have shared many a laugh together, many arguments and fights, even lived together. Yet he still persists in being associated with me. Some wonders never cease.

While I have been fortunate to meet fantastic people over the last 5 years, it has been a rare occurrence that any of them have influenced me too much. Nikos Laoutaris is an exception. If George has been like an affable elder brother, Nikos has been that and beyond. He is a mentor, a fellow philosopher and guide in the truest sense. Nikos has always given me the right advice, both professional and personal, even if sometimes I did not want to hear it. His support has been unwavering and unconditional. Nikos also knows as much about music as me (a lot), another feat that makes him rare. Both George and Nikos have survived my bad phases with patience and good-naturedness. I'm beholden to them. The next culprit is Panos Papapetrou, who also has been around for most of my PhD life, and boy am I glad for it. While he is one of the main reasons for the delay in my PhD program, for he would always drag me to extended coffee breaks or evenings at his favorite watering hole, I can surely attribute my fairly balanced and happy graduate life to him. We would

give each other advice on our problems with the fairer sex, only to realize both of us know little about the subject. He is a remarkable person and a terrific friend.

Giorgos Zervas has shared my love for indie rock music and would be game for concerts, as well as wiping the floor with me on squash courts. Michalis Potamias, while a late entrant to the Greek family, has done his part in being my partner in crime in love for punk music, appreciation for the 'Dude', social networks, general anarchy as well as letting me wipe the floor with him on the same squash courts. He also provided valuable feedback for the thesis, taking time out of his own busy schedule. Fragkiska Tsioutsiou (Francie) is a true friend; she is my cheerleader (despite surviving my cooking) and I can't thank her enough for it. My other Greek family members include Vassilis Athitsos, Yola Katsargyri, Dinos and others of the Greek community here in Boston who have provided me with many hours of joy.

My Desi Family: My Indian family in Boston has to start with Debajyoti Bera. While it has been a pleasure to share interests like love for video games, Indian food, and angst over the Indian cricket team, there are a few aspects beyond common interests that make Deb special. His brilliance, simplicity, kindness as well as his upbeat nature in the face of adversity has always inspired me. His friendship is one I am grateful for. Kanishka was a constant companion the first couple of years. Anukool, Ankit, Swapnil, Ashwin, Prashant, Manish, Dhiman, Sumant, Madhu, Satya, and Sowmya have given me memorable company as well as useful counsel from time to time. One of my best friends Akshay Damarla has given me valuable feedback on different aspects of this thesis from an outsider's perspective. In addition he has donned the role of Jeeves to my Wooster for more than 10 years, 'nuff said.

My other families include my Italian family, which comprises of Flavio and Gabriele. Flavio has fed me many an amazing Italian meal, thereby directly contributing to my sustenance. Gabriele has shared my enthusiasm for all things Apple. My Brazilian family includes Nahur and Marisa. Nahur, apart from being my academic brother and fellow Red Sox fan, has been a good sport in playing Halo, where he would repeatedly get his rear

end kicked. The sole member of my Bulgarian family is Irena Tsvetkova. She has patiently endured trips with me to record stores, to trivia nights and to cafes around Cambridge and Boston, always putting up a cheerful front, and never complaining. And she has also taught me many a important lesson in humility on the squash court. Special thanks to the rock climbing group, the acting school group, the local DNC chapter as well as Ben, Scott, Michael, Jess, Jef, Vitaly, Kyle, Nenad, Kate and Gaku for great company over the years.

I would like to acknowledge a very special person, Niky Riga. While she has been a good friend the last 5 years, it is the last 2 years where she has gone above and beyond the defining roles of friendship. She has opened doors for me (and closed them), has supported me, made my life richer by her presence in it and most importantly, taught me that one can talk without talking. She made the difficult process of finishing this thesis easy, while going thru tough times herself, all the while displaying exemplary patience and grace. I cannot fathom this thesis and indeed the last two years without her presence. Friendship, in this case, I believe is too poor a word to describe our relationship, as is *ευχαριστώ* for all that she has done and endured. I don't think I can ever repay her.

The person who inspired me to go for a PhD was my grandfather, the late Dr. E.V. Chelam. If he was the inspiration, I decided to mold myself after my uncle Dr. Ashok Erramilli. After my parents, my uncle Ashok has had tremendous influence on my life. He instilled in me the love for networks during my undergrad days when he got me started on analyzing traffic traces and finding self-similar behavior. He has provided me constant encouragement, supported me financially during my Masters degree and is the primary reason for me gravitating towards a career in research. My uncle Prof. Shyam Erramilli has always been an awe-inspiring figure and a role model. He has given me encouragement, financial help as well as sound counseling on any and all issues. My vocabulary is too poor to describe the influence both my uncles have had on me. I am eternally indebted to them. My uncle Chakri, being the lone computer scientist in the family, has done more than his part by giving useful advice, encouragement and support. My uncle Raysa and my aunt Lakshmi as well as my uncle Murali and aunt Radha have been my parents by proxy here

in the US. Their capacity for unlimited love and encouragement is staggering. I have been fortunate that my sojourn in Boston has been bookended by the delightful company of two cousin sisters, Sumana and Vyju. Sumana helped me settle down in Boston during my first year with her cheerful presence and help. My cousin sister Vyju and her husband Kiran have, over the last year, fed me amazing food, been my companions to try different restaurants in Boston, and have in general brought sunshine in my life with their cheerful demeanor. I am grateful to them. My cousins Satchal and Seetal have done their bit by raising ruckus every time they made it to Boston, leading to interesting times. My extended family in India has always been a constant source of support, encouragement and joy.

There is a quote by the director Mira Nair that goes - "It is because my roots are so strong that I can fly so high". My roots are in 14 Munirka Enclave, New Delhi and the erstwhile fellow residents of this apartment nourished these roots. My brother Vishnu, although younger, is wiser and more mature than me. I draw the courage necessary to follow the less traveled path thru him. This thesis is dedicated to my parents. They have sacrificed their own personal dreams and aspirations so that I can live mine and in the process made my dreams theirs. They give meaning and reason for this thesis and for this accomplishment.

This work was supported by NSF grants CCR-0325701 and ANI-0322990, and by funding from the HAGGLE FET project, as well as private funds by the Erramilli family.

FORWARDING IN MOBILE OPPORTUNISTIC NETWORKS

(Order No.)

VIJAY ERRAMILI

Boston University, Graduate School of Arts and Sciences, 2009

Major Advisor: Mark E. Crovella, Professor of Computer Science,

Boston University

ABSTRACT

Recent advances in processor speeds, mobile communications and battery life have enabled computers to evolve from completely wired to completely mobile. In the most extreme case, all nodes are mobile and communication takes place at available opportunities – using both traditional communication infrastructure as well as the mobility of intermediate nodes. These are *mobile opportunistic* networks.

Data communication in such networks is a difficult problem, because of the dynamic underlying topology, the scarcity of network resources and the lack of global information. Establishing end-to-end routes in such networks is usually not feasible. Instead a store-and-carry forwarding paradigm is better suited for such networks. This dissertation describes and analyzes algorithms for forwarding of messages in such networks.

In order to design effective forwarding algorithms for mobile opportunistic networks, we start by first building an understanding of the set of all paths between nodes, which represent the available opportunities for any forwarding algorithm. Relying on real measurements, we enumerate paths between nodes and uncover what we refer to as the *path explosion* effect. The term path explosion refers to the fact that the number of paths between a randomly selected pair of nodes increases exponentially with time. We draw from the theory of epidemics to model and explain the path explosion effect. This is the first contribution of the thesis, and is a key observation that underlies subsequent results.

Our second contribution is the study of forwarding algorithms. For this, we rely on trace driven simulations of different algorithms that span a range of design dimensions. We compare the performance (success rate and average delay) of these algorithms. We make the

surprising observation that most algorithms we consider have roughly similar performance. We explain this result in light of the path explosion phenomenon.

While the performance of most algorithms we studied was roughly the same, these algorithms differed in terms of cost. This prompted us to focus on designing algorithms with the explicit intent of reducing costs. For this, we cast the problem of forwarding as an optimal stopping problem. Our third main contribution is the design of strategies based on optimal stopping principles which we refer to as *Delegation* schemes. Our analysis shows that using a delegation scheme reduces cost over naive forwarding by a factor of $O(\sqrt{N})$, where N is the number of nodes in the network. We further validate this result on real traces, where the cost reduction observed is even greater.

Our results so far include a key assumption, which is unbounded buffers on nodes. Next, we relax this assumption, so that the problem shifts to one of prioritization of messages for transmission and dropping. Our fourth contribution is the study of message prioritization schemes, combined with forwarding. Our main result is that one achieves higher performance by assigning higher priorities to young messages in the network. We again interpret this result in light of the path explosion effect.

Contents

Acknowledgments	v
Contents	xiii
List of Tables	xvi
List of Figures	xvii
List of Abbreviations	xix
1 Introduction	1
1.1 Our Setting: Mobile Opportunistic Networks	2
1.1.1 Applications	3
1.2 Challenges: Mobility as a Double-Edged Sword	4
1.2.1 Challenges	4
1.3 Contributions	6
1.3.1 Study of Forwarding Paths: Path Explosion Effect	6
1.3.2 Study of Forwarding Algorithms: Similarity of Performance	6
1.3.3 Minimum Cost Forwarding: Delegation Forwarding	7
1.3.4 Forwarding under Resource Constraints	7
1.4 Thesis Roadmap	7
2 Related Work	9
2.1 Mobile Networks	9
2.2 Understanding Mobility	11
2.2.1 Abstract Models	11
2.2.2 Models Based on Measurements	12
2.3 Routing and Forwarding Algorithms	13

2.3.1	Routing Algorithms for MANETs	13
2.3.2	Forwarding Algorithms for DTNs	14
2.3.3	Forwarding Algorithms for Mobile Opportunistic Networks	14
2.4	Summary	17
3	Data Collection	18
3.1	Conference Datasets	18
3.1.1	Issues with Datasets	20
3.2	University Datasets	21
3.2.1	UCSD datasets	21
3.2.2	RealityMining	21
3.2.3	Issues with Datasets	22
3.3	Characterization Results	23
3.3.1	Variability of Contact Rates	26
3.4	Summary	27
4	Characterization of Paths	29
4.1	Overview	30
4.2	Forwarding	31
4.3	Path Enumeration	32
4.3.1	Observations	35
4.4	Path Explosion	38
4.4.1	A Homogenous Model for Path Explosion	39
4.5	The Effects of Inhomogeneity	44
4.6	Summary	48
5	Evaluating Forwarding Approaches	50
5.1	Trace-Driven Simulations	51
5.2	Algorithms	52
5.3	Observations	55
5.3.1	Similarity of Performance	55

5.4	Effective Forwarding	59
5.5	Summary	61
6	Reducing Forwarding Costs: Delegation Forwarding	63
6.1	Overview	63
6.2	Delegation Forwarding	65
6.3	Analysis of Cost	67
6.3.1	Assumptions	67
6.3.2	Cost	69
6.4	Cost Imbalance	72
6.5	Evaluation on Real Traces	76
6.5.1	Experiments	76
6.5.2	Results	80
6.6	Summary	82
7	Resource Constrained Scenarios	84
7.1	Overview	84
7.2	Assigning Priorities to Messages	86
7.2.1	Formulation	86
7.3	Delegation Number based Priorities	89
7.4	Evaluation	90
7.4.1	Experiments	90
7.4.2	Message Dropping Policies	93
7.4.3	Message Transmission Policies	94
7.5	Relation to Path Explosion	95
7.6	Summary	97
8	Conclusions	98
8.1	Summary of Contributions	98
	Bibliography	101
	Vita	111

List of Tables

3.1	Details on Conference Data Sets	20
3.2	Details on University Data Sets	22
3.3	Data sets used and chapters they are studied in	26

List of Figures

1.1	Example of Mobile Opportunistic Networks	2
3.1	Total Contacts for 24 hours (a) Infocom 06 (b) Conext 06 (c) UCSD (d) Reality GSM	23
3.2	Time Series of Total Contacts (a) Infocom 06 9AM-12PM (b) Infocom 06 3PM-6PM (c) Conext 06 9AM-12PM (d) Conext 06 3PM-6PM	24
3.3	Time Series of Total Contacts (a) UCSD 9AM-3PM (b) UCSD 3PM-9PM (c) RealityGSM 12PM-3PM (d) RealityBT 12PM-3 PM	25
3.4	CDFs - Total Contacts (a) Infocom 06 (b) Conext 06 (c) UCSD (d) RealityMining	27
4.1	Example space-time graph.	35
4.2	Algorithm for k shortest paths enumeration	36
4.3	(a) CDFs of optimal path duration Infocom 06 9-12 and 3-6 (b) CDFs of time to explosion Infocom 06 9-12 and 3-6	37
4.4	Optimal path duration vs. Time to Explosion, Infocom '06 9-12.	38
4.5	Histogram of number of paths for pairs with time to explosion ≥ 150 sec, Infocom '06 9-12.	39
4.6	Path Explosion Plots Info 06 9-12 (a) in-in (b) in-out (c) out-in (d) out-out.	47
5.1	Average Delay (sec) vs Success Rate (a) Infocom '06 9-12 (b) Infocom '06 3-6 (c) Conext 9-12 (d) Conext 3-6	53
5.2	Delay Distributions (a) Infocom '06 9-12 (b) Conext '06 9-12	54
5.3	Message Reception Times Infocom '06 9-12	56

5.4	Paths taken by forwarding algorithms for two messages in Infocom '06 9-12	
	(a) Msg 88-82 (b) Msg 86-26	57
5.5	Performance of Forwarding Algorithms on Different Types of Source-Destination Pairs: (a) Average Delay (b) Success Rate.	59
5.6	Mean rates of Intermediate Nodes Rates Infocom '06 9-12	60
5.7	Mean rates of Intermediate Nodes BoxPlots (Ratios) Infocom '06 9-12	60
6.1	Cost as function of number of nodes.	73
6.2	Cost Imbalance, 100 nodes. (a) Node Memory Load and (b) Node Transmission Load; as a function of quality.	75
6.3	Performance of Forwarding Algorithms. Top Row: Infocom 06 9-12; Infocom 06 3-6; Conext 06 9-12. Middle Row: Conext 06 3-6; UCSD 9-3; UCSD 3-9. Bottom Row: RealityGSM; RealityBT. For each dataset, plot is Cost versus Success Rate	77
6.4	Performance of Forwarding Algorithms. Top Row: Infocom 06 9-12; Infocom 06 3-6; Conext 06 9-12. Middle Row: Conext 06 3-6; UCSD 9-3; UCSD 3-9. Bottom Row: RealityGSM; RealityBT. For each dataset, plot is Cost versus Delay	78
6.5	Plots (Infocom 9-12) (a) Number of Transmissions (b) Queue Lengths	81
7.1	Delegation Tree for node A	90
7.2	Message Dropping Schemes Info 06 9-12 (a,b,c) Destination Frequency (d,e,f) Frequency; (a,d) Success Rate, (b,e) Cost and (c,f) Delay	91
7.3	Message Transmission Schemes Info 06 9-12 (a,b) Destination Frequency (c,d) Frequency; (a,c) Success Rate vs Cost, (b,d) Success Rate vs Delay	92

List of Abbreviations

AODV	Ad Hoc On-Demand Distance Vector
CDF	Cumulative Distribution Function
CDMA	Code Division Multiple Access
DLE	Drop Last Encountered
DSDV	Destination Sequenced Distance Vector
DSR	Dynamic Source Routing
DTN	Delay Tolerant Network
FIFO	First In First Out
GSM	Global System for Mobile Communications
IP	Internet Protocol
LIFO	Last In First Out
MANET	Mobile Ad-Hoc Network
MON	Mobile Opportunistic Network
P2P	Peer-to-Peer
PSN	Pocket Switched Network

TCP Transport Control Protocol

Chapter 1

Introduction

Mobile opportunistic networks (MONs) are networks where all available communication opportunities are exploited. These opportunities include using conventional communication technologies like the Internet, wireless technologies like cell-phone technology (GSM/CDMA) as well as non-conventional means like exploiting the physical mobility of nodes to transfer messages.

The celebrated result of Grossglauser and Tse [GT02] showed that incorporating physical mobility of nodes is indeed beneficial for wireless networks, leading to increase in network capacity. In addition, the last decade has borne witness to the rapid and wide-scale proliferation of mobile hand held devices. With advancements in technology, mobile devices today come equipped with multiple communication interfaces, powerful processors, enhanced battery life, and unused bandwidth. The two observations suggest that studying and exploiting the mobility of nodes to augment the existing communication infrastructure can be useful.

A key problem for mobile opportunistic networks is that of forwarding of messages between nodes. Mobility, while adding an extra dimension to be exploited, also introduces complexities that make the problem of forwarding of messages challenging.

This thesis presents a principled study of the problem of forwarding of messages in mobile opportunistic networks. We rely on real measurements to guide our study and let the characterization results of the measurements inform the design of effective forwarding

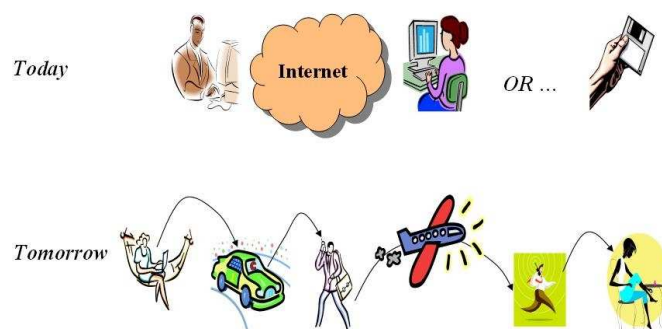


Figure 1.1: Example of Mobile Opportunistic Networks

algorithms for MONs.

This chapter is organized as follows. We first present a very high-level description of MONs and the specific type of environment this dissertation is concerned with in Section 1.1. We then pose the main question that this thesis deals with in Section 1.2 and highlight the challenges that makes answering this question hard. In Section 1.3, we state the main contributions of the thesis and we end in Section 1.4 with the outline of the rest of the dissertation.

1.1 Our Setting: Mobile Opportunistic Networks

The last decade has seen networks evolving from a largely wired and stationary environment to a wireless and mobile environment. Indeed, rapid advances in wireless technologies as well as in processor speeds and battery life of handheld devices have helped in wide-scale proliferation of portable devices [Kes05, BBC, Res].

In highly mobile environments, communication can take place by relying on traditional communication infrastructure like the Internet, or cellphone technology like GSM/CDMA or short range wireless technologies like Bluetooth or IrDA.

In addition wireless networks also include another dimension that can be exploited for communication purposes: the physical mobility of nodes. Communication can take place by relying on intermediate nodes to physically carry and transfer messages. Mobile op-

opportunistic networks are networks where all possible opportunities can be exploited for communication - use traditional networking infrastructure whenever possible or rely on physical mobility of the nodes when convenient. Fig. 1.1 shows an example of such a network, where the hope is that in the near future, one can augment existing infrastructure for data communications with physical mobility of nodes (in the scenario titled “Tomorrow”).

Our Focus: While mobile opportunistic networks can arise in many different contexts, we restrict ourselves to scenarios where the mobile nodes are essentially handheld devices carried by humans in conference settings, college campuses, and similar social settings. Such wireless networks are also referred to **Pocket Switched Networks (PSNs)** [CHC⁺07, CHC⁺05] and have close connections to mobile ad-hoc networks (MANETs) and delay tolerant networks (DTNs) [JFP04]. The social settings in which MONs operate in lead to dense connectivity as well as high mobility.

1.1.1 Applications

Some possible applications of MONs include:

- **Delay Tolerant Applications:** These include email, bulletin board services and file transfers. All of these applications can tolerate various degrees of delay.
- **Content Distribution:** With the recent increase of P2P traffic, ISPs may seek to use alternate distribution means like mobile opportunistic networks for content distribution.
- **Infrastructure-less Environments:** In areas where infrastructure does not exist, or cannot, mobile opportunistic networks allow communication between nodes that are not directly connected.
- **Circumvent censorship:** Mobile opportunistic networks can help circumvent censored networks since they rely on no third-party equipment.

All of the applications mentioned above assume the availability of an underlying **forwarding** protocol that can successfully deliver messages from one node to another. The

primary focus of this thesis is the study and design of forwarding for MONs. We formalize this issue in the next section.

1.2 Challenges: Mobility as a Double-Edged Sword

Routing messages is a key function of any communication network. While routing for wired networks has been extensively studied, routing for mobile networks is still an open research area. As pointed out earlier, mobility brings new opportunities that be exploited. However at the same time, mobility also brings complexities that need to be addressed.

The underlying topology for mobile networks is highly dynamic. Therefore, establishing and maintaining end-to-end paths between nodes is generally not practical. Hence, routing solutions for wired networks do not translate well [Fal03]. Instead, a store-carry-and-forward paradigm is more suited for mobile networks. In networks that rely on this paradigm, messages are stored at nodes and are carried to their intended destination or are forwarded to nodes that can carry the messages to their intended destination. This in contrast to standard routing solutions for wired networks where a path is setup between two nodes, and the message follows that path to its intended destination. We now highlight the key challenges facing the design of forwarding algorithms for MONs.

1.2.1 Challenges

- Heterogeneity of contact rates: Message transfer in MONs takes place when two nodes are in close physical proximity of each other. We refer to such events as *contacts*. The rate of contacts between mobile nodes is extremely important in determining how forwarding will take place, and even in determining the feasibility of forwarding in MONs. It has been shown that contact rates between nodes in MONs is heterogeneous, varying over two orders of magnitude [CHC⁺07]. This heterogeneity adds to the design complexity of forwarding algorithms.
- Unpredictable mobility: In some MONs, the underlying mobility is predictable and

periodic [PFH04, BHT⁺03, Fal03]. In the settings we are interested in, the underlying mobility is very aperiodic and unpredictable, especially on smaller time scales. This affects the design of forwarding algorithms.

- Resource Constraints: Since the nodes in MONs are portable handhelds, they bring with them the following constraints:
 - Energy Constraints: Mobile devices come equipped with short battery lives. Therefore it is imperative to design protocols that require low energy.
 - Short Contact Durations: The high dynamic nature of the underlying network gives rise to short contact durations between nodes. Short contacts limit the number of messages that can be transferred during a contact opportunity.
 - Limited Buffers : While memory has become extremely cheap over the years, the amount of memory that a node might allocate for forwarding to other nodes might still be limited. Limited buffer sizes therefore impose an additional constraint.
- Lack of global knowledge: The highly dynamic nature of the underlying nodes leads to the general lack of accurate and timely global information of the network topology. Mobile nodes have local connectivity information *at best*.

The most naive solution for forwarding in MONs would be flood replicas of a message to all nodes in the network, whenever possible. This scheme is also known as the **Epidemic** scheme [VB00]. While epidemic forwarding scheme gives the best success rate with the lowest delay, it suffers from high resource consumption. This solution cannot scale to a large number of nodes. Therefore there is a need for forwarding algorithms that give performance close to that of epidemic, but that consume much less resources. Thus, the main challenge of this thesis is:

We seek to design forwarding algorithms for mobile opportunistic networks that provide a high delivery rate, achieve low average delay and make efficient use of resources.

We next summarize the main contributions of this thesis that addresses this challenge.

1.3 Contributions

Our research is geared towards the design and study of effective forwarding algorithms for mobile opportunistic networks. To that end, we pursue a principled study of the problem by relying on real measurements of contacts between mobile nodes to guide and inform the design of forwarding algorithms. The rest of this section deals with our main contributions in further detail.

1.3.1 Study of Forwarding Paths: Path Explosion Effect

Departing from previous work on the design of forwarding algorithms for mobile opportunistic networks, our work begins with the following observation: in order to study forwarding, start by studying the opportunities available for forwarding, that is the paths between nodes.

We rely on real contact traces and enumerate paths between nodes. Studying the properties of paths leads to our first contribution: for most pairs of nodes, the number of paths connecting them increases exponentially with time. We refer to this phenomenon as *path explosion*. We study the path explosion phenomenon by building an analytical model of the network we are studying, as well as by conducting a characterization study of the traces.

1.3.2 Study of Forwarding Algorithms: Similarity of Performance

Having grasped an understanding of the paths that exist between nodes in MONs, we next focus on studying various forwarding algorithms, paying particular attention to how forwarding algorithms perform under the following two metrics: success rate (the percentage of messages successfully delivered to their intended destination) and average delay.

We find the surprising result that most of the forwarding algorithms we consider perform similarly. This is our second contribution and we explain it in light of the path-explosion

phenomenon.

1.3.3 Minimum Cost Forwarding: Delegation Forwarding

While the algorithms we considered performed similarly when we consider success rate and average delay, they differed in terms of cost incurred in the network. That is, some algorithms incurred less cost while achieving the same performance as other algorithms. This observation prompts us to focus on costs and to design forwarding algorithms that seek to reduce costs. Drawing from the principles of optimal stopping theory, we develop and analyze a new method, *delegation forwarding*, that we show reduces cost by a factor of $O(\sqrt{N})$ over naive schemes. We further demonstrate the reduction in costs in real traces. This is our third contribution.

1.3.4 Forwarding under Resource Constraints

Finally, we note that the preceding work was done under the assumptions of infinite bandwidth and buffer sizes in mobile nodes. However as mentioned in Section 1.2, mobile networks face resource constraints. Hence we complete this thesis by studying forwarding under resource constraints. The main contribution of this study is an improved understanding of message prioritization schemes. In particular we show that assigning higher priority to young messages in the network can lead to higher success rate when mobile nodes have small buffers.

1.4 Thesis Roadmap

The rest of this dissertation is organized as follows. In Chapter 2, we review related work on the main problem raised in this chapter and highlight the key differences between the work presented in this thesis and related work.

Chapter 3 is concerned with discussing the measurements we use for our study. Since our thesis is driven by empirical study, subsequent chapters build up-on characterization

observations reported in this chapter. We present results of studying forwarding paths in Chapter 4. We also introduce useful notation to analyze forwarding algorithms in this chapter that is used in later chapters as well. Armed with the knowledge gleaned by studying paths, we next study forwarding algorithms in Chapter 5. Some of the algorithms we introduce in this chapter are revisited in later chapters. We explain this by using results from Chapter 4. We investigate cost in detail in Chapter 6, and we propose a new algorithm that we show to reduce cost drastically. We investigate forwarding algorithms under resource-constrained scenarios in Chapter 7. We summarize the contributions of the thesis in Chapter 8.

Chapter 2

Related Work

The key contributions of this thesis build up-on, and are related to, previous work done on mobile networks (Mobile Ad-Hoc Networks (MANETs) and Delay Tolerant Networks (DTNs)). More specifically we build on the study of mobility and algorithms developed for information dissemination in these two networks. In addition there are methodological similarities to measurement driven research for this problem. We review the previous work done in this chapter and structure the chapter as follows.

We first present some background material on MANETs, DTNs and MONs in Section 2.1. Since mobility of nodes is the most crucial factor for the study of forwarding, we review the work done on understanding mobility in Section 2.2. We then move to discussing different algorithms that have been proposed for MONs in Section 2.3. We summarize in section 2.4.

2.1 Mobile Networks

Mobile networks can largely be bracketed into three categories: Mobile Ad-Hoc networks (MANETs), Delay-Tolerant Networks (DTNs) and Mobile Opportunistic Networks.

Mobile Ad-hoc Networks (MANETs): MANETs are created when wireless nodes form a network without the presence of any infrastructure. All nodes in a MANET are sources and destinations of messages, as well as routers [PB94, JM96]. The nodes compris-

ing network could be mobile, although the mobility of nodes is assumed to be low. The underlying topology changes over long time-scales (hours or even days). Hence the mobility of nodes is assumed to be predictable. Variants of MANETs also include Vehicular Ad-Hoc Networks (VANETs) [VAN], where a network can be formed between moving vehicles in an urban setting or over a highway, and messages can be propagated over multiple vehicles. While the underlying mobility for VANETs is higher, there is an element of predictability which can be exploited for message propagation purposes.

Delay-Tolerant/Disruption-Tolerant Networks (DTNs): Many existing protocols such as TCP/IP make certain assumptions about the underlying network topology. Typical assumptions include: (1) the existence of an end-to-end path between any two nodes, (2) the fact that maximum round trip-time is bounded, and (3) that message drop probabilities are low. However these assumptions do not hold for some networks. Some examples include networks where messages are delivered to rural areas using commuter buses that act like a store and forward message switches with limited wireless range [PFH04, WSG⁺04], or deep space communications networks [BHT⁺03], or even military ad-hoc networks [Fal03]. These networks are characterized by periodic or predictable mobility and the applications for DTNs generally operate at longer time scales (days to months), and hence are tolerant of delays by design.

Mobile Opportunistic Networks (MONs): As discussed in Chapter 1, the key difference between MONs and the networks discussed above, is that the underlying mobility is high, where the underlying topology changes in a matter of seconds and is unpredictable. In addition MONs are generally formed in smaller geographic areas like a university campus, downtown area of a city, office environments. The applications for networks, while being delay-tolerant, generally operate under time scales ranging from minutes to hours.

2.2 Understanding Mobility

The wide scale proliferation of wireless, mobile devices has given rise to questions that concern both networking researchers and operators: (a) How do we cope with mobility in the design of protocols for wireless, mobile networks and (b) how do we exploit and leverage mobility to design more efficient protocols for wireless, mobile networks. This thesis is primarily interested in the second question, although an understanding of the first question is important.

Work devoted to answering the first question includes studying the performance of protocols for mobile networks (including MANETs), network dimensioning and location based services among others.

For the second question, there has been a lot of work starting with the seminal paper by Grossglauser and Tse [GT02] which showed that the capacity of a wireless network can increase due to node mobility. Much work has been done since then in the area of opportunistic networks.

Both the questions (a) and (b) have to start with a firm understanding of the underlying mobility of nodes and we briefly review the work done in this area.

2.2.1 Abstract Models

We refer to the mobility models that have been designed purely by intuition and observations as abstract models. The emphasis for constructing abstract models is to use them in the design and evaluation of protocols, and not in understanding mobility per se.

The first models to be put forth are the simple Random walk [BCSW98, Dav00] and the Random Waypoint model [JM96]. Under the random waypoint model, nodes are assigned speeds drawn from a distribution and are assigned random destinations. Once the nodes reach their destination, they pick a new destination - independent of the previous destination, a new speed and the process repeats itself. These models are not very realistic, as human movement is more complicated but are very simple and are amenable to analysis.

Efforts to add more realism in the simple random waypoint model led to a plethora of variants, where the main focus was to incorporate more ‘human-like’ movement traits. These include correlated movement patterns [LH03, RMSM01], movement that represents social settings, for instance there are always more popular destinations [BdAF05] as well as incorporating physical obstacles like buildings and movement constraints like streets and roads [JBRAS03]. In addition, new applications (military-related applications) [HGPC99, SK99] and new scenarios (vehicular networks) [CB05, SJ04] have led to new models for mobility. These new models incorporate vehicle-like speeds for nodes (acceleration and deceleration at traffic stops), topology of streets as well as traffic patterns observed in transportation networks.

The work done in this thesis relies on real measurements and does not rely on any model of the underlying mobility of nodes. It could be possible that a simple model like Random waypoint may lead to some characteristics observed (like highly variable inter-contact time distribution between nodes) in the traces we use [CE07, KBV07] but that is not the concern of this thesis .

2.2.2 Models Based on Measurements

To develop more realistic models of mobility, as well as to understand mobility itself, there has been work that relies on empirical measurements. This includes work that tried to model users on a 802.11 network [JLB05, MV03, HKA04, KKK06, TB02, BC03, HH06], cellular network [EP05a, HW05] as well as encounter traces [CHC⁺05, EP05a, SMO06, MMC08]. The primary differences between all these works is the granularity of mobility captured, as well as the geographic area in which the users are mobile. For instance, the work done on capturing mobility in 802.11 networks primarily focus in university settings or office buildings. The mobility is captured by logging users connected to different APs, across time. In contrast, work done on understanding mobility in cellphone networks focuses on a much larger geographical area, a town or a city. The mobility or movement patterns that are logged at cellphone towers are of people moving from one cellphone cell to another, associ-

ating themselves with different towers as they move, over time. And finally the encounter traces provide fine-grained person-to-person contact data, and this data is usually collected in smaller areas like conference venues, and university campuses. Hence the different types of traces collected underscore the point that mobility should be studied under different environments and different settings. In this thesis, we use encounter traces as well as user logs in 802.11 networks.

As mentioned, this thesis does not concern itself with modeling mobility or movement of nodes. Instead we focus on exploiting mobility in forwarding, and so focus on certain properties unearthed in the relevant studies, specifically the underlying heterogeneity of contact rates [CHC⁺05, KBV07]. We discuss heterogeneity of contact rates in greater detail in Chapter 3.

In summary, much work has been done in developing, analyzing and using models for mobility. This previous literature informs our work, although we do not contribute to mobility modeling per se.

2.3 Routing and Forwarding Algorithms

From the previous two sections, we learned what really separates different types of mobile networks is the underlying mobility - the intensity of mobility as well as how predictable the underlying mobility is. This separation is similar to the one proposed by Borrel et al [BAZ07], where the authors organize different mobile networks based on the the underlying node mobility.

We therefore discuss routing solutions for MANETs, DTNs and for MONs separately, for the solutions proposed are different from each other.

2.3.1 Routing Algorithms for MANETs

Much work has been done in routing algorithms for MANETs. Starting with Destination-Sequenced Distance Vector (DSDV) [PB94] and continuing with Dynamic Source Routing

(DSR) [JM96], Ad hoc On-Demand Distance Vector (AODV) [PR99] , most of MANET routing algorithms have been designed by assuming low mobility. That is, the underlying topology changes slowly with time. This essentially means that contemporaneous paths between nodes may exist for long periods of time - thereby encouraging solutions for routing that can be directly taken from the wired domain and adapted to deal with occasional topology changes.

2.3.2 Forwarding Algorithms for DTNs

DTNs generally differ from MANETs, in that in a DTN, nodes are more mobile. Higher mobility entails that routing-based solutions for wired networks as well as for MANETs will not work, as contemporaneous paths do not exist. Therefore, store-carry-and-forward solutions are more suited for DTNs.

Most of the forwarding solutions proposed for DTNs rely on predictability of node mobility. This could be for interplanetary networks [BHT⁺03] or nodes that move with regular schedules like vehicles [LW07, SDPG06]. The key idea is that since, the mobility schedules of planets or vehicles like public transport buses [BBL05, BGJL06, BLV07] are known in advance, one can design forwarding schemes to take advantage of this advance information. An example is the bus network in rural India that is used to transport messages [WSG⁺04]. Routing for such networks can be done if the schedules of the buses is known in advance.

2.3.3 Forwarding Algorithms for Mobile Opportunistic Networks

The design and study of forwarding algorithms for mobile opportunistic networks is a different problem than for MANETs and DTNs as the underlying network topology is highly dynamic, the network is heterogeneous, and resources like bandwidth, buffer space and energy are scarce. We discuss previous work done on design of algorithms.

To date most DTN forwarding algorithms have been analyzed under the assumption that contact rates between nodes are homogeneous [DFGV03, GT02, SPR04, SPR05]. Likewise,

the most common mobility model used for forwarding evaluation is random waypoint [JM96, CBD02] in which all nodes' speeds are drawn from the same distribution, as well as the directions which the nodes move towards. In contrast, work in this thesis shows that in our setting, it is the *differences* in behavior of nodes in the network that is key to understanding performance. The closest work to the results in our thesis that deal with the implications of heterogeneity on forwarding can be found in [CLF06]. The authors of [CLF06] claim that in completely homogeneous conditions, past contact information cannot be used for making forwarding decisions. In heterogeneous conditions, past contact information becomes relevant, and the authors develop schemes to exploit the heterogeneity of contact rates.

We can frame forwarding of messages as a search problem; the search for a path between the source node and destination node. The simplest solution is to replicate and forward the message upon every opportunity. This is *epidemic* forwarding [VB00]. While epidemic forwarding will find a path if it exists, this algorithm also consumes resources like battery life and buffer sizes that are scarce to begin with. Therefore most of the work done has focused on the design of schemes that achieve comparable performance in terms of success rate and delay as epidemic but at a lower cost.

Some algorithms use variants of past contact history to predict future contacts and therefore make forwarding decisions accordingly. Jain et al [JFP04] propose a routing algorithm which seeks to reduce average delay of message delivery. The amount of future knowledge provided to the algorithm is varied and the algorithm is tested. Jones et al [JLW05] and Lindgren et al [LDS03] develop algorithms which rely on historical contact information to make forwarding decisions. The premise is that past contacts between nodes is a good indicator for future contacts. We study related algorithms in this thesis.

Some algorithms use temporal information gleaned from past contacts to make decisions [GV06, DFGV03]. The authors of [GV06] propose a scheme called EASE where forwarding decisions are made by incorporating location information along with recency of contact information with a particular destination. The same authors propose FRESH [DFGV03]

where the authors show that only recency of contacts with the destination, temporal information is sufficient for making forwarding decisions. We study FRESH along with the algorithms we develop, in this thesis. Yet some algorithms also impose pre-defined quotas on the number of replicas per message that can exist [DH07, SPR08].

The main difference between all the previous work on forwarding and the contributions of this thesis is the following. Instead of choosing a setting and trying to design and analyze forwarding algorithms within that setting, our approach is markedly different - we start by analyzing the properties of *paths* and letting that guide the design and study of forwarding algorithms.

Contributions of this thesis include studying existing algorithms on real measurements, as well as the design and study of algorithms that reduce costs dramatically. Theoretical analysis of MONs assume a homogeneous network where all nodes are equally likely to meet the destination of a message [DFGV03, CHC⁺07, SPR08]. In that case, the performance/cost trade-off is simply determined by the number of nodes that are used for the delivery of a single message. There has been recent work that considers heterogeneous conditions showing the maximum flow that can be achieved by static routing if global information about nodes' schedules is known [GGL07]. The algorithms we develop in this thesis do not assume global information, and forwarding decisions are made in an on-line manner when nodes meet.

While design of schemes that reduce cost is valuable, we also focus on the performance of our schemes in a resource constrained environment – in an environment where bandwidth and buffer sizes is limited. Under buffer size and bandwidth constraints, the focus shifts to assigning priorities to messages. There has been prior work done in designing message priority schemes. Epidemic routing [VB00] uses FIFO scheme, while the main idea behind Drop Least Encountered (DLE) (developed in [DFL01] and subsequently used with different forwarding proposals [LDS03, SDPG06, GV06]) is to drop the message with least likelihood of delivery. A similar idea has been proposed in PREP [RHB⁺07] where messages that are farthest from their destination are assigned low priorities. These proposals primarily differ

in how they define the notion of distance to the destination. In our work, we take the probability of successful delivery to the intended destination as the distance. The higher the probability of delivery, the lower is the distance to the destination. There has been work in message prioritization in related networks, like vehicular DTNs, most notably MaxProp [BGJL06] where a node assigns high priorities to messages that are relatively new in the network.

Our work is different (and indeed harder) as we have to consider short contact durations as well as ensure low overall cost in the network given that energy is a precious resource unlike in vehicular DTNs [BGJL06, BLV07].

2.4 Summary

In this chapter, we have discussed the relevant related work to the main contributions of our thesis, namely contributions to the problem of forwarding in mobile opportunistic networks. Starting with a brief discussion of type of networks we are dealing with, and how they differ from other related networks, we turned our attention to the work done on understanding and modeling mobility of nodes - that is a central aspect of mobile networks.

We then discussed previous work done in routing and forwarding in related networks as well as in the settings we are working with. Most of the previous work usually share the following template - decide on a scenario, pick a mobility model that fits that scenario, design forwarding solutions for that scenario, analyze the algorithm and compare the algorithm against other proposed algorithms. Our approach in this thesis is markedly different. We start addressing the problem of forwarding by first studying the properties of forwarding that are available. In order to do so, we start by studying paths formed in real networks, as given to us by measurements. We describe these measurements in the next chapter.

Chapter 3

Data Collection

This chapter describes the specific data sets used in our work, along with a preliminary characterization study of these data sets. Our data sets are obtained from the following two settings: (a) people moving in a conference environment and (b) people moving in a university campus environment. The two groups of data sets are markedly different in terms of how they were gathered, the granularity of information contained, and the scale of the network. It is for these reasons we present them separately.

This chapter is organized as follows. We begin in Section 3.1 with an overview of the conference data sets. Section 3.2 follows with a description of the university data sets. We base our study on specific parts of the datasets and we present characterization results and properties of datasets in Section 3.3. Finally, in Section 3.4 we provide a précis of the characterization results and how they relate to the contributions of the thesis.

3.1 Conference Datasets

The practical implications of any study of forwarding algorithms will be limited by the accuracy of the mobility of the underlying nodes. In order to capture mobility in a realistic manner, we rely on measurements of human movement that have been collected under the Huggle project [Hag]. The basic idea is that human mobility is captured by people carrying Intel Motes (iMotes) [NDGG07] equipped with Bluetooth radios, in various environments.

These devices periodically scan their surroundings for other Bluetooth devices and record them as potential communication opportunities in a trace that can be analyzed.

The Huggle datasets have been described in detail by Chaintreau et al [CHC⁺07, CHC⁺05] and Nordstrom et al [NDGG07]. However we reproduce some of the main distinguishing points about these datasets here for completeness.

The measurements were collected using iMotes that were given to human volunteers at two conferences, **Infocom 2006** (April 23-29, 2006) and **Conext 2006** (December 3-6, 2006). At the two conferences, the set of individuals included people from 18 countries, affiliated with 36 organizations, and this set of individuals were uniformly distributed across the countries and organizations. The participant set included students and professionals (in academia and industry) alike. The individuals were volunteers and no criteria were used to select volunteers except a first-come-first-serve basis.

The iMotes use the Bluetooth radio to collect samples of the surrounding Bluetooth devices by alternating between the inquiry scan and inquiry states. The inquiry scan state is periodically entered every 120 seconds. A random number of seconds is added or subtracted to avoid synchronized scans. The rest of the time the iMote is in the inquiry state that allows it to respond to scans. The collected samples are subsets of the population of surrounding Bluetooth devices at discrete points in time, whose sizes depend on the inquiry scan length, the level of noise and interference.

A successful response from an inquiry scan constitutes a *contact*, that is a potential opportunity to transfer messages. A contact lasts until it is no longer seen for two consecutive inquiry scans. Every device thus, saves the following information - the device ID of a contact, time of contact and for how long the contact existed. At the end of each experiment, the devices were taken back from the volunteers and the contact information was collected into a single trace. The encounter traces have the following format:

Device ID 1	Device ID 2	Time of Contact	Duration of Contact
--------------------	--------------------	------------------------	----------------------------

Some basic statistics about the conference datasets are described in Table 3.1. The

	Infocom 06	Conext 06
Event Duration (days)	4	4
# iMotes (mobile/long range)	78/20	72/17
Inquiry Scan Time (s)	5	5
Random Inquiry Scan Interval (s)	120 ± 12	120 ± 5
# External Devices Seen	13655	5961
# iMote (internal) Contacts	196,259	93411
iMote contact time (median/ μ / σ) (s)	227/382.1/781.7	121/156.8/98.1
# of internal contacts/pair/day	5.16	2.45

Table 3.1: Details on Conference Data Sets

main take away points from the conference datasets (i) Heterogeneity of Contact Rates [CHC⁺07], as can be noted from the standard deviation figures given in the Table 3.1 and (ii) Density of contacts - as can be observed from Table 3.1, the Infocom data sets have, on average, 5.16 contacts per node pair per day, and the Conext datasets have density to be 2.45 contacts per node pair per day, that show that the contacts are relatively dense.

3.1.1 Issues with Datasets

As investigated and reported in [NDGG07], software and hardware issues in the iMotes led to some issues with the datasets. The biggest problem was that for some iMotes, memory issues led to the iMote being reset. Once an iMote is reset, the clock offset for that iMote is set to zero, leading to synchronization issues. This software bug affected **Infocom 2005** and **Infocom 2006** datasets, however the bug was addressed before the **Conext 2006** datasets were collected. In addition, the resets also sometimes erased the contents of the memory that contained the contact records. One of the implications of the resets is that the measured density is lower than the actual density; the contacts were *under-sampled*. We note later that this issue does not have an adverse affect on the results presented in this thesis; if anything the results are conservative.

3.2 University Datasets

The conference datasets give us fine-grained data about device-to-device contacts. While such fine-grained datasets are very useful, they are also somewhat limited in their scope, for they were collected in a controlled conference environment. In order to broaden the scope of our study, we also utilize datasets collected in universities, in particular UCSD [MV03] and MIT [EP05a]. Both these datasets are publicly available from CRAWDAD [EP05b]. In addition, both the university datasets are much larger in scope and are collected from a larger environment.

3.2.1 UCSD datasets

UCSD datasets essentially consist of client-based logs of the visibility of WiFi Access Points (APs) spread around the UCSD campus. Our study, however, requires device-to-device contact information. Therefore, these datasets are pre-processed. The key assumption made is that two devices connected to the same AP at the same time, should be able to communicate with each other directly. The ‘nodes’ in this dataset are generally students carrying PDAs.

3.2.2 RealityMining

The traces from the Reality Mining [EP05a] project at MIT consist of 100 participants walking around the campus, logging visible GSM towers. The same assumption of two devices connected to the same AP at the same time should be able to communicate with each other directly was used to pre-process this dataset as the UCSD dataset. In addition, the Reality Mining datasets also log direct bluetooth contacts between devices where the scanning and logging rate is every 300 seconds. This portion of the dataset is similar to the conference dataset and is more fine-grained.

We present some statistics of both the datasets in Table 3.2.

Similar to the take away points for the conference datasets, we note: (i) Heterogeneity of

	UCSD	Reality Mining (GSM)	Reality Mining (BT)
Event Duration (days)	77	246	246
Random Inquiry Scan Interval (s)	120 \pm 10	10	300
Devices participating	273	100	100
# Internal Contacts	195,364	572,190	54,667
# of internal contacts/pair/day	0.034	0.23	0.022

Table 3.2: Details on University Data Sets

Contact Rates [CHC⁺07], as can be noted from the standard deviation figures given in the Table 3.2 and (ii) Density of contacts - as can be observed from Table 3.2, the UCSD data sets have, on average, 0.034 contacts per node pair per day, and the RealityMining datasets have density 0.23, 0.022 contacts per node pair per day, that show that the contacts are relatively sparse, compared to the university datasets.

3.2.3 Issues with Datasets

The problem with the university datasets is that certain inaccuracies exist due to the assumption made during the pre-processing stage. First of all, one may end up with more contact events than actually exist since two devices attached to the same (WiFi or GSM) base station may still be out of range of each other. However, certain contact events may be missed, such as when two devices pass each other at a place where there is no instrumented access point. Another potential issue with these data sets is that the devices are not necessarily co-located with their owners at all times. Despite these inaccuracies, the traces are a valuable source of data as they are collected in an environment which is different from a conference environment. In addition, considering two devices connected to the same AP/base station as being potentially in contact with each other is not altogether unreasonable. These devices may indeed be able to communicate locally through the AP/base station.

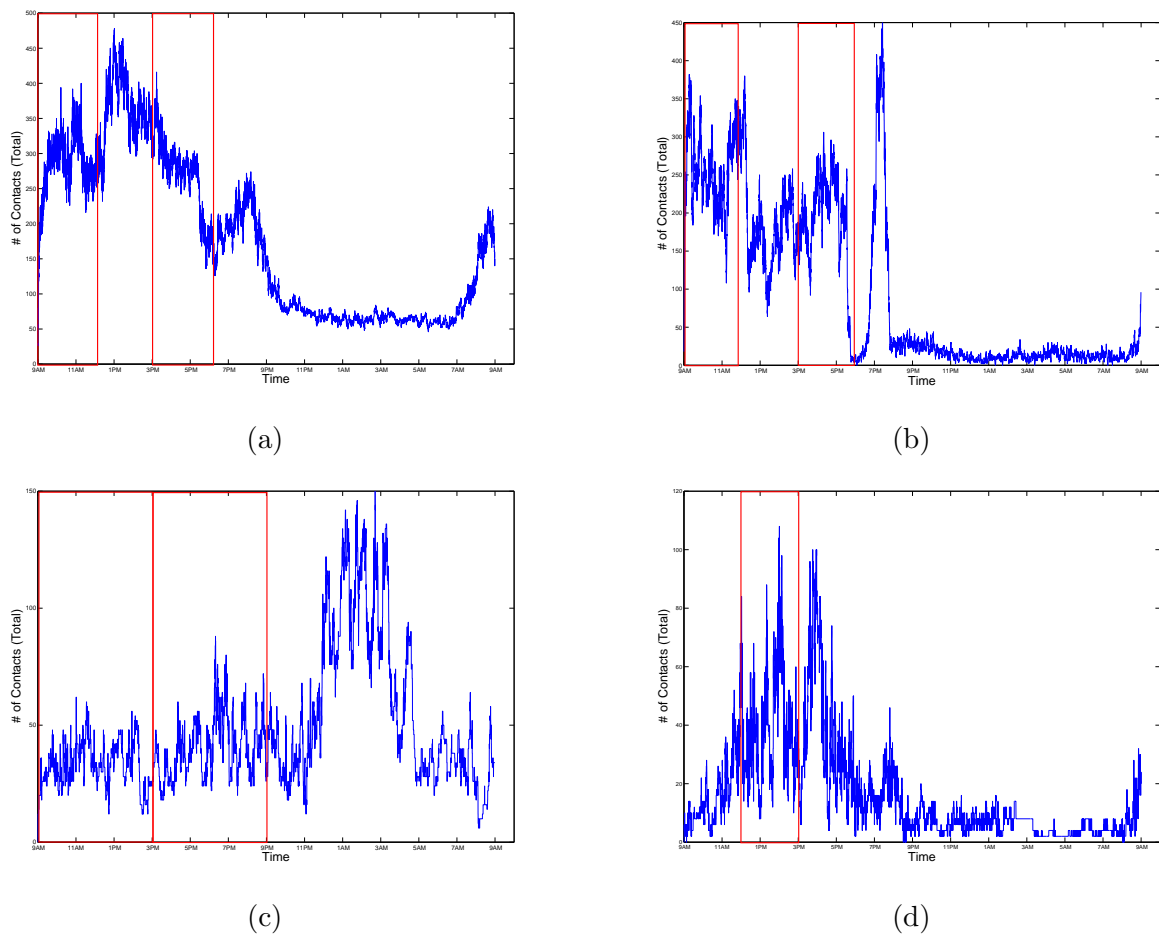


Figure 3.1: Total Contacts for 24 hours (a) Infocom 06 (b) Conext 06 (c) UCSD (d) Reality GSM

3.3 Characterization Results

Before we proceed with the main results of this thesis, we take a closer look at the datasets and present some characterization results here.

To start with, we pick representative days :

- Tuesday, 25 April 2006 for Infocom 2006
- Tuesday, 4 Dec. 2006 for Conext 2006
- Wednesday, 16 Oct. 2003 for UCSD

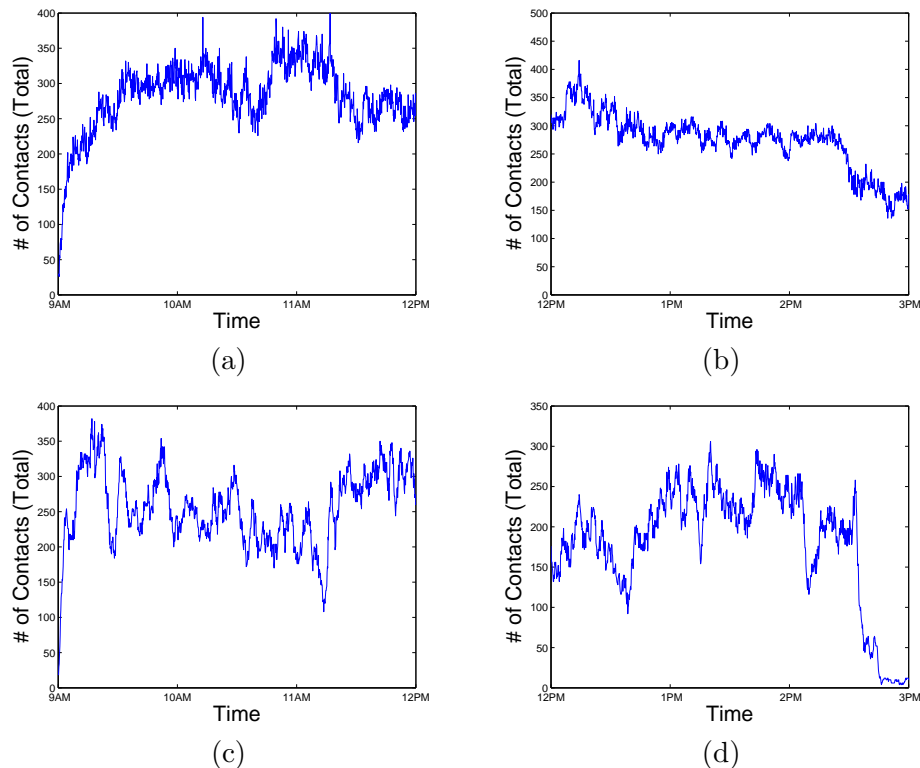


Figure 3.2: Time Series of Total Contacts (a) Infocom 06 9AM-12PM (b) Infocom 06 3PM-6PM (c) Conext 06 9AM-12PM (d) Conext 06 3PM-6PM

- Tuesday, 21 Sept. 2004 for Reality Mining BT
- Wednesday, 22 Sept. 2004 for Reality Mining GSM

from the data sets and plot the total number of contacts, occurring in a 10-second bin, across the 24 hours we pick. We present the results in Fig. 3.1. We note that the number of contacts vary across the day and as well as the prevalence of a strong diurnal trend, not unlike that observed for traffic flows in the Internet [LPC⁺04].

In order to simplify the study of forwarding, we deliberately focus on time periods extracted from our datasets. We extract time periods where the total number of contacts are approximately *stable*. By stable, we mean the total contact rates are approximately constant. We also use stable contact rates as an underlying assumption for most of our analysis. As can be seen in Fig. 3.1, we pick two 3-hour periods from the Infocom 2006

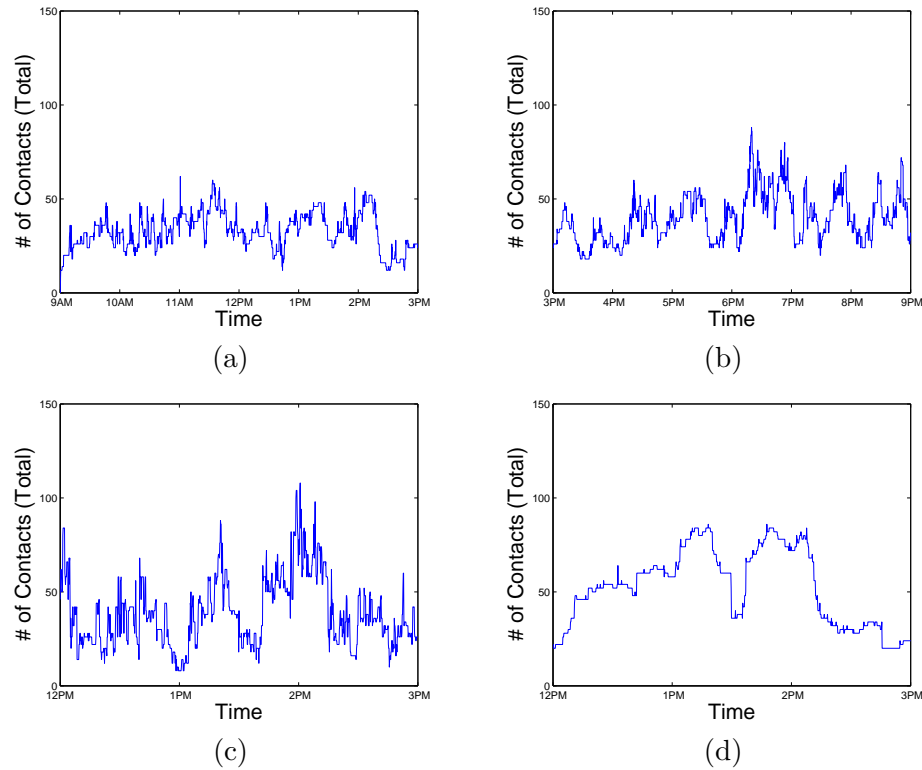


Figure 3.3: Time Series of Total Contacts (a) UCSD 9AM-3PM (b) UCSD 3PM-9PM (c) RealityGSM 12PM-3PM (d) RealityBT 12PM-3 PM

dataset and two 3-hour periods from the Conext2006 dataset, two 6-hour periods from the UCSD dataset and one 3-hour period each from the Reality Mining GSM and BT dataset respectively.

We present the magnified datasets in Fig. 3.2, 3.3. We can see that the datasets we have picked have approximately-stable total contact rates. This is important because we can be confident that any inferences we make are not affected by high or sudden variability.

The data which is logged is unidirectional; we log when one device contacts another device as one event. However, for the purposes of the work in this thesis, we make contacts bidirectional. That is, when a device contacts another device, this is recorded as two events. We modify the datasets accordingly.

Data Sets	# of Nodes	Density	Chapters
Infocom 2006 9AM-12PM	98	0.6925	4,5,6,7
Infocom 2006 3PM-6PM	98	0.419	4,5,6,7
Conext 2006 9AM-12PM	98	0.40	4,5,6,7
Conext 2006 3PM-6PM	98	0.32	4,5,6,7
UCSD 9AM-3PM	77	0.038	6
UCSD 3PM-9PM	77	0.050	6
Reality GSM	39	0.070	6
Reality BT	33	0.719	6

Table 3.3: Data sets used and chapters they are studied in

3.3.1 Variability of Contact Rates

One of the unrealistic assumptions in much related work (as is that all nodes contact each other at the same rate. In fact, prior work has shown that the per-node contact rates in the datasets we consider can be quite variable across different nodes. For example, the authors of [CHC⁺07, KBV07] show that the distributional tails of inter-contact times for similar data sets approximately follow a power law.

In Fig. 3.4 we plot the CDF for the total number of contacts each node has over each three/six hour periods for the data sets we consider. We refer to the number of contacts a node makes per unit time as the node’s *contact rate* or just *rate*. The CDFs suggest that the distribution of contact rates can be approximated as uniform over the range $(0, max)$ where *max* varies depending on the dataset. The key observation is that some nodes have rates quite close to zero, yielding extremely large average intercontact times. In fact, it is clear that the population consists of some nodes that are quite frequently in contact with a large number of other nodes, while there are also some nodes that rarely encounter other nodes at all. This observation is in line with some studies that show that movement patterns of humans are not well modeled using uniform or homogeneous assumptions [BdAF05, MM06] due to underlying social reasons - the fact that humans tend to contact some humans more than others due to social or geographic reasons.

We describe the data sets we use along with the chapters that they are studied in Table 3.3. For the university data sets, we note a decrease in the number of active nodes from

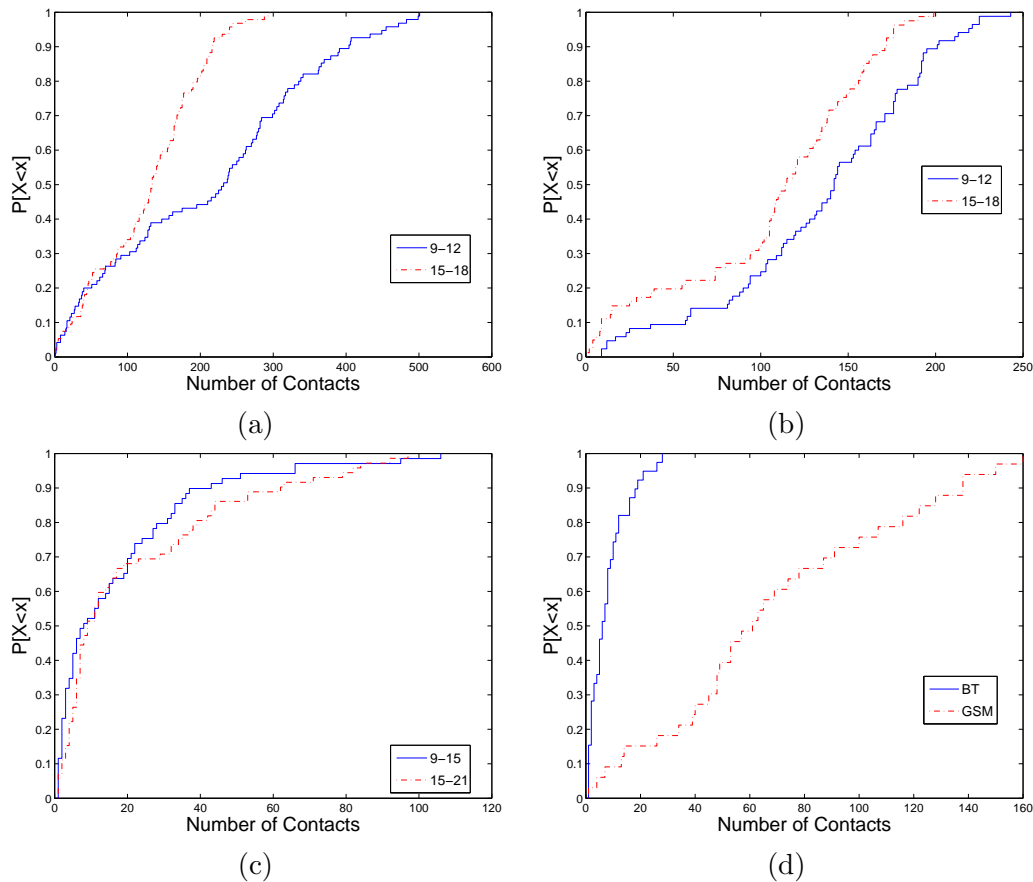


Figure 3.4: CDFs - Total Contacts (a) Infocom 06 (b) Conext 06 (c) UCSD (d) RealityMining

the total number of nodes present in the experiment - this is because we consider only the number of nodes that have at-least one contact in the time window we choose. We would also note that the density (which is defined as total number of contacts per node pair per second) varies considerably between the datasets.

3.4 Summary

We have described the data sets that we rely on for answering the main questions posed in this thesis. The data sets span different scenarios: university settings and conference settings. We have detailed how they were collected, processed and highlighted the problems with the data sets. While all the data sets show heterogeneity of contact rates, they also

differ in terms of density of contacts.

We start analyzing and building up the main results of this thesis using these data sets in the next chapter.

Chapter 4

Characterization of Paths

Forwarding of messages in a mobile opportunistic network is a challenging problem. It depends on diverse factors such as the mobility of nodes and the physical space in which the nodes are moving. Previous work on designing of forwarding algorithms for MONs have made assumptions about the underlying mobility and developed algorithms based on them. We have reviewed some of the related work in Chapter 2.

Our approach to the problem of designing forwarding algorithms is different from the previous work. We believe that in order to design effective forwarding algorithms for mobile opportunistic networks, it is necessary to start by understanding the opportunities for forwarding that exist. There is little understanding to date on the *nature* of the forwarding problem in the settings under which MONs normally operate. In particular, little is known about the kinds of paths (making use of both mobility and multiple hops) that exist in MONs.

This chapter discusses the study of paths which are formed in mobile opportunistic networks. Our credo is: understanding paths can lead to the design of better algorithms. The results in this chapter lay the foundation for subsequent chapters.

4.1 Overview

We define a path as a sequence of contacts that could potentially be used by an algorithm to forward a message. We develop an efficient method to enumerate all paths of interest between nodes and apply it to the conference traces mentioned in Chapter 3 to capture all paths of interest for a set of randomly generated messages.

The property of the collection of paths we obtain is the presence of a phenomenon we term *path explosion*. Path explosion refers to the case in which, once the first path reaches the destination, the number of subsequent paths reaching the destination grows rapidly with time, so there exist many near-optimal paths. We find that path explosion occurs for the large majority of messages in our datasets.

To explore this effect we build an analytic model describing how paths are created in a homogeneously mixing population, we then use the model to show how path explosion arises. Our homogeneous model, however, does not explain all aspects of the phenomena observed in our data. In order to understand the nature of forwarding paths more accurately, we show that it is critically necessary to take into account the different contact rates exhibited by different nodes in the population. We find that it is useful to characterize the source and destination as either high contact rate ('in') nodes or low contact rate ('out') nodes. We show that when the contact rates of the source and destination nodes are taken into account, the empirical properties of paths and the path explosion process can be understood more completely.

This chapter is organised as follows. We begin in Section 4.2 with the formalization of the forwarding problem. We continue in Section 4.3, where we discuss our path enumeration method in detail and apply it to our traces. The main observation is the path explosion phenomenon. We discuss this effect in greater detail in Section 4.4, where we construct an analytical model to understand the path explosion phenomenon. We then explore the effect of non-homogeneity in Section 4.5 to better understand and explain the path explosion phenomenon. We end with a summary of results in Section 4.6.

4.2 Forwarding

A forwarding algorithm solves a decentralized search problem—it searches for a short path between a source and destination node, starting at a given point in time. This path exists in space and time, and its duration is the amount of time between message generation and message delivery. In order to understand how hard the search problem is for mobile opportunistic networks, and specifically the settings we study MONs in, we start by studying the solution space.

Given a set of nodes \mathbb{M} with $|\mathbb{M}| = N$, and a continuous time index $0 \leq t < t_{max}$, we define a *path* as a sequence of tuples:

$$((\mathcal{N}_1, t_1), (\mathcal{N}_2, t_2), \dots, (\mathcal{N}_k, t_k))$$

where for all i , $t_i \leq t_{i+1}$, $\mathcal{N}_i \in \mathbb{M}$, and

$$\dots, (\mathcal{N}_i, t_i), (\mathcal{N}_{i+1}, t_{i+1}), \dots$$

may be present only if \mathcal{N}_i is in contact with node \mathcal{N}_{i+1} at time t_{i+1} . Each tuple in the sequence is a *hop* and the *length* of a path is the number of hops it contains.

We assume that communication occurs via *messages* which are transmitted in whole from node to node in zero elapsed time. Messages travel along paths, i.e., they are only transmitted between nodes that are in contact. For any source $\sigma \in \mathbb{M}$ and destination $\delta \in \mathbb{M}$, a successful message delivery beginning at time t_1 can occur if there is a path

$$((\sigma, t_1), (\mathcal{N}_2, t_2), \dots, (\delta, t_k)).$$

The most basic goals of a forwarding algorithm are, given σ, t_1 , and δ : (a) to find a path if it exists; and (b) to find the path with shortest-achievable duration if more than one path exists. We use $\mathcal{P}_A(\sigma, \delta, t_1)$ to denote the event that forwarding algorithm A can find a path from (σ, t_1) to (δ, t_k) for some $t_k < t_{max}$. If forwarding algorithm A finds at least one path, we denote the duration of the shortest-found path ($t_k - t_1$) as $\mathcal{T}_A(\sigma, \delta, t_1)$. We define $\mathcal{R}_A(\sigma, \delta)$ to denote the number of copies or replicas of a message under algorithm A in the

network. Note that this message need not be delivered; we count all replicas of un-delivered messages as well.

Our concern centers on the *performance* of forwarding algorithms. We define the performance of forwarding algorithm A as the *average delay*

$$D_A = \mathbb{E} [T_A(\sigma, \delta, t_1) \mid \mathcal{P}_A(\sigma, \delta, t_1)]$$

and the *success rate*

$$S_A = \mathbb{E} [\mathbb{I}_{\{\mathcal{P}_A(\sigma, \delta, t_1)\}}]$$

when σ and δ are chosen uniformly at random over \mathbb{M} , t_1 is chosen uniformly at random over $[0, t_{max})$ and $\mathbb{I}_{\{X\}} = 1$ iff $X \neq \{\}$. In addition to the above two metrics, an additional but important metric is the cost incurred by forwarding of messages. We define *average cost* of forwarding algorithm A as

$$C_A = \mathbb{E} [\mathcal{R}_A(\sigma, \delta)]$$

where σ and δ are chosen uniformly at random over M .

This chapter primarily deals with performance of forwarding algorithms, so we focus only on average delay and success rates; we focus on cost in Chapter 6.

4.3 Path Enumeration

As discussed in Chapter 2, many previous studies of forwarding algorithms have looked at forwarding performance in specific settings. However, our work takes a different approach by first empirically characterizing the set of paths that are available for use by forwarding algorithms. To accomplish this properly, there are a number of specific considerations.

First, it is important to specify the characteristics that are expected of all forwarding algorithms because these define the set of paths of interest. We assume that under any reasonable forwarding algorithm, a node holding a message for a destination node will deliver that message whenever it encounters the destination. We call this the assumption of *minimal progress*. We also restrict our attention only to loop-free paths, i.e., paths in which

no node appears more than once. We make this restriction because if looping paths are considered, then an arbitrarily large number of paths may be generated simply by following a loop a varying number of times. One can imagine a scenario where, due to high mobility, a node is encountered more than once over time and hence can potentially show up in a path multiple times. Enumerating such cases does not expose truly distinct forwarding paths. While looping paths may occur in practice this is not a concern, because disregarding such looping paths means that our counting results are conservative—so, in fact, even more paths may be present than our results indicate. For a similar reason, we assume nodes have infinite buffers and we do not consider paths that may be created because a node is forced to drop a message. Thus, once a node receives a message it holds the message forever.

A path respects minimal progress and loop avoidance if it does not contain more than one instance of any node, and the destination appears only at the end of the path (if at all). Furthermore, the assumption of minimal progress combined with unlimited message holding implies a more subtle condition, which we call *first preference*. Consider a path

$$P = ((\sigma, t_1), \dots, (\mathcal{N}_i, t_i), \dots, (\mathcal{N}_j, t_j), (\delta, t_k)).$$

If it also happens that \mathcal{N}_i encounters the destination δ at time $t' < t_k$, then the following is also a path:

$$P' = ((\sigma, t_1), \dots, (\mathcal{N}_i, t_i), \dots, (\mathcal{N}_j, t_j), (\delta, t')).$$

Note that no forwarding algorithm respecting minimal progress would take path P rather than path P' . Thus, path P is not a *first preference* path. Any path respecting loop avoidance and minimal progress (including first preference) is a *valid* path.

Second, in moderately-large datasets (such as those we work with), a complete enumeration of all valid paths with a given (σ, δ, t_1) is prohibitively expensive. Therefore, a key element of our approach is efficient enumeration of the most *important* paths. Our strategy for tackling this problem has two steps:

1. Define the problem as one of path enumeration on a space-time graph (defined below) and

2. At each time step, use dynamic programming to maintain the (up to) k shortest valid paths reaching each node. By ‘shortest’ we mean the path with the least number of hops.

This strategy allows us to determine the *optimal* path reaching the destination, that is, the path with shortest achievable duration under any forwarding algorithm. We denote the duration of the optimal path $\mathcal{T}(\sigma, \delta, t_1)$ and note that it is the minimum duration path found by epidemic forwarding, i.e.,

$$\mathcal{T}(\sigma, \delta, t_1) = \min_A \mathcal{T}_A(\sigma, \delta, t_1) = \mathcal{T}_{Epidemic}(\sigma, \delta, t_1).$$

This strategy also allows us to determine each subsequent valid path that reaches the destination, up until the point at which k or more valid paths reach the destination in a single timestep.

To recap, we want to answer the following questions: How many paths exist between a source-destination node pair? What are the properties of these paths?

To organize the search process, we convert the sequence of node contacts into a *space-time* graph, which is a special kind of directed weighted graph. Our use of the space-time graph is based on [MAZ04].

Time is discretized in increments of Δ . In all our work we use $\Delta = 10$ sec. Vertices in the space-time graph are pairs (\mathcal{N}_i, T) with $\mathcal{N}_i \in \mathbb{M}$ and $T = c\Delta$ for $c \in \{1, 2, \dots, \lfloor t_{max}/\Delta \rfloor\}$. Edges in the space-time graph come in two kinds:

1. There is an edge from vertex (\mathcal{N}_i, T) to (\mathcal{N}_j, T) iff node \mathcal{N}_i was in contact with node \mathcal{N}_j at any time during $[T - \Delta, T)$. Such an edge has weight zero.
2. There is an edge from vertex (\mathcal{N}_i, T) to $(\mathcal{N}_i, T + \Delta)$ for all $\mathcal{N}_i \in \mathbb{M}$. Such edges have unit weight.

For example, consider a network with three nodes. Nodes 1 and 2 are in contact during the first timestep, while all three nodes are in contact with each of the others during the

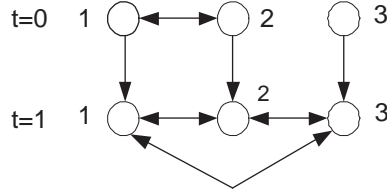


Figure 4.1: Example space-time graph.

second timestep. Then the corresponding space-time graph is as shown in Fig. 4.1. The horizontal edges have weight zero and the vertical ones have weight 1.

Given a space-time graph discretized by Δ and a message defined as (σ, δ, t_1) , we enumerate shortest paths using dynamic programming. The algorithm uses as its data structure an $N \times k$ array of paths, denoted P . It maintains the following invariant: at any given timestep $T = c\Delta$ the entry P_{ij} is the j^{th} shortest path reaching from (σ, t_1) to (\mathcal{N}_i, t') for some $t' \in [T - \Delta, T)$. The algorithm is given in Fig. 4.2.

4.3.1 Observations

Using the algorithm described in the previous subsection, we are able to enumerate paths for a given message with (σ, δ, t_1) up to the time when k paths reach the destination in a single timestep. For any given message, we will use T_n to denote the time at which the n^{th} path reaches the destination, in order of increasing delivery time.

Our results show that for many messages, the duration of the optimal path can be quite long (thousands of seconds). However, the main result is that in the vast majority of cases, once the first path reaches the destination a very large number of additional paths reach the destination soon thereafter – typically tens or a few hundreds of seconds later. We refer to this phenomenon generically as *path explosion*.

To make the notion concrete, we define the time that path explosion occurs to be the time by which 2000 paths in total have reached the destination, i.e., T_{2000} . We would like to point out that there is nothing sacrosanct about the number 2000; we believe that by enumerating 2000 paths (a large number of paths) we can develop a concrete picture of the

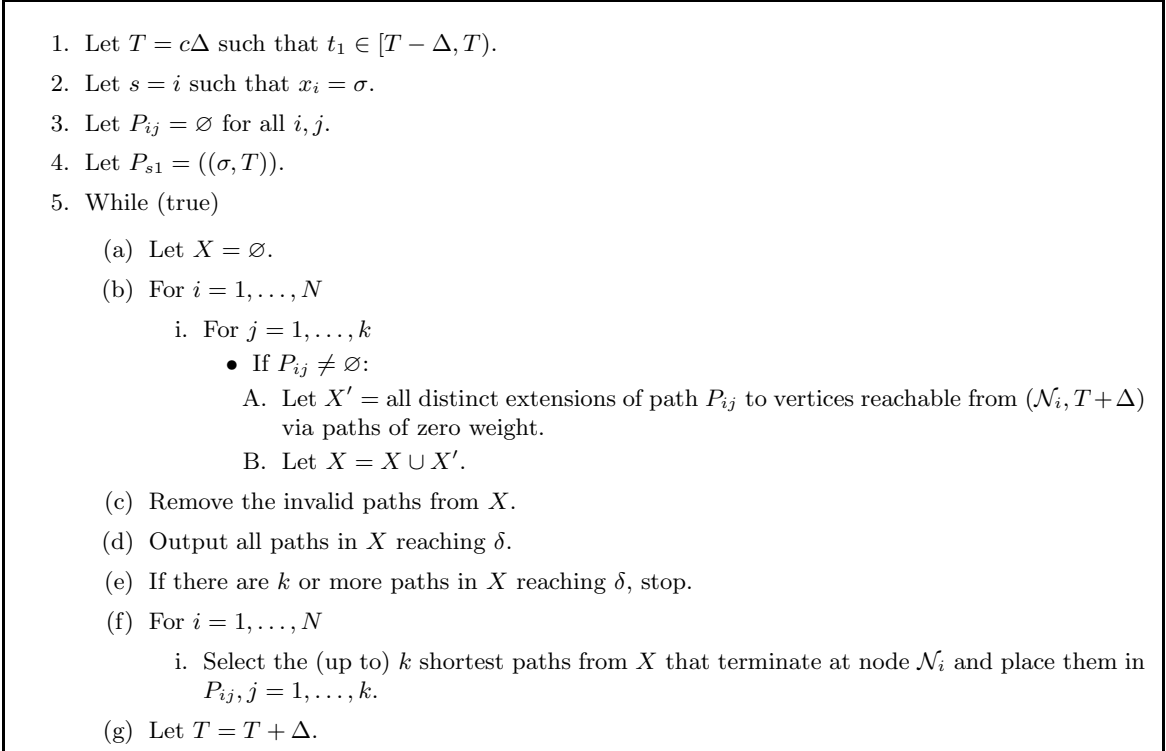


Figure 4.2: Algorithm for k shortest paths enumeration

set of paths available. Note that we can always identify this time accurately (to within an error of Δ) as long as we set k in our algorithm to be 2000 or greater. We refer to the ‘time to explosion’ (T_E) as the elapsed time between the arrival of the first path and the arrival of the 2000th path, i.e., $T_E = T_{2000} - T_1$. We would also like to point out that there is a possibility that some messages might not have 2000 paths, but may still have a fairly large number of paths between them. We do not consider such messages.

In Figure 4.3(a) we show the CDFs for optimal path duration in datasets Infocom ’06 9-12 and Infocom ’06 3-6. The figure shows that optimal path duration can be quite long; in fact, a large fraction of messages (over 25%) require over 1000 seconds before the first path reaches the destination. On the other hand, Figure 4.3(b) shows CDFs for the time to explosion. Here, the story is quite different: almost half the messages see an explosion with little or no delay, and 97% of them have T_E less than or equal to 150 secs.

These figures show the surprising fact that optimal path duration can be an order of

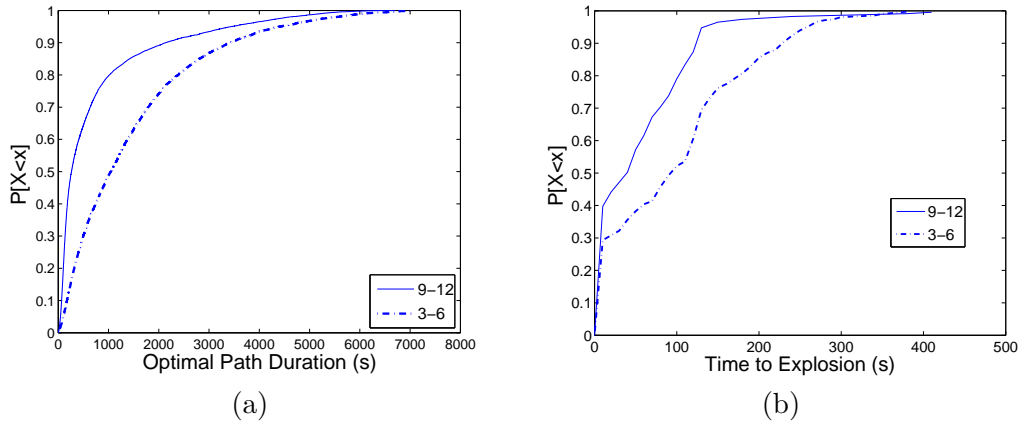


Figure 4.3: (a) CDFs of optimal path duration Infocom '06 9-12 and 3-6 (b) CDFs of time to explosion Infocom '06 9-12 and 3-6

magnitude or more greater than time to explosion. That is, in many cases, the first path takes a long time to reach the destination, after which many paths reach the destination in relatively short order. In a phrase, path explosion means that *shortly after the optimal path reaches the destination, there are a large number of nearly-optimal paths to the destination.*

A closer look at the path explosion phenomenon is shown in Figure 4.4, which is also based on dataset Infocom '06 9-12. Each point in the figure corresponds to (T_1, T_E) for a single message (σ, δ, t_1) (note the difference in scale on the x and y axes). The figure shows that there is no clear relationship between optimal path duration and time to explosion; there are many cases in which the optimal path reaches the destination quickly but the path explosion occurs comparatively late, and there are many cases in which the optimal path reaches the destination late while the path explosion occurs quickly thereafter.

It is difficult to characterize how the number of successful paths grows in time, in part because path explosion generally occurs quite rapidly and our time measures are rather coarsely discretized. However, we can get a rough sense of how the explosion occurs if we look at the slowest cases. In Figure 4.5 we examine all paths of the messages for which the time to explosion T_E was 150 seconds or greater, again using dataset Infocom '06 9-12. Note that although for each of these messages, the 2000th message arrives at least 150 seconds after the optimal time, nonetheless many messages still arrived during the period between

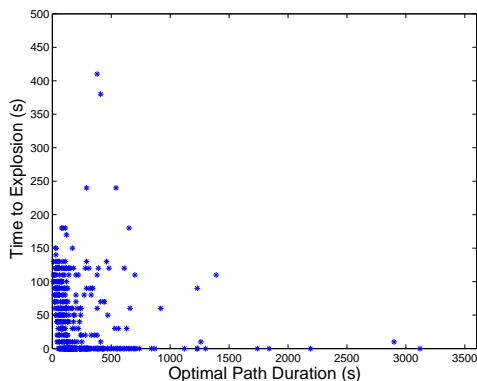


Figure 4.4: Optimal path duration vs. Time to Explosion, Infocom '06 9-12.

T_1 and T_{2000} . The figure shows that the number of paths arriving over time grows in an approximately exponential fashion.

In summary, we find that once paths are enumerated, a number of surprising properties emerge. While the optimal path duration can be long, there is a path explosion effect that occurs relatively quickly after the first path reaches the destination. However, there is no clear relationship between optimal path duration and time to explosion. Finally, it appears that the explosion process is roughly exponential in time. We seek to explain these observations in the following sections, and we study the implications of these observations in later chapters of this thesis.

4.4 Path Explosion

In this section we use models and data analysis to understand the questions raised in Section 4.3.1. In particular the next two subsections ask: (1) How and why does path explosion occur? and (2) What determines the relationship between optimal path duration and time to explosion? To answer the first question we develop an analytic model making the assumption of homogeneity – equal contact rates among all nodes. To answer the second question we incorporate considerations of unequal contact rates (inhomogeneity).

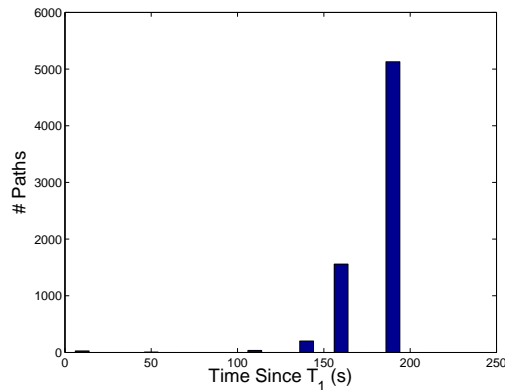


Figure 4.5: Histogram of number of paths for pairs with time to explosion ≥ 150 sec, Infocom '06 9-12.

4.4.1 A Homogenous Model for Path Explosion¹

As a first step, in trying to understand the path explosion phenomenon, we use a highly simplified model. We consider a setting in which nodes contact other nodes randomly and at constant rate. For a given message, we count the number of paths arriving at each node over time. The model is not restricted to counting just valid paths (which would add considerable complexity) but we seek qualitative results which are presumably insensitive to such details. The main result we show in this section is that the number of paths arriving at any node (such as the destination) grows exponentially in time, with rate given by the (homogeneous) contact rate between nodes.

Our results use known techniques from the study of epidemic process [Kur71, ZNKT06]; however, we extend those tools to study the set of all forwarding paths, instead of just the first path obtained to each destination.

Assumptions

Our model makes two assumptions, which derive from the uniformity of the setting.

- **Poisson contacts:** We associate with each node \mathcal{N}_n a process of contact opportunities with all other nodes that is a homogeneous Poisson process with intensity λ .

¹This section relies on work by collaborators; it is included for completeness.

- Homogeneity: Whenever a contact opportunity occurs for node \mathcal{N}_i , the contacted node \mathcal{N}_j is chosen independently from the past, and uniformly among the N nodes in \mathbb{M} . Note that this model does not include variability of contact rates among pairs, an issue we take up in Section 4.5.

Without loss of generality we study forwarding paths for a message created by a source (denoted by σ) at time $t = 0$.

Evolution and limit for large networks

Our model is adapted from the analysis of population dynamics. The *state* $S_n(t)$ of a node \mathcal{N}_n at time t is the number of paths from the source that reach \mathcal{N}_n before t . When \mathcal{N}_n has an opportunity to contact \mathcal{N}_m , the following transition occurs at \mathcal{N}_m :

$$S_m(t) \leftarrow S_m(t) + S_n(t).$$

For any $k \geq 0$, we denote

$$U_k(t) = \# \{ \mathcal{N}_n \in \mathbb{M} \mid S_n(t) = k \},$$

for the number of nodes with state k . To focus on the evolution of the population of nodes as a whole, we study the evolution of the collection of variables

$$U(t) \in \mathbb{N}^{\mathbb{N}}, U(t) = \{ U_k(t) \mid k \geq 0 \}.$$

Note that we have $\sum_{k \geq 0} U_k(t) = N$, so $U(t)/N$ represents the empirical density of nodes in each states. If one does not differentiate between nodes in the same state, then $U(t)$ entirely characterizes the system at time t .

Note that when node \mathcal{N}_n having state $i \neq 0$ contacts node \mathcal{N}_m having state j , the collection of variables U is modified as follows.

$$U_{i+j} \leftarrow U_{i+j} + 1 \quad ; \text{ and } \quad U_j \leftarrow U_j - 1.$$

We denote by $\lambda_{n,m}$ the contact rate between node \mathcal{N}_n and node \mathcal{N}_m . For any fixed $i > 0$ and $j \geq 0$, transitions of the type (i, j) as defined above occur with the following rate:

$$\beta_{i,j} = \sum \{ \lambda_{n,m} \mid n \neq m \text{ s.t. } S_n(t) = i, S_m(t) = j \}.$$

Proposition 4.4.1. The rate of transition $\beta_{i,j}$ is a function of the density of nodes in each state $U(t)/N$.

$$\beta_{i,j} = N \cdot \lambda \frac{U_i(t)}{N} \frac{U_j(t)}{N} = N \tilde{\beta}_{i,j} \left(\frac{U(t)}{N} \right)$$

Proof. Let us fix an n such that $S_n = i$. Then we have

$$\sum_{\{ \mathcal{N}_m \mid S_m(t)=j \}} \lambda_{n,m} = \lambda \frac{U_j(t)}{N},$$

since we have assumed that the contact process for each node \mathcal{N}_n has rate λ and is homogeneous with regard to other nodes.

Summing for all nodes \mathcal{N}_n having state i we obtain

$$\beta_{i,j} = U_i(t) \lambda \frac{U_j(t)}{N} = N \cdot \lambda \frac{U_i(t)}{N} \frac{U_j(t)}{N}$$

□

We can describe the process $U(t)$ taking on values in $\mathbb{N}^{\mathbb{N}}$ as a Markov jump process, with transitions indexed by $\{ (i, j) \mid i > 0, j \geq 0 \}$.

The rate of transition (i, j) is a function $\tilde{\beta}_{i,j}$ only of the density $U(t)/N$ of nodes having states i and states j , multiplied by N . In other words, the process $U(t)$ may be written as

$$U(0) + \sum_{i>0, j \geq 0} (e_{i+j} - e_j) \cdot \int_0^t \mathcal{N}_{i,j} \left(N \tilde{\beta}_{i,j} \left(\frac{U(s)}{N} \right) \right) ds$$

where, for all $i > 0, j \geq 0$, $\mathcal{N}_{i,j}$ denotes a Poisson counting process with intensity 1, and e_k denotes the infinite vector with all entries null except for 1 at position k .

When one considers the density process $U(t)/N$ in the case where N is large, Kurtz's limit theorem [Kur71] shows that one can replace the Poisson counting process in the above expression by the process mean. This means that the trajectory of the density process for large N closely approaches the solution of a deterministic ordinary differential equation.

Proposition 4.4.2. As N goes to infinity, if we assume that $U(0)/N \rightarrow u(0) > 0$, then we have for all $K \geq 0$:

$$\sup_{0 \leq s \leq t, 0 \leq k \leq K} \left| \frac{U_k(s)}{N} - u_k(s) \right| \rightarrow 0 \text{ a.s.}$$

where $u : [0; \infty[\rightarrow \mathbb{R}^{\mathbb{N}}$ is the solution of the ODE

$$u(t) = u(0) + \sum_{i>0, j \geq 0} (e_{i+j} - e_j) \cdot \int_0^t \tilde{\beta}_{i,j}(u(s)) ds$$

Proof. A priori the process $U(t)/N$ evolves in a space with an infinite number of dimensions $\mathbb{N}^{\mathbb{N}}$, hence Kurtz's limit theorem [Kur71] does not immediately apply to it. However, one can consider for any $K > 0$ a threshold process where all nodes in states $\{ k \mid k > K \}$ are collapsed into a collection of nodes in a single sink state. This threshold process has a finite number of dimensions, and satisfies all the assumptions of the Kurtz limit theorem. To complete the proof, we note that the threshold process defines exactly the same dynamics for states in $\{ 0, \dots, K \}$ as in the infinite process. \square

Proposition 4.4.3. For any $k \geq 0$, we have

$$u_k(t) = u_k(0) + \lambda \int_0^t \sum_{i=0}^k u_i(s) u_{k-i}(s) - u_k(s) ds \quad (4.1)$$

Proof. From the definition of u found in Proposition 4.4.2, we obtain that $u_k(t) - u_0(t)$ may be written for all $k \geq 0$ as:

$$\sum_{i>0, j \geq 0, i+j=k} \int_0^t \tilde{\beta}_{i,j}(u(s)) ds - \sum_{i>0, j=k} \int_0^t \tilde{\beta}_{i,j}(u(s)) ds$$

which may be rewritten as:

$$\begin{aligned} & \sum_{i=1}^k \int_0^t \tilde{\beta}_{i,k-i}(u(s)) ds - \sum_{i>0} \int_0^t \tilde{\beta}_{i,k}(u(s)) ds \\ & \sum_{i=1}^k \int_0^t \lambda u_i(s) u_{k-i}(s) ds - \sum_{i>0} \int_0^t \lambda u_i(s) u_k(s) ds \end{aligned}$$

After adding $\lambda u_0(s) u_k(s)$ to each of the above terms and using $\sum_{i \geq 0} u_i(s) = 1$, the proposition is proved. \square

In other words, from a non trivial initial condition, the average number of nodes with exactly n paths in a large network evolves according to a deterministic equation.

Solution for ODEs, moments

We introduce for all $x \in \mathbb{R}$ the series $\phi_x : t \mapsto \sum_{k \geq 0} x^k u_k(t)$. After multiplying each equation in (4.1) by x^k and summing all of them we obtain

$$\frac{d\phi_x}{dt} = \lambda(\phi_x^2 - \phi_x).$$

Note that this equation may be rewritten as:

$$\lambda \cdot dt = \frac{d\phi_x}{\phi_x(\phi_x - 1)} = \frac{d\phi_x}{\phi_x - 1} + \frac{d\phi_x}{\phi_x}.$$

This ordinary differential equation can be solved for all $x \in \mathbb{R}$. One needs to distinguish between two cases, depending on the initial value taken by ϕ_x .

- Assume $0 < \phi_x(0) < 1$ (for instance, by choosing x in $[0; 1]$). We know that ϕ_x and $1 - \phi_x$ remains positive. The previous equation may then be written as:

$$\ln(1 - \phi_x(t)) - \ln(\phi_x(t)) - \lambda t \text{ is constant.}$$

$$\text{Hence, } \phi_x(t) = \frac{\phi_x(0)}{\phi_x(0) + (1 - \phi_x(0))e^{\lambda t}}. \quad (4.2)$$

- On the other hand, assume $\phi_x(0) > 1$ (for instance, by choosing $x > 1$). We know that ϕ_x and $\phi_x - 1$ remains positive. The previous equation may then be written as:

$$\ln(\phi_x(t) - 1) - \ln(\phi_x(t)) - \lambda t \text{ is constant.}$$

$$\text{Hence, } \phi_x(t) = \frac{\phi_x(0)}{\phi_x(0) - (\phi_x(0) - 1)e^{\lambda t}}. \quad (4.3)$$

The first case above allows us to derive closed form expression for the evolution with time of the mean number of paths per node (see below), as well as the other moments. In the second case, for any $x > 1$ the series ϕ_x becomes infinite in finite time $T_C(x) = \frac{1}{\lambda} \ln \left(\frac{\phi_x(0)}{\phi_x(0) - 1} \right)$. In other words, if the initial distribution for the number of paths per node is light-tailed for a given coefficient, it loses this property within a finite time.

Mean number of paths According to the definition of ϕ_x , we can compute the expected number of paths for a node as follows.

$$\begin{aligned}\mathbb{E}[S_n(t)] &= \mathbb{E}\left[\frac{1}{N} \sum_{m \in \mathbb{M}} S_m(t)\right] = \sum_{k \geq 0} k \cdot u_k(t) \\ \text{thus } \mathbb{E}[S_n(t)] &= \left. \frac{\partial \phi_x(t)}{\partial x} \right|_{x=1} = \mathbb{E}[S_n(0)] e^{\lambda t}.\end{aligned}\tag{4.4}$$

The last equation is obtained from (4.2) since

$$\left. \frac{\partial \phi_x(t)}{\partial x} \right|_{x=1} = \frac{\frac{\partial \phi_x(0)}{\partial x}(x)}{D(x, t)} - \frac{\phi_x(0) \cdot \frac{\partial \phi_x(0)}{\partial x}(x) \cdot (1 - e^{\lambda t})}{D(x, t)^2},$$

with $D(x, t) = \phi_x(0) + (1 - \phi_x(0)) \cdot e^{\lambda t}$, so $D(1, t) = 1$ for all t .

Second moment for the number of paths It follows from the properties of ϕ_x that:

$$\begin{aligned}\mathbb{E}[S_n(t)(S_n(t) - 1)] &= \mathbb{E}\left[\frac{1}{N} \sum_{m \in \mathbb{M}} S_m(t)(S_m(t) - 1)\right] \\ &= \sum_{k \geq 0} k(k - 1) \cdot u_k(t) = \left. \frac{\partial^2 \phi_x(t)}{\partial^2 x} \right|_{x=1}.\end{aligned}$$

From the above expression of $\frac{\partial \phi_x(t)}{\partial x}$, one can obtain the second derivative in x . Again the expression simplifies when $x = 1$ and it implies.

$$\mathbb{E}[(S_n(t))^2] = \left(\mathbb{E}[(S_n(0))^2] + 2(e^{\lambda t} - 1)\mathbb{E}[S_n(0)]^2\right) \cdot e^{\lambda t}.$$

If $\mathbb{V}[X]$ denotes the variance of a random variable X ,

$$\text{then } \mathbb{V}[S_n(t)] = \mathbb{V}[S_n(0)] e^{2\lambda t} + \mathbb{E}[S_n(0)] \left(e^{2\lambda t} - e^{\lambda t}\right).$$

Note that the variance of the number of paths increases exponentially fast with time t as $\mathbb{E}[S_n(0)] e^{2\lambda t}$, even if it is null when $t = 0$.

4.5 The Effects of Inhomogeneity

The analysis leading to Equation 4.4 sheds considerable light, but does not explain all the phenomena noted in Section 4.3.1. The analytic results confirm that path explosion *can*

occur and that it should be exponential in nature. However, the analysis in Section 4.4.1 would also predict that optimal paths should be short, and that path explosion should occur immediately after the first path reaches the destination. These two predictions are not borne out in the data, as already shown in Section 4.3.1. To explain these phenomena, we need to examine the model assumptions.

The most unrealistic aspect of our analytic model is the assumption that all nodes contact each other at the same rate. In fact, prior work has shown that per-node contact rates in datasets like ours can be highly variable across different nodes. For example, the authors in [HCS⁺05] show that the distributional tails of inter-contact times for such data sets approximately follow a power law.

In Chapter 3, we plotted the CDF for the total number of contacts each node has over each three hour period (Infocom '06 9-12, 3-6, Conext 9-12, and 3-6). We showed that the contact rates are highly variable and there exist some nodes which have a high number of contacts and some nodes that have very few number of contacts.

Our analytic results suggest that for a message generated at time 0, the number of paths reaching node i will grow proportional to $e^{\lambda t}$ for a population in which all nodes have the same rate λ .

To discuss the case where nodes have different rates, we introduce some notation. We define the random variable $T(\sigma, \delta)$ to be equal to $\mathcal{T}(\sigma, \delta, t_1)$ when t_1 is chosen uniformly at random in $[0, t_{max})$. Furthermore we no longer assume equal contact rates, so we define λ_i to be the contact rate of $\mathcal{N}_i \in \mathbb{M}$, i.e., $\lambda_i = \sum_k \lambda_{i,k}$. Finally we define the expected time for the first path as

$$H = \mathbb{E}[t | S_i(t) = 1].$$

In the case of the homogeneous model in Section 4.4.1,

$$H = (t | \mathbb{E}[S_i(0)] e^{\lambda t} = 1) = \frac{\ln N}{\lambda}$$

since $\mathbb{E}[S_i(0)] = 1/N$.

To understand the case when nodes have different rates, we reason as follows. Assume

that at time t_0 a message is held by a node \mathcal{N}_i having contact rate λ_i . Then at minimum, we expect a path explosion to occur starting at t_0 with rate λ_i among the subset of nodes with contact rates greater than or equal to λ_i . That is, we can infer a lower bound on $\mathbb{E}[S_j(t - t_0)]$ proportional to $e^{\lambda_i(t-t_0)}$ for all j such that $\lambda_j \geq \lambda_i$. This *subset* path explosion means that the number of paths arriving at nodes with rate λ_i or greater grows at least as fast as $e^{\lambda_i t}$.

Now if λ_i is relatively small compared to typical contact rates, then initially the subset path explosion may proceed very slowly. However if at time $t_1 > t_0$, \mathcal{N}_i encounters a node \mathcal{N}_j with contact rate $\lambda_j > \lambda_i$, then a more rapid path explosion beginning at t_1 will occur with rate λ_j among nodes with contact rates λ_j or greater.

Thus, if the source is a low-rate node, then there will be some initial time before it encounters a high-rate node, at which point a high-rate path explosion occurs. This initial time is related to H and so we argue that the time until high-rate path explosion occurs is on the order of $1/\lambda_i$. Thus in this case, T_1 will tend to be larger than is typical.

Furthermore, if the destination is a low-rate node, then path explosion may not be able to reach a high rate. Thus in this case, T_E will tend to be large.

This model suggests that a critically important role is played by the contact rates of the source and destination nodes. Then we can make the following hypotheses concerning the relative sizes of T_1 and T_E for four situations:

- First, when λ_σ is high and λ_δ is high, then path explosion begins immediately and at high rate. So both T_1 and T_E will tend to be small.
- Next, when λ_σ is high and λ_δ is low, then path explosion begins immediately but at a low rate. So T_1 will tend to be small but T_E will tend to be large.
- Next, when λ_σ is low and λ_δ is high, then there is a significant period before path explosion begins (on the order of $1/\lambda_\sigma$) but, once begun, path explosion proceeds at high rate. So T_1 will tend to be large but T_E will tend to be small.
- Finally, when λ_σ is low and λ_δ is low, then both T_1 and T_E will tend to be large.

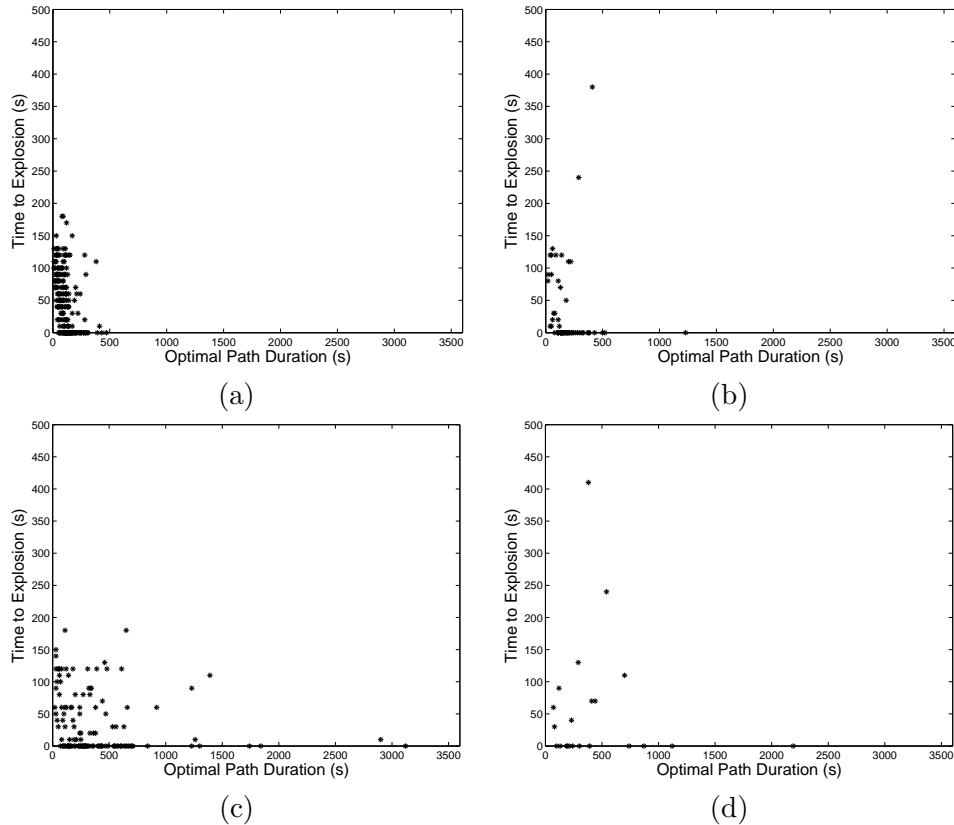


Figure 4.6: Path Explosion Plots Info 06 9-12 (a) in-in (b) in-out (c) out-in (d) out-out.

To explore whether these hypotheses hold in our data, we separated nodes in each dataset into two equal-sized groups. The *in* set are those nodes with contact rates greater than the median rate. The *out* set are those nodes with contact rates lower than the median rate. Since rate distribution is approximately uniform (as shown in Chapter 3), the median rate in each case is approximately half of the maximum rate.

Then each message (σ, δ, t_1) can be placed into one of four cases: *in-in*, where λ_σ and λ_δ are both high; *in-out*, where λ_σ is high and λ_δ is low; *out-in*, where λ_σ is low and λ_δ is high; and *out-out*, where λ_σ and λ_δ are both low.

Using this labeling, we separate the points in Fig. 4.4 into four groups, which are plotted separately in Fig. 4.6. Each plot has the same format as Fig. 4.4, except that only a subset of points are included.

The figures suggest that our hypotheses hold to a certain extent. In particular,

- For all the in-in messages, the optimal path duration T_1 is very low and the time to explosion T_E is less than 150 seconds.
- For the in-out messages, the optimal path duration T_1 is similar to the in-in messages, i.e., small. However the time to explosion T_E is much more variable and can take on relatively large values.
- For the out-in case, the optimal path duration T_1 tends to be larger than in the first two cases, while the time to explosion T_E is relatively small (as in the in-in case).
- Finally, for the out-out case, both T_1 and T_E can take on large values.

Results for other datasets are not shown due to lack of space, but generally showed similar behavior.

In summary, we have shown two key results in this section. We have used an homogeneous analytical model to show how path explosion occurs. We have shown that path explosion is an exponential function of both elapsed time and the rate λ at which nodes come into contact. This addresses the first question about our empirical results, namely, how and why does path explosion occur? Second, we have shed light on the other set of questions arising from our empirical results by considering the effects of inhomogeneity in rates. We have shown that by looking at the contact rates of the source and destination nodes, one can gain insight into why the optimal path duration and the time to explosion vary, and one can relate these quantities to the relative magnitudes of the two contact rates.

4.6 Summary

In order to understand the nature of the forwarding problem, we have taken the novel approach of enumerating and characterizing the set of forwarding paths that are available in settings under which MONs operate under.

The main result presented in this chapter is the observation that for most messages there is a path explosion effect, meaning that while the optimal path may be long, there

are usually a large number of nearly-optimal paths to the destination. We support this result both analytically and empirically. We present a simple model of homogeneous inter-node contact and show that path explosion is to be expected in that case. And we show empirically that path explosion does occur in our data, although in a more complicated manner due to varying contact rates across nodes. Our empirical results focus attention on the difference between high contact rate ('in') nodes and low contact rate ('out') nodes.

Armed with the understanding of the nature of forwarding, we study different forwarding algorithms in the next chapter.

Chapter 5

Evaluating Forwarding Approaches

In the previous chapter, we have developed insight into what sorts of paths are available for use by forwarding algorithms in settings described in Chapter 1. This has laid the groundwork for an understanding of how various forwarding algorithms perform in this setting. In this chapter, we study the performance of a wide range of forwarding algorithms and relate their performance to the results developed in previous chapter.

This chapter is organized as follows. In Section 5.1 we describe in detail the trace driven methodology we used to simulate and study different algorithms. We will use trace-driven simulations extensively for the rest of the thesis, so the details in this section are important for understanding results presented in later chapters as well. We describe the algorithms we study in Section 5.2. Some of the forwarding algorithms we develop in this chapter are used in subsequent chapters as well. In Section 5.3, we discuss the results we obtain from the trace-driven simulations. We show that performance of the different algorithms we consider are roughly similar. We then investigate the reasons behind this observation, and find connections to the path explosion effect described in Chapter 4. Based on the observations, we focus on one forwarding scheme, forwarding to the highest contact rate node, and we show that such a scheme can be effective in Section 5.4. We conclude in Section 5.5.

5.1 Trace-Driven Simulations

In order to study and compare different forwarding algorithms, we develop a trace-driven simulator and implement these algorithms in the simulator. We use as input to our simulator the contact traces described in Chapter 3. Since the traces we study are either 3 or 6 hours long, each simulation therefore ran for simulated time of either 3 or 6 hours.

For each trace and forwarding algorithm, we generate a set of messages (σ, δ, t_1) uniformly at random. We generate messages according to a Poisson process with rate one message per 4 seconds. As discussed in Chapter 4 we model nodes as having infinite buffers and nodes carry all the messages they receive till the end of the simulation. We also assume infinite bandwidth; we disregard contact durations for the simulations and assume that message transfer happens in an instant and this transfer is not prone to channel effects and other physical layer idiosyncrasies. We however modify this for Chapter 7, where we study the effect of finite buffer sizes and finite bandwidth.

However there are a few special additions to the simulator that are worth mentioning:

- If a node with a message encounters the destination, and the destination has not yet received the message, the message is delivered.
- If a node with a message encounters the destination, and the destination has received the message, the message is not delivered. In addition, the node stops replicating and forwarding this message.

All our results are averaged over 10 simulation runs. In this chapter, the metrics under consideration are success rate S and average delay D as defined in Chapter 4. To avoid end-effects, we generated messages for the first 2 hours for three hour traces and the first 5 hours for the 6 hour traces used that collection of messages for simulation purposes.

To summarize, we use a discrete-event simulator, which takes as input contact traces that contain real contact events and messages generated in the simulator are replicated and transmitted during these contact events. However in order to decide if and when messages

are replicated and transmitted, we need forwarding algorithms to make these decisions. We discuss the algorithms next.

5.2 Algorithms

Our goal in this section is not to determine which forwarding algorithm is “best” but to compare the performance of a wide range of algorithms and gain insight into what properties of a forwarding algorithm yield good performance. Hence we choose a set of algorithms designed to span a range of design choices. However some of the algorithms we detail here are revisited in later chapters as well.

The design choices include:

- Destination aware vs Destination unaware. Destination aware algorithms take the choice of δ into consideration in forwarding, while destination unaware algorithms do not. There is an explicit information trade-off here - the latter schemes need less information.
- Single hop vs Multi hop Single hop algorithms use information about the most recent contact or next expected contact; multi hop algorithms take into account sequences of past or expected future contacts.
- Complete history vs Recent history vs Future knowledge. Complete history algorithms take into account the entire past history of other nodes when forwarding. Recent history algorithms taken into account only a limited amount of history (e.g., only the most recent encounter with the destination). Future knowledge algorithms make use of oracles that provide knowledge of future behavior of the nodes. Future knowledge algorithms are not practical but provide useful comparison cases. These algorithms use information implicitly, through aggregate statistics of contact rates seen over the entire experiment.

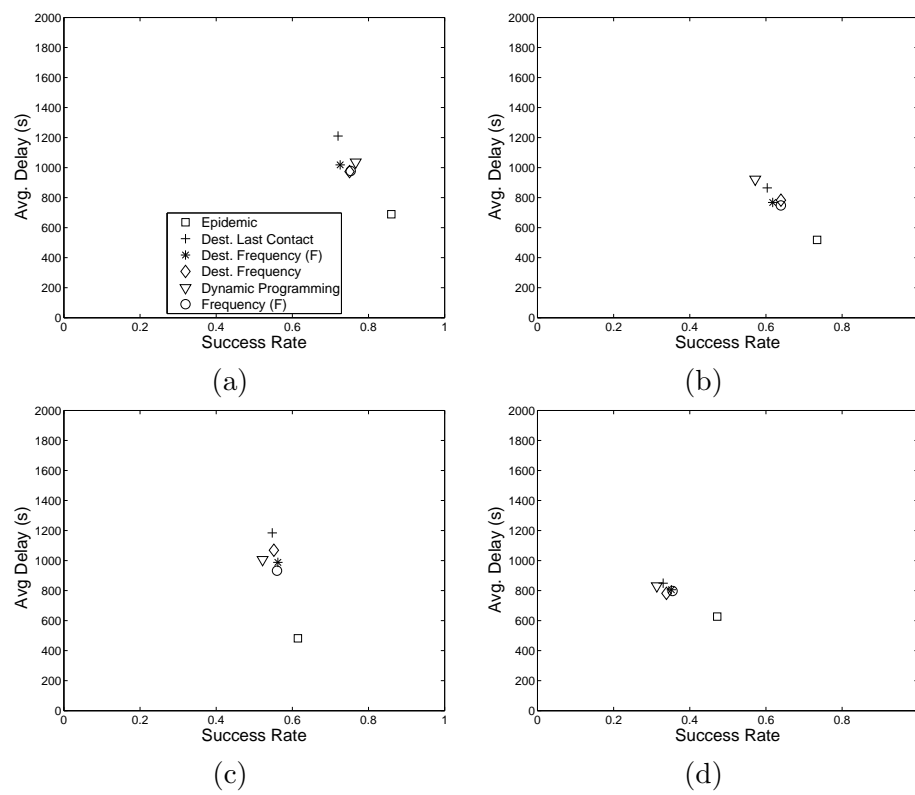


Figure 5.1: Average Delay (sec) vs Success Rate (a) Infocom '06 9-12 (b) Infocom '06 3-6 (c) Conext 9-12 (d) Conext 3-6

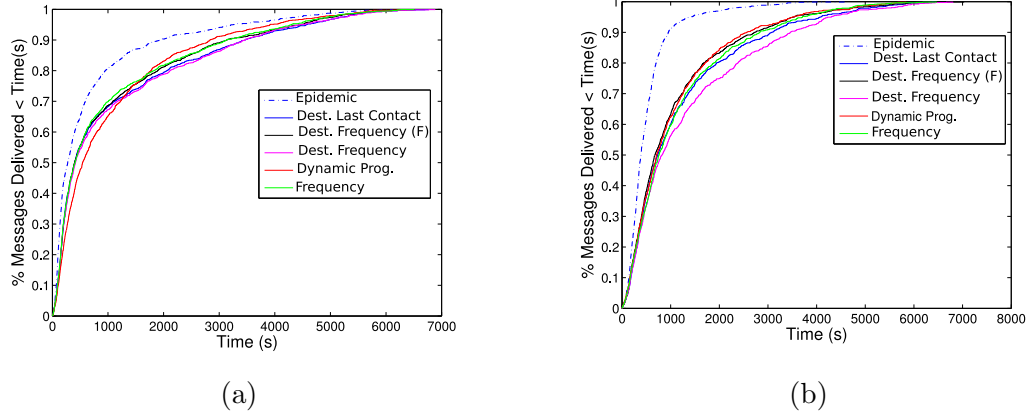


Figure 5.2: Delay Distributions (a) Infocom '06 9-12 (b) Conext '06 9-12

To span these design choices we use the following forwarding algorithms. Some of these algorithms are known from the literature and others are modifications or extensions.

- Epidemic (Flooding) [VB00]: Node \mathcal{N}_i forwards \mathcal{M}_m to \mathcal{N}_j unless \mathcal{N}_j already has a replica of \mathcal{M}_m . Epidemic forwarding achieves the best possible performance, so this algorithm yields upper bounds on success rate and average delay. However it is also the case that epidemic forwarding will have the highest costs.
- Frequency [ECCD07]: Node \mathcal{N}_i forwards \mathcal{M}_m to node \mathcal{N}_j if \mathcal{N}_j has more total contacts (with all other nodes) than does \mathcal{N}_i . This algorithm is destination independent. We use two variants of this - an online version and a version which incorporates future contacts as well (the version we hyphenate with 'F').
- Destination Frequency: Node \mathcal{N}_i forwards \mathcal{M}_m to \mathcal{N}_j if \mathcal{N}_j has contacted \mathcal{M}_m 's destination more often than has \mathcal{N}_i . We use two versions of this - one which incorporates future contacts (F) and one which is purely online; the statistics are updated with the simulation.
- Destination Last Contact [DFGV03]: Node \mathcal{N}_i forwards \mathcal{M}_m to \mathcal{N}_j if \mathcal{N}_j has con-

tacted \mathcal{M}_m 's destination more recently than has \mathcal{N}_i . This algorithm is also known as FRESH [DFGV03]. This algorithm incorporates temporal information, as compared to Frequency based algorithms noted above.

- **Dynamic Programming:** Node \mathcal{N}_i calculates the average delay between all pairs of nodes then finds the optimal path. This is based on the Minimum Expected Delay algorithm [JFP04]. Dynamic Programming is based on both past and future knowledge of internode contacts.

As can be seen our algorithms span different design choices. These algorithms will be revisited again in later chapters, and new algorithms will be considered.

5.3 Observations

In this section, we describe the results obtained after simulating the algorithms described in the previous section.

5.3.1 Similarity of Performance

The most striking aspect of our results is the similarity in performance of the various different forwarding algorithms. This is illustrated in Figure 5.1. For each data set and forwarding algorithm we plot the average delay vs success rate.

The figure shows that almost all forwarding algorithms show virtually identical performance. The exception is epidemic routing (shown using square symbols in the plots) which shows somewhat better performance than the others, since it always finds the optimal path if one exists.

While the plots in Fig. 5.1 show the similarity of average delay, a more detailed view is given in Fig. 5.2. This figure shows the entire distribution of delay. The figure shows that the different algorithms show quite similar distributions of delay as well.

To explore reasons behind the similar performance of the different algorithms, we first verify that message delivery is not ‘bursty’. That is, we confirm that the phenomenon is not

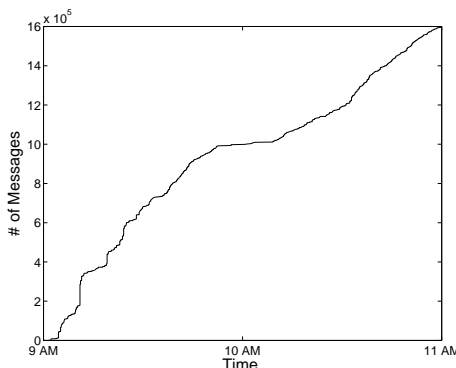


Figure 5.3: Message Reception Times Infocom '06 9-12

simply due to many nodes all making contact at relatively infrequent times. An example scenario in the conference could be that participants convened at a common area after a session and it was during this time period that many packets were delivered. To confirm that this is not the case, we plot the cumulative totals of in Fig. 5.3. We plot the cumulative totals of delivery times of all optimal and near-optimal paths. As can be seen, from the roughly constant slope, the delivery rate is fairly uniform in time. This shows that there are no special periods of time in which most messages are delivered.

Likewise, the fact that epidemic routing does noticeably better than the other algorithms means that similarity in performance among algorithms does not occur because they are all finding the same, optimal paths. That is, in many cases there are better paths to be found than those chosen by most forwarding algorithms. However algorithms generally seem to find paths that are close to, though not exactly, optimal.

To understand the remarkably similar performance of the many different forwarding algorithms we examine *which* paths are being taken by the different algorithms. Some typical results are shown in Fig. 5.4. Each plot in the figure corresponds to a particular message (σ, δ, t_1) . The x axis measures the time since T_1 for this particular message, and the y axis shows the number of paths that reach the destination over time. Furthermore, for each algorithm, its symbol is superimposed on the bar corresponding to the path chosen by the algorithm.

The two cases in the figure show the typical path explosion phenomenon — the number

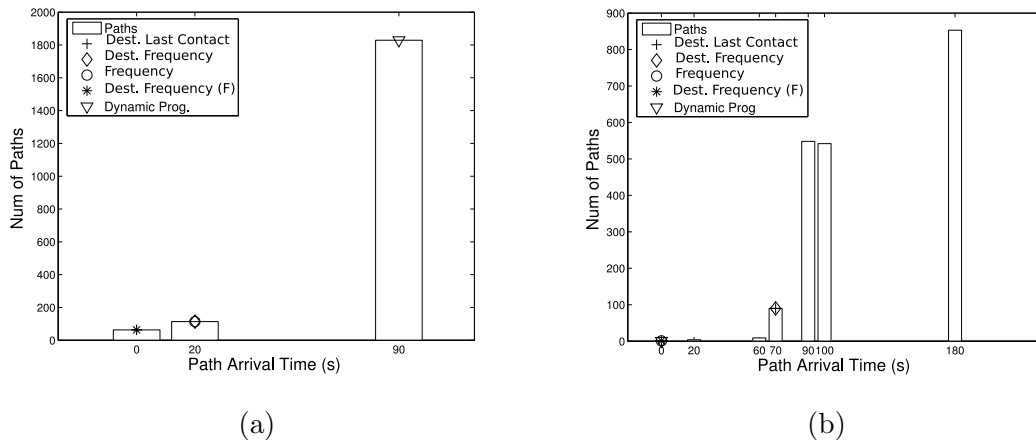


Figure 5.4: Paths taken by forwarding algorithms for two messages in Infocom '06 9-12 (a) Msg 88-82 (b) Msg 86-26

of paths arriving at the destination grows approximately exponentially with time. Furthermore, the plots show that the paths used by all the forwarding algorithms reach the destination early in the path explosion process.

For example, the left hand plot shows that for this particular message, Destination Last Contact and Destination Frequency (F) are able to find paths arriving at time T_1 (i.e., optimal paths) while Destination Frequency and Frequency (F) find paths arriving in the next burst, 20 sec after T_1 . Finally Dynamic Programming finds a path arriving in the third burst at 90 seconds after T_1 . The situation is similar for the case of the message in the right hand plot.

This figure suggests that different algorithms may have similar (though non-optimal) performance because of the large number of paths that reach the destination *shortly after* T_1 . When there are a large number of nearly-optimal paths, a forwarding algorithm may be able to find one of those paths relatively easily. That is, the existence of path explosion may be a factor that allows many algorithms to achieve somewhat similar performance.

If this hypothesis is correct, then the performance of forwarding algorithms should be strongly influenced by the way in which path explosion occurs. In Chapter 4 we found that

path explosion occurs in different ways for different types of source-destination pairs, i.e., in-in, in-out, out-in, and out-out. Here we can put that insight to use.

If the path explosion effect is a major reason why different forwarding algorithms have similar performance, then algorithm performance should be fairly similar within pair types and quite different between pair types. For example, algorithm performance should be similar across all in-in pairs, but performance on in-in pairs should be quite different from that on in-out pairs.

To see whether this is the case, we separate our simulation results by pair type. The results are shown in Fig. 5.5. The figure shows the average delay and success rate across the four pair types for each of the six forwarding algorithms.

The figure shows that success and delay depend primarily on the type of the source-destination pair as opposed to the type of algorithms used. All forwarding algorithms, with the exception of epidemic, again show similar performance.

However the figure also shows that, once messages are broken down by pair type, differences between forwarding algorithms start to emerge. In particular, we note a difference for algorithms making use of maximum information about contact patterns – i.e., Destination Frequency (F) and Dynamic Programming, which use both past and future information. These algorithms outperform others, but *only* in the case where one node is an ‘out’ node. That is, in the case where both nodes are ‘in’ nodes, then information about contact patterns is not particularly helpful. However, when one or both nodes are ‘out’ nodes, then maximum information about contact patterns is helpful.

Digging deeper, we see that Frequency (F) performs particularly well when the source node is an ‘out’ node. Recalling the discussion in Chapter 4, when a source node has low rate λ_σ , then rapid path explosion does not occur until the message has been moved to a high rate node with $\lambda_i > \lambda_\sigma$. This is consistent with the strategy of Frequency (F). Note that Frequency (F) is a destination-unaware strategy — it only seeks to move the message toward nodes with higher contact rates, i.e., those with $\lambda_i > \lambda_\sigma$.

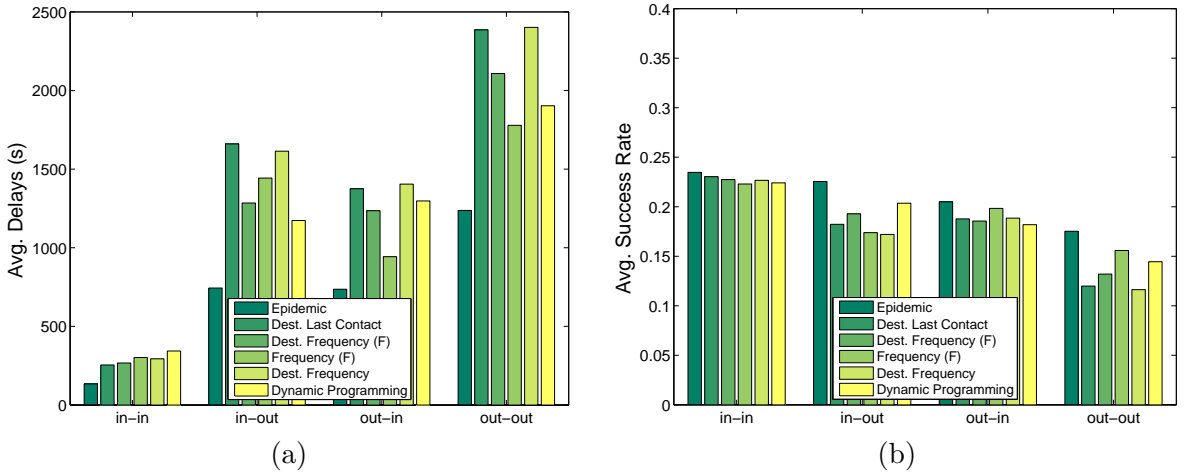


Figure 5.5: Performance of Forwarding Algorithms on Different Types of Source-Destination Pairs: (a) Average Delay (b) Success Rate.

5.4 Effective Forwarding

These results suggest some heuristics for effective forwarding in settings like pocket switched networks. In particular, they suggest that a forwarding algorithm will be successful if it causes path explosion to take place as quickly as possible. This is a somewhat different principle for forwarding than has typically been used in past proposals. That is, rather than seeking to find a short path to the destination directly, a forwarding algorithm may instead work to cause path explosion to occur as quickly as possible. This suggests forwarding toward high rate nodes preferentially, regardless of their relationship to the destination. We can use our enumeration of paths to check whether forwarding toward high rate nodes is an effective strategy. If it is, then the nodes making up the near-optimal paths should tend to increase in rate along the path. That is, hops along successful paths should tend to be from lower-rate nodes to higher-rate nodes.

We test this hypothesis by inspecting the contact rate of nodes appearing at each hop along near-optimal paths. Note that for different paths, message delivery will occur after different numbers of hops, so the set of nodes at any given hop may contain nodes at the beginning, middle, or end of their respective paths. Nonetheless this aggregate view can

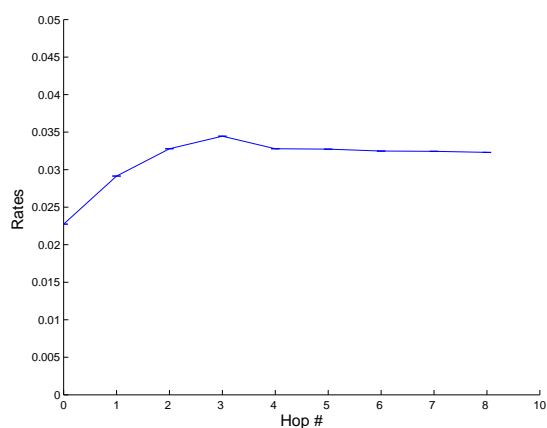


Figure 5.6: Mean rates of Intermediate Nodes Rates Infocom '06 9-12

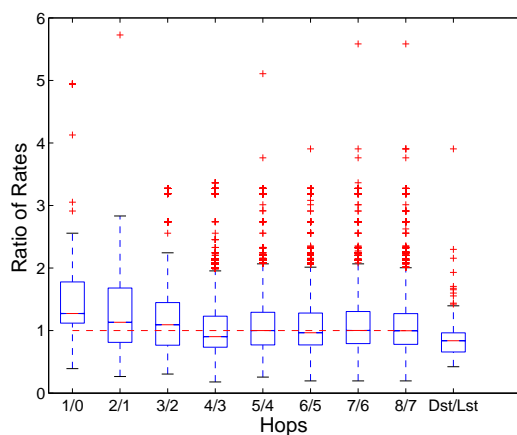


Figure 5.7: Mean rates of Intermediate Nodes BoxPlots (Ratios) Infocom '06 9-12

tell us if any general trend is present. The results are shown in Fig. 5.6, which plots the mean of node contact rates along the near-optimal paths for the Infocom 2006 9-12 dataset. Error bars are plotted corresponding to a 99% confidence interval on the mean; in many cases the confidence intervals are too small to see. The figure shows that for the first three hops of near-optimal paths, the contact rates of the nodes tends to increase. After a slight drop-off at hop 4, the mean contact rate then levels off. The reasons for the drop at hop 4 are not clear, but it may be because by that point most messages have been delivered and sample sizes are small.

While Fig. 5.6 gives an aggregate view of nodes on forwarding paths, it does not indicate whether individual paths tend to progress from lower-rate toward higher-rate nodes. To verify that this is indeed the case, we look at the ratio of rates $r = \lambda_j/\lambda_i$ for consecutive nodes i, j along near-optimal paths. If our hypotheses is correct, then these ratios will tend to be greater than 1. We plot the results as box-and-whiskers plots in Fig. 5.7. These plots show the distribution of r values for successive hops. For each hop, the box shows the limits of the 25 and 75 percentiles of the distribution and the median is marked. The figure shows that nearly all of the first hops are to nodes with higher rate than the source. Likewise, the second and third hops tend to be toward higher-rate nodes as well.

The picture that emerges from the results in this chapter gives a better view of the nature of node mobility, contact patterns, and forwarding performance in a conference type setting. In broad terms, the picture is as follows. Connection patterns between nodes lead to an exponential path explosion effect whose rate depends on the contact rates of nodes. Since contact rate varies considerably across nodes, path explosion occurs much faster among the higher rate nodes than the lower rate nodes. In this setting successful forwarding relies on moving the message toward high contact rate nodes so that path explosion can occur quickly. Once path explosion occurs among high contact rate nodes, contact between one of them and the destination leads to message delivery.

5.5 Summary

In this chapter, we studied the performance of different forwarding algorithms using real mobility traces. A principal observation is that most of these algorithms show similar performance, where performance is defined as success rate and delay. This led us to investigate the reasons for this similar performance, and we found the path explosion effect (described earlier in Chapter 4) is a driver for the observed similarity of performance. Due to the path explosion effect, most of the algorithms we studied find one of the many sub-optimal paths. Our results also prompted us to investigate one particular scheme in detail – namely, that

of always forwarding to the highest rate node. We find that while most of the algorithms perform similarly in terms of success rate and delay, they do differ in terms of cost incurred. We therefore focus our attention on cost in the next chapter.

Chapter 6

Reducing Forwarding Costs: Delegation Forwarding

From the previous chapters, we have learned that forwarding algorithms spanning different design choices do not differ much in terms of performance, that is, success rate and average delay. We explain this with the help of the path explosion phenomenon we described in Chapter 4. The focus therefore, shifts to cost of forwarding algorithms.

One of the biggest challenges in designing effective forwarding algorithms for mobile opportunistic networks is the scarcity of resources. Operating under such constraints, one of the main goals for any forwarding algorithm would be to be effective, while reducing costs in the network. This chapter describes our attempt at designing forwarding algorithms which leave a very small cost footprint in the networks we study. We refer to the algorithm we develop as *Delegation Forwarding*.

6.1 Overview

Forwarding algorithms can be placed on a spectrum from epidemic forwarding [VB00] which relies on flooding the network with messages to wait-for-destination scheme or one-hop relay in which a source node forwards only if it encounters the destination. While the former scheme guarantees delivery of the message if a path exists, it comes at a high cost; the latter scheme has the least cost but also has a low success rate.

Most forwarding algorithms seek to find a middle ground between these two extremes

by relying on information that can be learned during contacts. Algorithms differ in the type of information used as well as how it is used; however, many algorithms make use of some kind of forwarding metric. We refer generically to the value of a node’s metric as its *quality*. At any contact, a node with a lower quality metric will forward messages to the node with higher quality. Examples of this include FRESH [DFGV03], in which a node will forward if it encounters another node that has seen the destination more recently; greedy-total [ECCD07] in which a node will forward if it encounters nodes with a higher contact rate than itself and SimBet routing [DH07] which relies on a metric calculated using social analysis techniques. In essence, most algorithms try to select good intermediate nodes for forwarding using only local information while endeavoring to minimize costs. A secondary goal is to ensure fairness by balancing costs across nodes.

This work starts from the observation that to reduce costs, we might seek to forward *only* to the *highest*-quality nodes. This suggests that the problem consists of making timely forwarding decisions by observing a sequence of samples. The main contribution is a new forwarding strategy based on this observation that is explicitly designed to reduce costs while achieving high performance. We refer to this as *delegation forwarding*.

The main idea of delegation forwarding is as follows. We assume each node has an associated quality metric. A node will forward a message only if it encounters another node whose quality metric is greater than any seen by the message so far. We show that despite the simplicity of this strategy, it works surprisingly well.

This chapter is organized as follows. We first formalize the problem of minimum cost forwarding and describe *delegation forwarding* in Section 6.2. We devote Section 6.3 to analyze delegation forwarding where we show that in a N -node network, delegation forwarding has expected cost $O(\sqrt{N})$ while the naive scheme of forwarding to any higher quality node has expected cost $O(N)$. We next discuss the issue of cost imbalance under the presence of delegation forwarding in Section 6.4. In order to investigate whether delegation schemes are effective in real-world scenarios, we study delegation forwarding on real mobility traces in Section 6.5. We find that delegation forwarding shows performance as good as other

schemes at a much lower cost. We compare algorithms in terms of performance (rate of successful delivery and mean delivery delay) as well as cost (number of message replicas created). We also look into cost imbalance on a per-node basis of different algorithms. We find that while most delegation strategies do a good job in balancing cost, the delegation strategy with destination contact rate as the metric does very well. We summarize our findings in Section 6.6.

6.2 Delegation Forwarding

We work in the following setting: we assume a set of mobile nodes $\mathcal{N}_i \in \mathbb{M}$ with $|\mathbb{M}| = N$. Nodes generate messages over time; each message has a particular source $\sigma \in \mathbb{M}$ and destination $\delta \in \mathbb{M}$. At random times nodes come into contact, meaning that they are capable of exchanging messages. Messages are transmitted in whole from node to node at time instants during node contact intervals, after which both nodes hold message *replicas*. In our analysis we make no assumptions about the time instants when messages are generated or the time needed for transmission; in our simulations we generate messages according to a Poisson process, and messages are transmitted with no transmission lag. Nodes do not possess any *a priori* knowledge of the number of nodes in the system or knowledge of any properties of the other nodes.

The metrics we are concerned with are the ones we defined in Chapter 4. They are: (1) *cost*, which is the number of replicas per generated message in the network; (2) *success rate*, which is the fraction of generated messages for which at least one replica is eventually delivered; and (3) *average delay*, which is the average duration between a message’s generation and the first arrival of one of its replicas at the destination. By “high performance” we mean high success rate and low average delay. Furthermore, we distinguish per-node cost variants: (1) *node transmission load*, which is the number of message replicas a node has to forward and (2) *node memory load*, which is the number of message replicas a node has to store in its buffer.

For any given node \mathcal{N}_i , the forwarding problem in this setting reduces to the simple question: “upon contact with node \mathcal{N}_j , which (if any) of the messages I am holding should I forward to \mathcal{N}_j ?” As stated in Section 6.1, we abstract the information available during a contact event to a quality metric associated with each node and message. By the nature of the metric, moving the message to a node with higher quality for this message makes the message more likely to be delivered.

For many algorithms, the answer to the forwarding question is “forward message m if \mathcal{N}_j ’s quality for message m is higher than mine.” This is a tradeoff between the high cost of flooding the network and the low success rate of waiting to encounter the destination. However, the cost of this approach can still be quite high, as we show in Section 6.3.

Algorithm 1 Delegation Forwarding

```

Let  $\mathcal{N}_1, \dots, \mathcal{N}_N$  be nodes
Let  $\mathcal{M}_1, \dots, \mathcal{M}_M$  be messages
Node  $\mathcal{N}_i$  has quality  $x_{im}$  and threshold  $\tau_{im}$  for  $\mathcal{M}_m$ .
INITIALIZE  $\forall i, m : \tau_{im} \leftarrow x_{im}$ 
On contact between  $\mathcal{N}_i$  and node  $\mathcal{N}_j$ :
for  $m$  in  $1, \dots, M$  do
  if  $\mathcal{M}_m$  is currently held by  $\mathcal{N}_i$  then
    if  $\tau_{im} < x_{jm}$  then
       $\tau_{im} \leftarrow x_{jm}$ 
      if  $\mathcal{N}_j$  does not have  $\mathcal{M}_m$  then
        forward  $\mathcal{M}_m$  from  $\mathcal{N}_i$  to  $\mathcal{N}_j$ 
      end if
    end if
  end if
end for

```

To reduce costs even more, we make the requirement for forwarding more stringent. Our approach seeks to forward the message only to the *highest* quality nodes in the system. Conceptually, we would like to create a small number of replica copies, and place them with the nodes which are the very best candidates for eventual delivery to the destination. Thus the forwarding question in our approach becomes “is \mathcal{N}_j among the very highest quality nodes for message m ?”

Since there is no *a priori* or global knowledge of node quality, our forwarding question

is an instance of an *optimal stopping problem* [Shi08]. The problem of optimal stopping is concerned with choosing a time to take an action based on sequentially observed random variables in order to maximize an expected payoff; the classic secretary problem is the best-known example [AMW95]. In our case the expected payoff is the average quality of the nodes that eventually are holding the message.

Optimal stopping theory suggests that a simple strategy appropriate for this problem is to select the *maximum over the observations so far*. In fact, this approach has similarities to the hiring strategies studied in [BKK⁺08]. That paper considered a company interviewing candidates one by one and seeking to maximize average employee quality. A possible strategy would be to only hire candidates better than all current employees, called the **max** strategy. In fact, **max** is not a particular good strategy for hiring, since the rate of hiring decreases as time goes on, as shown in [BKK⁺08]. However, the forwarding problem is subtly different: when messages are forwarded, they are replicated and so become more numerous. This counteracts the slowdown in forwarding rate of any given message replica, and makes **max** surprisingly effective.

Delegation forwarding then consists of using the **max** strategy over quality to answer the forwarding question. A formal statement of delegation forwarding is given in Algorithm 1.

6.3 Analysis of Cost

In this section we analyze delegation forwarding and show that it reduces expected costs dramatically compared to the naive alternative.

6.3.1 Assumptions

As described in the previous section, we assume the existence of a quality metric with the property that nodes with higher quality are better candidates as intermediate carriers of a message than are nodes of lower quality. The quality metric can be destination-specific or destination-independent. A destination-specific quality metric is one that varies depending

on the destination of a message. For example, FRESH [DFGV03] uses the time elapsed since the last contact with the destination as a metric. Other quality metrics, such as the total contact rate of a node (described in Chapters 4, 5) is the same for all destinations and hence destination-independent. The results we present on the cost of delegation forwarding apply equally to both cases; hence for simplicity we drop the message subscript m from x_{im} and refer only to x_i . We study cost imbalance only for the destination-independent case. That case is intuitively the worst case scenario for imbalance, since in that case high quality nodes are the same regardless of destination. In Section 6.5 we will consider forwarding algorithms in simulation having destination-dependent quality metrics and we will see resulting improvements in cost imbalance.

Depending on the forwarding algorithm used, a node’s quality metric and contact rate may be dependent or may be independent. For example, choosing the total contact rate of a node as its quality metric has been discussed in Chapter 4. We show below that our results apply in the two extreme cases where quality and rate of nodes are independent, and when they are identical.

We assume that the node quality metric x_i follows a uniform distribution on the interval $(0, 1]$. On one hand, we note that if the quality metric is the node’s contact rate, it corresponds to the distribution observed empirically in some conference settings (see [ECCD07, CHC⁺05]). Note that a uniform distribution of rates is not inconsistent with a power-law distribution of inter-contact times, which have been observed in [CE07, CHC⁺05, KBV07]. On the other hand, we note that when quality is *independent* of contact rate, this assumption is not a restriction. In that case, the absolute value of the metric can be changed arbitrarily as long as the ordering between nodes is preserved. Any distribution can then be mapped to the uniform case.

We further assume that the node quality metric x_i is time-invariant. However we relax this assumption in our simulations, where some of the node quality metrics we use are continuously updated.

6.3.2 Cost

In the following, we consider a single message and study how many times it is forwarded before reaching a node with a high quality metric. This allows us to prove a bound on the number of copies created for each message. The initial part of our analysis makes use of the framework set out in [BKK⁺08].

Quality Independent of Contact Rate

For any node i maintaining a quality metric x_i (which lies between $(0; 1]$) and a threshold value τ_i , we focus on the gap $g_i = 1 - \tau_i$ between the current threshold and 1. The node that generates the message has threshold initially equal to its quality, *i.e.*, $\tau_i = x_i$. We denote the initial gap $g = 1 - x_i$.

Consider a node that updated its gap value n times. We denote the node's current gap as the random variable G_n . Since nodes meet according to rates that are independent of node quality, the node is equally likely to meet a node with any particular quality value. The next update of the gap of the nodes then occurs as soon as it meets a node with a quality greater than G_n , and all values above this threshold are equally likely.

Hence, we can write

$$G_{n+1} = G_n \cdot U, \quad (6.1)$$

where U is independent of G_n and follows a uniform distribution on $(0, 1]$. By induction we then find:

$$\mathbb{E}[G_{n+1} | G_n] = \frac{G_n}{2}, \quad \text{hence } \mathbb{E}[G_n] = \frac{g}{2^n}.$$

Moreover, from Eq.(6.1), we see that G_n approximately follows a lognormal distribution (see §2.2 in [BKK⁺08]), with median $\frac{g}{e^n}$. Hence the distribution is highly skewed with most of the probability mass below the mean, and so with large probability we have $G_n \leq \frac{g}{2^n}$.

By setting $\mathbb{E}[G_n] = \epsilon$, we find that the number of handoff stages it will take to get within ϵ of the highest contact rate node is

$$g/\epsilon = 2^n, \quad \text{so } n = \log_2(g/\epsilon). \quad (6.2)$$

If we want to get to the highest contact rate node, then $\epsilon = 1/N$.

Let us describe the replication process via a dynamic binary tree T , which contains all the nodes that have a copy of the message. Initially T contains a single node with associated gap g . Each time a node with a copy of the message meets another node having higher quality than any node seen so far, we create two children of the node. The children represent each of the two nodes, and both have associated the updated gap value. Note that different branches of this tree may grow more quickly than others. We wish to bound the total size of this tree.

We define the set $B = \{i \mid x_i \geq 1 - \frac{g}{\sqrt{N}}\}$, which we call the *target set*. We will also identify a subtree of the tree T in which children are excluded for nodes having a threshold above $1 - \frac{g}{\sqrt{N}}$. We call this subtree the *target-stopped tree*.

The essential observation is the following: if n is close to $\log_2(\sqrt{N})$, then except with a small probability, a node at generation n in the tree has a gap at most $g/2^n \leq g/\sqrt{N}$. This is because of the highly skewed nature of the distribution of G_n , as described above. Hence, we can safely assume that the target-stopped tree has depth at most n . Note that the total number nodes of appearing at generations $0, 1, \dots, n-1$ is at most $2^n = \sqrt{N}$.

We can now bound the size of the entire tree T , since all nodes at generations $n, n+1, \dots$ are included in the target set B . Hence, the total size of this tree is at most:

$$C_{\text{delegation}}(g) \lesssim \sqrt{N} + |B| = (1 + \sqrt{g}) \cdot \sqrt{N},$$

hence

$$\mathbb{E}[C_{\text{delegation}}] \lesssim \frac{5}{3} \sqrt{N}. \quad (6.3)$$

In contrast, the usual style of forwarding algorithm makes no threshold adaptation. A message starting at a node with gap g will eventually reach each of the nodes with higher quality, so that the cost

$$C_{\text{no-delegation}}(g) = gN, \quad \text{hence} \quad \mathbb{E}[C_{\text{no-delegation}}] = \frac{N}{2}. \quad (6.4)$$

Hence we see that delegation forwarding narrows the set of targeted nodes as additional message copies get created. This saves a significant fraction of the cost, while still causing the message to reach the most important nodes.

Quality Equal to Contact Rate

So far we have analyzed the case where node contact rates are independent of their quality metric. In reality, this may not always be the case since in some cases a good candidate for forwarding a message is a node that is met frequently by other nodes. To address this, we consider here an extreme case, namely, where quality and contact rates are identical. This corresponds to a forwarding strategy “forward to high contact rate nodes.” We will show that the delegation scheme can take advantage of this correlation, and that the resulting costs are as good or better than in the previous section.

We assume that the contacts between the nodes follow a *product form*: each node i has a total contact rate λ_i , and the rate of contact between nodes i and j is simply equal to the product $\lambda_i \lambda_j / \sum_j \lambda_j$. In other words, the rate of contact for a given pair of nodes depends on the nodes chosen via their total contact rates.

Since quality and contact rates are identical, $x_i = \lambda_i$ and both are distributed uniformly on the interval $(0, 1]$. As a result, we note that nodes with higher quality are met more often than nodes with lower quality. Hence, the quality of the next node met is not distributed uniformly. Instead we have:

$$\begin{aligned} \mathbb{P} [\text{next node met has quality} \in [x, x + dx]] &= \\ \mathbb{P} [\text{next node met has rate} \in [\lambda, \lambda + d\lambda]] &= \\ \frac{\lambda d\lambda}{\int_0^1 \lambda d\lambda} &= 2\lambda d\lambda = 2x dx. \end{aligned} \tag{6.5}$$

As before, instead of considering a node’s threshold, we consider the gap $g_i = 1 - \tau_i$. Looking at one node, we denote by G_n its gap after n updates. After conditioning on the current value of the gap G_n and substituting $1 - g$ for x , we have in expectation:

$$\begin{aligned} \mathbb{E} [G_{n+1} | G_n] &= \frac{\int_0^{G_n} g \cdot 2(1 - g) dg}{\int_0^{G_n} 2(1 - g) dg} \\ &= G_n \left(\frac{1 - \frac{2}{3}G_n}{2 - G_n} \right). \end{aligned} \tag{6.6}$$

Note that the function $h : x \rightarrow \frac{1 - (2/3)x}{2 - x}$ is strictly decreasing, approaching $\frac{1}{2}$ when $x \rightarrow 0$

and $\frac{1}{3}$ when $x \rightarrow 1$. Hence we have,

$$\frac{G_{n-1}}{3} < \mathbb{E}[G_n | G_{n-1}] < \frac{G_{n-1}}{2},$$

and so by induction,

$$\frac{g}{3^n} < \mathbb{E}[G_n] < \frac{g}{2^n}.$$

So we obtain a gap on average within ϵ of the minimum value after n handoffs, with n in the range

$$\log_3(g/\epsilon) < n < \log_2(g/\epsilon).$$

This shows the number of handoff stages needed to get within ϵ of the highest quality node is less than in the independent case. Hence the independent case expression $C_{\text{delegation}}(g) \lesssim (1 + \sqrt{g})\sqrt{N}$ is an upper bound for the delegation cost in the case where quality is equal to contact rate, and so the improvement in cost when using delegation forwarding is even greater in this case.

Numerical Results

To confirm our analytical results, we simulated a collection of nodes interacting randomly with a mixture of contact rates, and generating messages uniformly at random with respect to source and destination, where the quality metric used was total contact rate. In Fig. 6.1 we compare the measured cost (message copies per message created) in simulation to that predicted by Equations (6.3) and (6.4). The plot shows that in practice the costs of delegation forwarding as a function of number of nodes is close to predictions, and it confirms the dramatic improvement in cost when using delegation forwarding.

6.4 Cost Imbalance

While the last section showed that the overall cost in terms of message replicas is dramatically reduced under delegation forwarding, it is also important to ask whether the costs are fairly (i.e., equally) shared among the nodes.

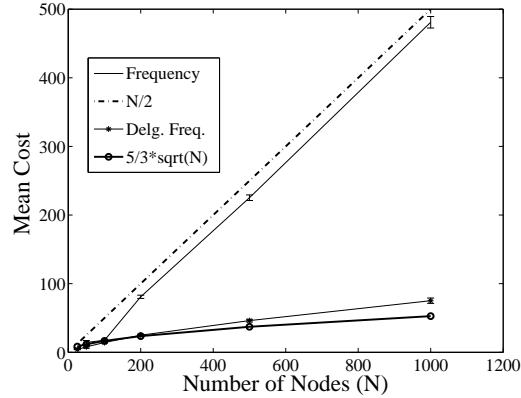


Figure 6.1: Cost as function of number of nodes.

To answer this question, we proceed in stages. We return to making the assumption that a node's quality and contact rate are independent. Then, the first question we ask is as follows. Given a node \mathcal{N} with quality a that is holding a message. What is the probability that this node will at some point forward this packet to a node \mathcal{N}' having quality x ?

Denote this probability density $p(x|a)$. Note that $p(x|a)$ is not a distribution over x because \mathcal{N} may forward its message more than once. However the probability of forwarding to any single node is not greater than 1. That is, for any integer $u < N$, $1/N \cdot p(u/N|0) \leq 1$.

Theorem 6.4.1. *Given a node \mathcal{N} with quality a that is holding a message. The probability that this node will at some point forward this packet to a node \mathcal{N}' having quality $x > a$ is proportional to $\frac{1-a}{1-x}$, for $x \leq 1 - 1/N$.*

Proof. First consider the case where $a = 0$. Denote the quality of the node forwarded to on forward number i , and not before as X_i . The probability distribution of X_i is $p_{X_i}(\cdot)$. Then $p(x|0)$ is equal to:

$$p(x|0) = \sum_{i=1}^{\infty} p_{X_i}(x) \quad (6.7)$$

These events are mutually exclusive so the sum is valid. Now the distribution of X_1 is the uniform distribution on $(0,1]$. We can obtain X_i for successive i s by the law of total probability:

$$p_{X_i}(x) = \int_0^1 p_{X_i}(x|X_{i-1} = u) p_{X_{i-1}}(u) du$$

$$\begin{aligned}
p_{X_2}(x) &= \int_0^1 p_{X_2}(x|X_1 = x_1)p_{X_1}(x_1) dx_1 \\
&= \int_0^x 1/(1-x_1) dx_1 \\
&= -\ln(1-x)
\end{aligned}$$

In the same manner we find that $p_{X_3}(x) = \frac{1}{2} \ln^2(1-x)$ and $p_{X_4}(x) = -\frac{1}{6} \ln^3(1-x)$. So we can see that $p_{X_n}(x) = \pm \frac{1}{n-1!} \ln^{n-1}(1-x)$ with alternating signs. Returning to Equation (6.7), we can now write:

$$\begin{aligned}
p(x|0) &= \sum_{i=1}^{\infty} p_{X_i}(x) \\
&= 1 - \ln(1-x) + \frac{1}{2} \ln^2(1-x) \\
&\quad - \frac{1}{6} \ln^3(1-x) + \dots
\end{aligned}$$

To evaluate this sum, consider it a function of x . That is, $f(x) = p(x|0)$. Then we note from differentiating the infinite sum above that $f'(x) = \frac{1}{1-x} f(x)$. Thus

$$f(x) = \frac{k}{1-x} \text{ for some } k > 0$$

since then $f'(x) = k/(1-x)^2 = 1/(1-x) f(x)$. Since $f(0) = 1$, we find that $k = 1$.

For the case when $a > 0$, we reason that the same relationship should apply, with the range $(a, 1]$ mapped to the range $(0, 1]$. Then,

$$p(x|a) = \frac{1}{1 - \frac{x-a}{1-a}} = \frac{1-a}{1-x}$$

Where the expression $\frac{x-a}{1-a}$ maps an x in the range $(a, 1]$ to the range $(0, 1]$. □

This agrees with intuition: if a node with quality zero is holding the message, the probability that a node of quality $1/N$ will receive the message is $1/N$; a node of quality $1/2$ will receive the message with probability $2/N$; and a node of quality $1 - 1/N$ will receive the message with probability 1.

Now we can begin to answer the question of cost imbalance. Given a node \mathcal{N} having quality x , what is the expected number of messages M that will be sent to it?

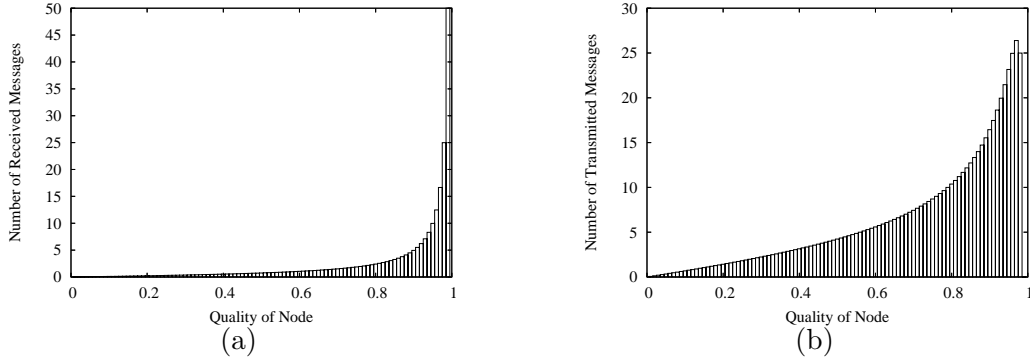


Figure 6.2: Cost Imbalance, 100 nodes. (a) Node Memory Load and (b) Node Transmission Load; as a function of quality.

Denote the probability that a node with quality x generates a message as $p_m(x)$. Assume that messages are generated uniformly, so $p_m(x) = 1$. Then this is

$$\begin{aligned}
 E[M|x] &= \int_0^x p(x|a) p_m(a) da \\
 &= \int_0^x \frac{1-a}{1-x} da \\
 &= \frac{2x - x^2}{2 - 2x}
 \end{aligned} \tag{6.8}$$

For example, assume each node generates one message and there are 100 nodes. Then the graph in Figure 6.2(a), which is a plot of Equation (6.8) at intervals of $1/100$, shows how many messages each node should be expected to receive. For example, node 99 will receive 50 messages.

Further, we can also address the question of forwarding cost. Given the same assumptions as above, how many messages should a node expect to have to forward, i.e., what is $E[F|x]$?

We reason, as in the analysis of overall forwarding cost in Section 6.3.2, that a node having quality x and holding a message needs to forward it approximately $k(x)$ times, where $k(x)$ is given by:

$$2^{k(x)} = \frac{(1-x)N}{2^{k(x)}}.$$

As before, the reasoning behind this is that it is approximately the case that at some point after n forwards, 2^n nodes will be holding the message, and there will be $gN/2^n$ nodes of

higher quality not holding the packet. Here g is the initial gap, i.e., $1 - x$. Then there will be $n + 1$ forwards altogether.

This means $k(x)$ is $n + 1 = \log_2((1 - x)N)$. So to answer our question,

$$E[F|x] = E[M|x]k(x).$$

Continuing the same example, the expected number of forwarded messages is shown in Figure 6.2(b). The median value is 4.2 and the maximum is 25.

Thus we conclude that both node memory load and node transmission load are unevenly distributed. Node memory load is more highly skewed than node transmission load because as a node's quality increases, the number of nodes it must send to diminishes. In a 100 node system the busiest node has about five times the transmission load of a typical node, and the top 5% have at least four times the transmission load of a typical node. Overall, this level of imbalance is undesirable but not prohibitive.

6.5 Evaluation on Real Traces

In the previous section we analyzed delegation forwarding and showed that it can dramatically reduce costs. In this section we show that, given realistic contact patterns, delegation forwarding can yield performance comparable to non-delegation approaches, and comparable to the best known forwarding algorithms. Further, we assess the degree of cost imbalance under delegation forwarding and show that in realistic contact patterns, certain delegation forwarding algorithms show cost imbalance no worse than the best alternatives.

We use the datasets described in Chapter 3, namely the four conference datasets and the four university based datasets.

6.5.1 Experiments

We implemented a variety of forwarding algorithms in a trace-driven simulator described in detail in Chapter 5. Our metrics are success rate, average delay, and cost (as defined in Chapter 4).

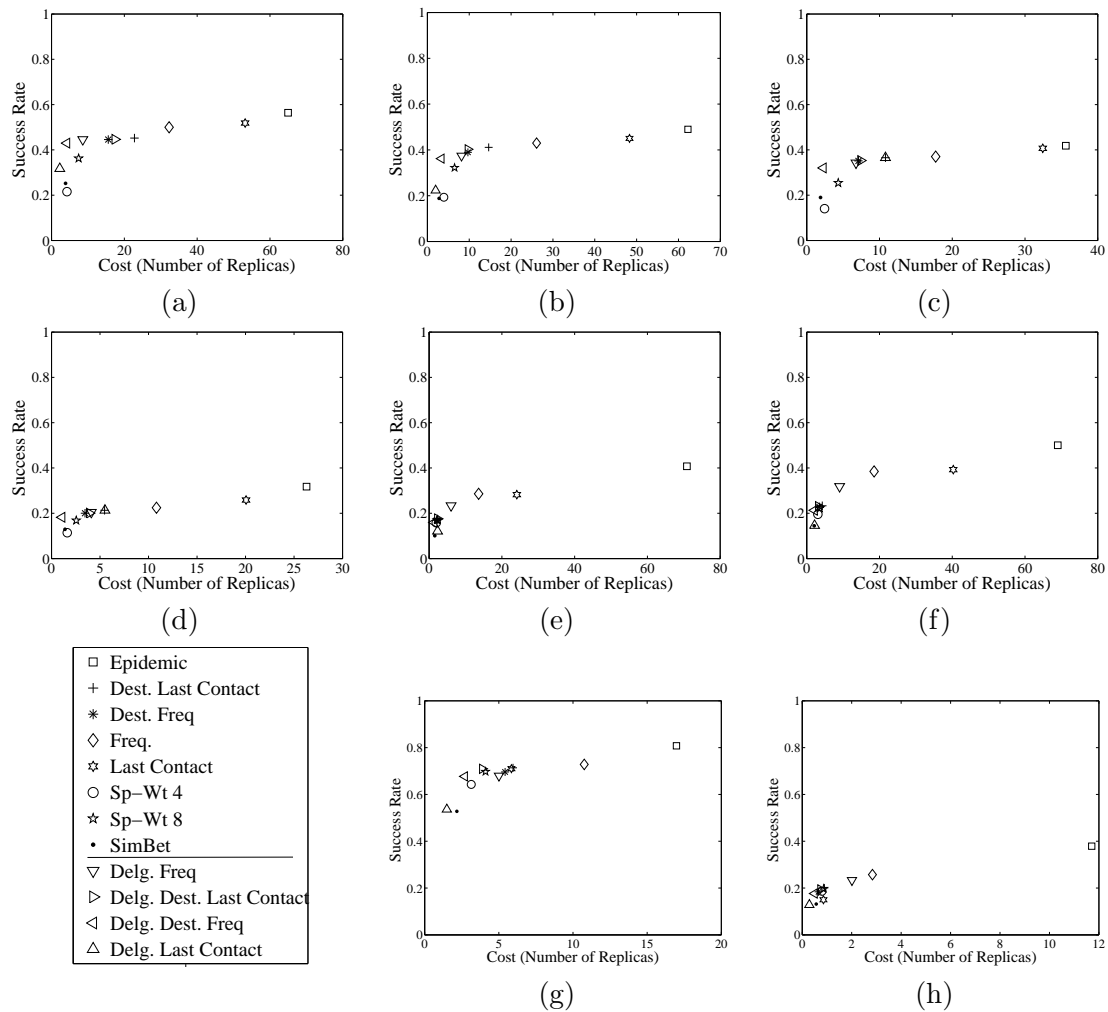


Figure 6.3: Performance of Forwarding Algorithms. Top Row: Infocom 06 9-12; Infocom 06 3-6; Conext 06 9-12. Middle Row: Conext 06 3-6; UCSD 9-3; UCSD 3-9. Bottom Row: RealityGSM; RealityBT. For each dataset, plot is Cost versus Success Rate

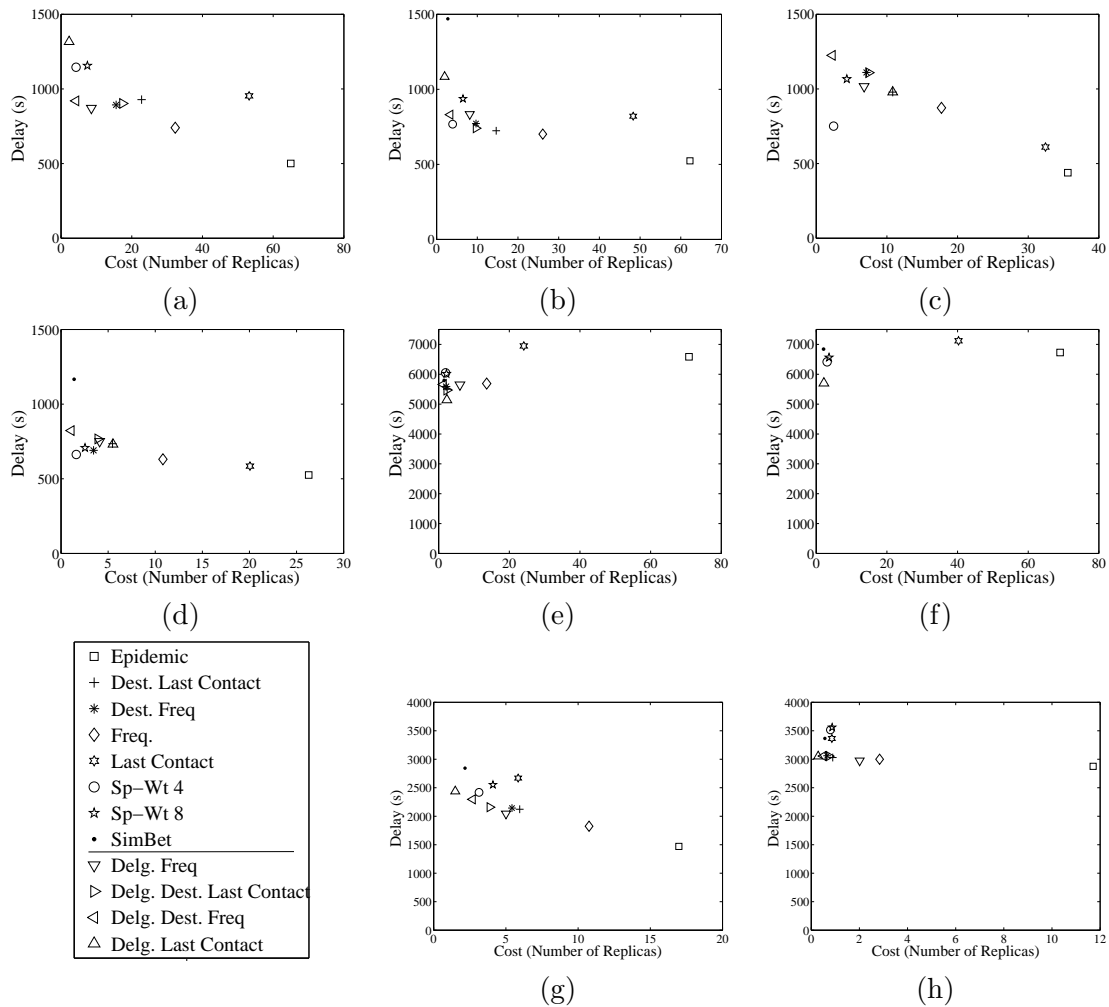


Figure 6.4: Performance of Forwarding Algorithms. Top Row: Infocom 06 9-12; Infocom 06 3-6; Conext 06 9-12. Middle Row: Conext 06 3-6; UCSD 9-3; UCSD 3-9. Bottom Row: RealityGSM; RealityBT. For each dataset, plot is Cost versus Delay

We selected forwarding algorithms so as to include both well-known existing algorithms as well as algorithms that span a wide range of design choices. All these algorithms are distributed and operate in an online manner working with local information. For each algorithm we describe the rule used to decide whether to forward a message \mathcal{M}_m held by \mathcal{N}_i when node \mathcal{N}_i meets node \mathcal{N}_j . Most of the algorithms have been described in Chapter 5. We describe the algorithms here again, for convenience. In addition, we also describe some other algorithms which we implement

Epidemic (Flooding) [VB00]: Node \mathcal{N}_i forwards \mathcal{M}_m to \mathcal{N}_j unless \mathcal{N}_j already has a replica of \mathcal{M}_m . Epidemic forwarding achieves the best possible performance, so this algorithm yields upper bounds on success rate and average delay. However it is also the case that epidemic forwarding will have the highest costs.

Frequency [ECCD07]: Node \mathcal{N}_i forwards \mathcal{M}_m to node \mathcal{N}_j if \mathcal{N}_j has more total contacts (with all other nodes) than does \mathcal{N}_i . This algorithm is destination independent.

Last Contact: Node \mathcal{N}_i forwards \mathcal{M}_m to \mathcal{N}_j if \mathcal{N}_j has contacted any node more recently than has \mathcal{N}_i . This algorithm too is destination independent.

Destination Frequency: Node \mathcal{N}_i forwards \mathcal{M}_m to \mathcal{N}_j if \mathcal{N}_j has contacted \mathcal{M}_m 's destination more often than has \mathcal{N}_i .

Destination Last Contact [DFGV03]: Node \mathcal{N}_i forwards \mathcal{M}_m to \mathcal{N}_j if \mathcal{N}_j has contacted \mathcal{M}_m 's destination more recently than has \mathcal{N}_i . This algorithm is also known as FRESH [DFGV03].

Spray and Wait (SpWt) [SPR08]: \mathcal{M}_m 's source initially creates l replicas of \mathcal{M}_m . If node \mathcal{N}_i has $k > 1$ replicas of \mathcal{M}_m and \mathcal{N}_j has no replicas, \mathcal{N}_i will forward half its replicas to \mathcal{N}_j and keep the other half. If node \mathcal{N}_i has just one replica of \mathcal{M}_m , it uses the destination last contact rule (described above). We use two variants, having $l = 4$ and 8.

SimBet [DH07]: Node \mathcal{N}_i forwards \mathcal{M}_m to node \mathcal{N}_j if \mathcal{N}_j scores higher on the simbet metric. To compute the simbet metric, one views the underlying contact graph as a social encounter graph, and incorporates two social measures (similarity and betweenness) of a node. Only one replica of the message exists in the network. We use the same parameters

used in [DH07].

The delegation schemes we consider are: **Delegation Destination Frequency**, **Delegation Destination Last Contact**, **Delegation Frequency**, and **Delegation Last Contact**, each of which is obtained by applying the delegation forwarding strategy to the corresponding algorithm above.

6.5.2 Results

Cost The results of our simulations are shown in Fig. 6.3 and Fig. 6.4. Each plot in Fig. 6.3 shows success rate versus cost and mean delay versus cost in Fig. 6.4. The eight pairs of plots correspond to the eight traces we analyzed, as listed in the caption to the figure. In these plots, the four triangles (at different orientations) correspond to the four delegation forwarding algorithms we studied.

In the success rate versus cost plots, the best algorithms are those closest to the upper left corner; in the delay versus cost plots the best are in the lower left corner. Our first observation is that usually, one of the delegation algorithms occupies the best position in these plots. This means two things: first, as expected, delegation forwarding has very low costs – usually the lowest of any algorithm. Second, and more surprisingly, delegation forwarding usually performs about as well as most other forwarding algorithms. Indeed, delegation schemes reduce cost drastically, by as much as 3/4 of the original cost, while maintaining approximately the same success rate with a modest increase in average delay. Taken as a whole across all these plots, delegation forwarding is clearly the best choice for trading off performance and cost.

We note that in terms of success rate, it is often the case that most forwarding algorithms have performance within a narrow range. This is also true for mean delay, although the range of variation can be larger. These results suggest that forwarding algorithms differ mostly in the costs incurred, rather than performance, which is consistent with previous results in Chapter 5 and argues for favoring delegation forwarding.

Looking more closely at the variants of delegation forwarding, we see that delegation

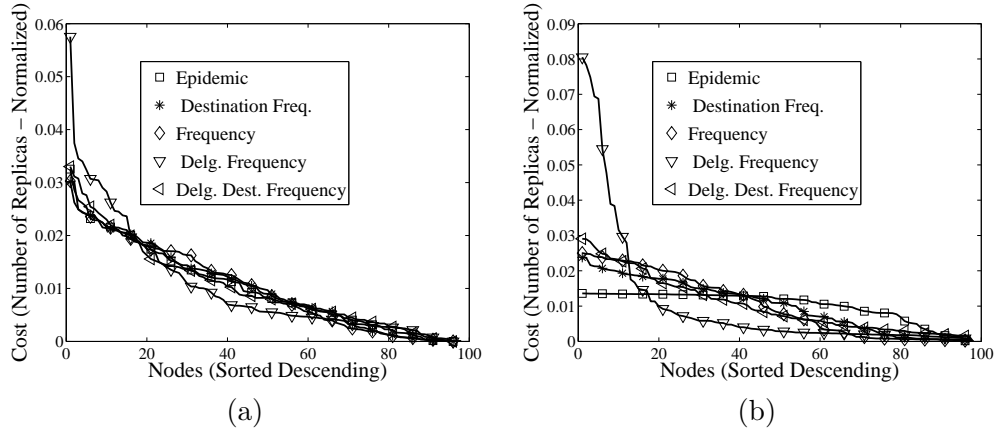


Figure 6.5: Plots (Infocom 9-12) (a) Number of Transmissions (b) Queue Lengths destination frequency is most often the best of the delegation approaches. This makes sense because, being destination-specific, the algorithm has more information to work with, and hence becomes more selective in forwarding.

Cost Imbalance In Section 6.3 we showed analytically that node memory and node transmission loads can be unevenly distributed across nodes when the quality metric is independent of destination. It is important to ask how serious this problem is in realistic settings, and also whether destination-dependent quality metrics show different behavior.

To answer these questions we plot the node transmission load and node memory load (normalized) for three non-delegation approaches and two variants of delegation forwarding. The results are shown in Fig. 6.5. Fig. 6.5(a) shows that all algorithms suffer approximately equally from imbalance in node transmission load, except delegation frequency which has noticeably higher imbalance. The imbalance seen for delegation frequency (where quality of node is equal to its contact rate) is in line with analytical results (Section 6.3), e.g., the top 5% of the nodes transmit almost 5 times more than a typical node. Delegation destination frequency however shows no more imbalance than any of the non-delegation schemes. This occurs because under the destination frequency approach, the highest quality nodes for different destinations tend to be different. (We have verified this via additional analysis, but space does not permit their inclusion.)

Fig. 6.5(b) shows imbalance in node memory load. As expected from analysis, node

memory load is more skewed for delegation frequency than was node transmission load. For example, the top 10% of the nodes have around 8 times the load of a typical node, consistent with analysis. Once again however, delegation destination forwarding shows no greater imbalance than the non-destination approaches.

6.6 Summary

Forwarding in mobile opportunistic networks is a challenging problem. An objective of many forwarding algorithms is to reduce cost while trying to keep success rate high and delays low. These algorithms often rely on a quality metric associated with nodes in the network to make forwarding decisions. The main contribution of this chapter is to propose a new forwarding strategy explicitly designed to reduce costs while maintaining the high performance of such algorithms.

We start with the observation that in order to reduce costs, a good strategy could be to forward *only* to the *highest*-quality nodes. However the distribution of node quality is not known, so every node has to decide if a current encounter is with a high-quality node or not by relying on past contacts. This is an instance of an optimal stopping problem, and seen in that light, a good strategy is for a node to only forward a message to nodes with quality greater than any seen so far for this message. This is delegation forwarding.

We analyze two variants of delegation forwarding, considering both the cases where quality is independent of the underlying contact rate and also where quality is identical to a node's contact rate. In both these cases we show that delegation forwarding reduces expected costs dramatically (from $O(N)$ to $O(\sqrt{N})$) while still ensuring that messages reach the highest quality nodes. We also study the fairness of delegation forwarding in terms of the distribution of per-node cost. We show that when node quality is independent of message destination, delegation forwarding can induce a moderate level of imbalance in per-node cost.

We then turn to studying the performance of delegation forwarding on real mobility

traces and observe that overall, delegation forwarding approaches are preferable to any of a set of commonly-studied forwarding algorithms. Delegation approaches generally achieve comparable performance to well-known alternatives while also generally achieving remarkably low costs. Furthermore, we show that when evaluated in realistic settings, delegation forwarding using destination contact frequency as its metric shows no greater per-node imbalance than non-delegation alternatives, suggesting that the worst-case results from analysis need not be experienced in practice.

All our studies on forwarding algorithms so far assume the presence of infinite buffers at nodes, as well as infinite bandwidth. Clearly these assumptions do not hold in practical situations. We therefore focus our attention on the same scenarios and consider finite buffers and bandwidth in the next chapter.

Chapter 7

Resource Constrained Scenarios

The previous chapters in this thesis, and indeed previous work [ECCD07, DFGV03, SPR08, DH07, ECCD08] have focussed on addressing the challenge of handling unpredictable mobility of underlying nodes. However, there has been little work done to address the challenge of forwarding under resource constraints, that include limited battery life, short contact durations and small buffers.

For a mobile network operating under such constraints, the joint question of which messages to transmit and which messages to drop becomes important. As with a forwarding algorithm, every node should be able to decide which messages are transmitted and which messages are dropped based on the information the node has, all the while balancing the trade-offs that exist between success rate, delay and cost. Hence there is a need to develop and study prioritization schemes for messages such that nodes can forward high priority messages and drop low priority messages.

7.1 Overview

Message prioritization can be performed independent of the underlying forwarding algorithm. Such schemes include FIFO [VB00], LIFO and TTL-based algorithms [SPR08]. While such schemes are easy to implement, the fact that they do not take into account network information can lead to poor performance and suboptimal use of the scarce resources.

For the schemes that do use network information to make informed decisions there is a crucial question: on the one hand, one can assign high priorities to messages that are close to their intended destination. A node following such a scheme, will upon an encounter, drop the message farthest from its destination and transmit the message closest to its destination. Proposed schemes which share this philosophy include PREP [RHB⁺07] and DLE [DFL01].

On the other hand, one can decide to assign high priorities to messages which are farthest away from their destination [BGJL06]. Following such a scheme, the first message that will get dropped will be the message that is closest to its destination, and the first message which will get transmitted (after delivering the those messages destined to the encountered node) will be the message farthest from the destination. Clearly the notion of network distance to a destination is extremely important in such schemes that use network information.

The work presented in this chapter starts with the observation that in order to develop an effective prioritization scheme, it is useful to study dropping schemes and transmission schemes independently of each other. In order to facilitate our study, we develop and study a message prioritization scheme based on delegation forwarding algorithms developed in Chapter 6.

In the previous chapter we have shown that delegation schemes reduce costs by relying on principles of optimal stopping theory to decide when to forward messages. These schemes rely on a forwarding metric assigned to every node, that is generically referred to as ‘quality’ for that node. This metric determines the probability of successful delivery of messages to their intended destination, and hence convey the notion of ‘distance’ to a destination. For the purposes of this study, the choice of delegation forwarding is not essential. However the combination of low cost and the use of a generic metric to convey distance make delegation schemes appealing for basing our study. We augment the delegation algorithms with a replication-count number, which we refer to as *delegation number*. This number is then used to assign priorities to messages.

In this chapter we study a variety of schemes using real encounter traces. The most significant result presented in this chapter is the scheme which assigns high priorities to

messages with low delegation number performs the best in terms of success rate, delay and cost. We interpret this result in light of the path explosion phenomenon observed before and reported in Chapter 4.

7.2 Assigning Priorities to Messages

In this section we first describe the setting we are working with. We then discuss the schemes which we study as well as the potential implications of using these schemes. We then describe the a prioritization scheme which is based on delegation forwarding.

7.2.1 Formulation

Building on the formulation given in Chapter 4 we assume a set of mobile nodes $\mathcal{N}_i \in \mathbb{M}$ with $|\mathbb{M}| = N$. Nodes generate messages over time; each message has a particular source and destination. We define M_i as the set of messages held by a given node \mathcal{N}_i . When nodes come into contact; have encounters, they are capable of exchanging messages. Messages are transmitted in whole from node to node during node contact intervals, after which both nodes hold message *replicas*. Nodes hold on to the message replicas until their respective buffers are full, at which time the nodes have to decide which message to drop. Nodes do not possess any *a priori* knowledge of the number of nodes in the system or knowledge of any properties of the other nodes. We are working in a setting where all nodes are mobile; we do not consider settings where some nodes may be stationary and are gateway nodes to the Internet.

We consider the following metrics, defined in Chapter 4: (1) *cost*, which is the number of replicas per generated message in the network; (2) *success rate*, which is the fraction of generated messages for which at least one replica is eventually delivered; and (3) *average delay*, which is the average duration between a message’s generation and the first arrival of one of its replicas at the destination. By “high performance” we mean high success rate and low average delay.

When a node \mathcal{N}_i encounters another node \mathcal{N}_j , the basic question that a forwarding algorithm addresses is which subset of messages m_i in the set M_i should node \mathcal{N}_i forward to node \mathcal{N}_j to maximize performance and minimize cost. Many of the proposed solutions differ in the type of network information used, as well as how they use it to answer the above question (refer to Chapter 6 for a detailed discussion on this point). However some of these schemes are designed and evaluated under the assumptions of infinite buffer capacity and infinite bandwidth [ECCD08, SPR08, LFC05, JFP04]. With no resource constraints there is no need for prioritization of messages.

However under the presence of finite contact durations and finite buffers, the questions of prioritizing the transmission and dropping of messages becomes equally important. The question then becomes which messages in the subset m_i should node \mathcal{N}_i transmit first and which messages in the subset should node \mathcal{N}_j drop first to minimize cost and maximize performance?

We start with the observation that it is important to study message dropping schemes independently of message transmission schemes. This will let us identify a consistent prioritization scheme if it exists. From a node's perspective, message prioritization schemes can either be independent of the forwarding algorithm or use the information provided by the forwarding algorithm to decide on the priorities.

Examples of the former include **FIFO**, **LIFO** and **Random**. These schemes do not use any extra information of the network - a node merely inspects its buffer and drop/transmits a message in FIFO/LIFO manner. Else the node will pick a random message to drop or transmit. We consider these schemes in our study. More details are provided in the next section.

When message prioritization schemes incorporate network information, one can think of two diametrically opposite schemes. On one hand we have the following scheme: upon an encounter, assign high priority to messages that are closest to their destination, and low priority to messages that are further away from their respective destination. The rationale for such a scheme is fairly intuitive — the network expends some energy in replicating and

forwarding a message close to its destination. Given a choice between spending energy on a message that will get closer still to its destination, or a message which is not certain to reach its destination, choose the former. Example schemes include PREP [RHB⁺07] and DLE [DFL01].

The main problem with such schemes is that some messages might suffer from ‘under-replication’; a node might only replicate and forward messages which are close to a destination, never giving a chance to relatively new messages to propagate in the network (Although PREP [RHB⁺07] does use a hop-count value to ensure under-replication does not occur).

On the other hand a node can adopt the following scheme: assign high priority to messages which are farthest from their destination and low priority to messages which are closest to their respective destinations. This scheme is less-intuitive — given that we rely on multi-hop forwarding, favoring a message that is far from its intended destination over a message close to its destination would entail spending more resources which are scarce in the first place. Besides, dropping a message which is close to its destination, with the hope that another replica of the message makes it to the destination may increase delays for that message. However a potential benefit of such a scheme is that it may address the ‘under-replication’ problem. As a message makes its way thru the network to get closer to its destination, multiple replicas of the message might propagate, increasing the probability of successful delivery.

We explore these tradeoffs using empirical measurements in the next section. However before we do so, we need to address the question of which metrics should we use that give a good indication of distance to the destination.

We use the delegation forwarding scheme described in Chapter6 which have been to shown to be efficient in terms of cost and deliver high performance.

Algorithm 2 Augmented Delegation Forwarding

```

Let  $\mathcal{N}_1, \dots, \mathcal{N}_N$  be nodes
Let  $\mathcal{M}_1, \dots, \mathcal{M}_M$  be messages
Node  $\mathcal{N}_i$  has quality  $x_{im}$  and threshold  $\tau_{im}$  for  $\mathcal{M}_m$ .
INITIALIZE  $\forall i, m : \tau_{im} \leftarrow x_{im}; d_m \leftarrow 1;$ 
On contact between  $\mathcal{N}_i$  and node  $\mathcal{N}_j$ :
for  $m$  in  $1, \dots, M$  do
  if  $\mathcal{M}_m$  is currently held by  $\mathcal{N}_i$  then
    if  $\tau_{im} < x_{jm}$  then
       $\tau_{im} \leftarrow x_{jm}$ 
      if  $\mathcal{N}_j$  does not have  $\mathcal{M}_m$  then
        increment  $d_m$ 
        forward  $\mathcal{M}_m$  from  $\mathcal{N}_i$  to  $\mathcal{N}_j$ 
      end if
    end if
  end if
end for

```

7.3 Delegation Number based Priorities

Recall that delegation forwarding forwards a message only if the encountered node has *quality higher than all seen so far by the message*. We now augment the delegation scheme by including a simple replication-count. We refer to this number as *delegation number* (d_m) of the message. The delegation number is incremented in the message, it is then replicated and forwarded.

A formal statement of the augmented delegation forwarding is given in Algorithm 2. Consider a node holding a message with a high delegation number. This would mean :

- (i) this node is one of the best candidate nodes to forward the message to its intended destination, and hence is close to the destination, else it would not be holding the message;
- (ii) even if the node itself is not a good candidate to carry the message to the destination, a high number signifies that the message has been passed to candidates that *are* good candidate nodes to carry the message. Likewise consider a message with a low delegation number: the message has not been replicated enough because the message is relatively young.

An example is given in Fig. 7.1. In this example, consider node A which has two messages

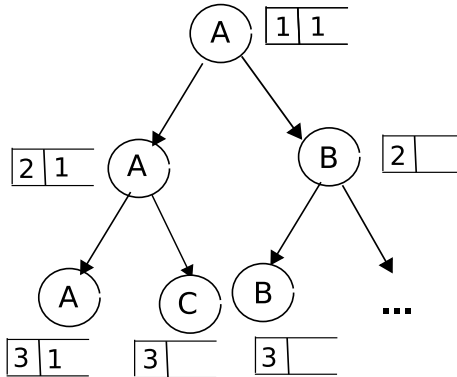


Figure 7.1: **Delegation Tree** for node A

in its buffer $(\mathcal{M}_1, \mathcal{M}_2)$, both with $d_1, d_2 = 1$. Node A encounters node B which turns out to be a better candidate for message \mathcal{M}_1 . d_1 is increased, the message is then replicated and forwarded to node B. Node A has another encounter say, with node C and forwards \mathcal{M}_1 again. d_1 is therefore 3, while d_2 , which has not been forwarded, remains 1. Clearly, the high delegation number for \mathcal{M}_1 in node A tells us that \mathcal{M}_1 has been forwarded to nodes which are good or better candidates to carry the message. Likewise the high delegation number for \mathcal{M}_1 in node C tells us that node C belongs to the set of good candidate nodes to carry the message. The low delegation number on \mathcal{M}_2 signifies that it has not been replicated enough, because it is relatively new in the network.

We formulate the two prioritization schemes and study them empirically using real encounter traces. The details are given in the next section.

7.4 Evaluation

In this section we evaluate various message drop and transmission schemes using real encounter traces described in Chapter 3.

7.4.1 Experiments

We again rely on the Infocom and Conext conference traces described in detail in Chapter 3 to evaluate various schemes.

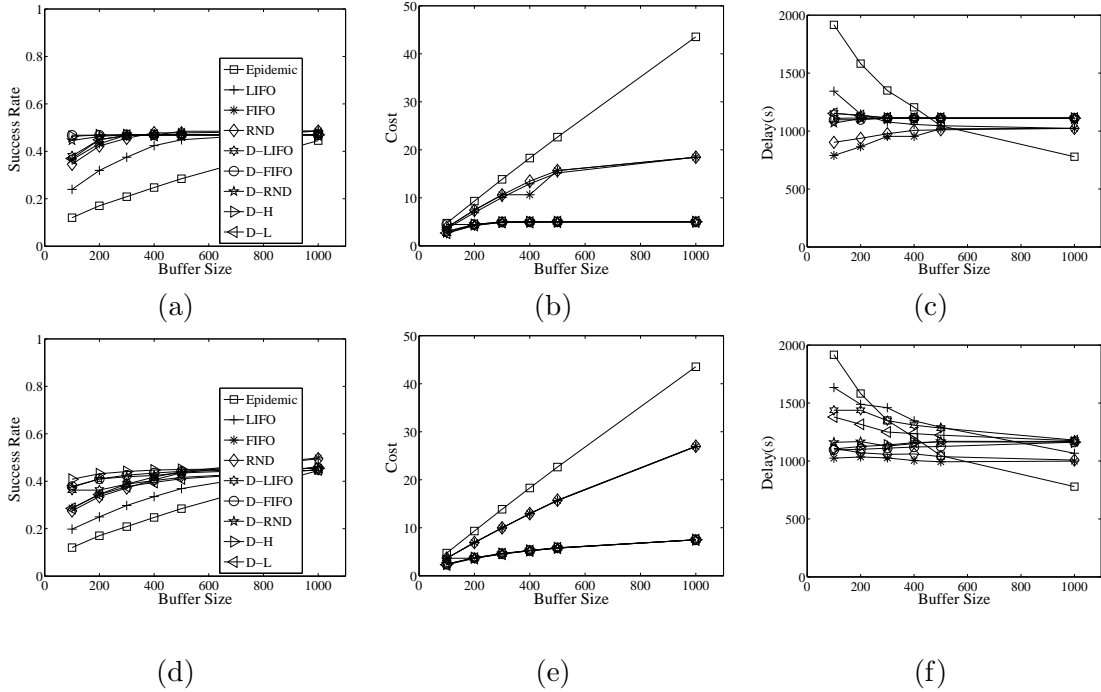


Figure 7.2: Message Dropping Schemes Info 06 9-12 (a,b,c) Destination Frequency (d,e,f) Frequency; (a,d) Success Rate, (b,e) Cost and (c,f) Delay

As mentioned earlier, we study message dropping and message transmission schemes independently of each other. In order to capture a diverse set of conditions, we choose a stateless algorithm epidemic [VB00], a destination agnostic algorithm (“frequency” Chapter 4) and a destination based algorithm (“destination frequency” Chapter 4) as the underlying forwarding algorithms. While epidemic algorithm relies on always forwarding messages on every contact, frequency relies on using contact rates of nodes to make forwarding decisions. Destination frequency uses contact rate with the destination to make decisions. Therefore, the frequency algorithm will operate as follows. A node upon an encounter with another node, forwards messages only if the the encountered node has a higher contact rate than itself. Destination frequency operates similarly but the information used is contact rate with the destination instead.

The latter two have been shown to give good performance while reducing cost in the network. Delegation schemes can be applied to any generic metric which gives the likelihood

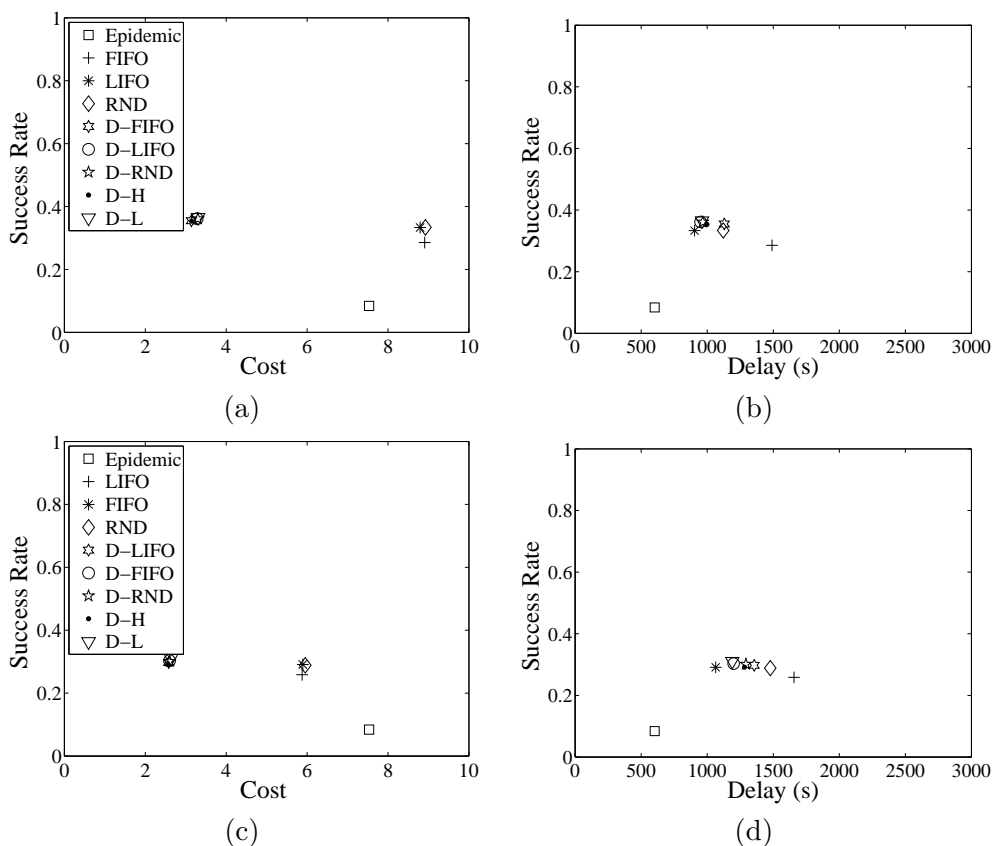


Figure 7.3: Message Transmission Schemes Info 06 9-12 (a,b) Destination Frequency (c,d) Frequency; (a,c) Success Rate vs Cost, (b,d) Success Rate vs Delay

of successful delivery of a message to its intended destination. We therefore also consider delegation versions of frequency and destination frequency for comparison. The delegation version of frequency would then be: a node, upon an encounter, forwards messages if the encountered node has contact rate higher than the rate of all nodes seen so far. A similar scheme can be designed with destination frequency. Refer to Chapter 6 for more details.

We implemented a variety of forwarding algorithms in a trace-driven simulator described in Chapter 5. The metrics are success rate, average delay, and cost (as defined in Chapter 5). Each simulation run was therefore 3 hours long; to avoid end-effects, no messages were generated in the last hour of each trace.

7.4.2 Message Dropping Policies

We limit and vary the size of buffers per node and choose to study the following set of policies which span a spectrum of possible design choices. Ties are broken randomly. We make the assumption that all messages are inserted in arrival order in a buffer. Given this, we have:

FIFO: Upon a buffer overflow condition, node \mathcal{N}_i drops the oldest message in the buffer.

LIFO: Upon a buffer overflow condition, drop the newest message in the buffer.

Random: Upon a buffer overflow condition, drop a random message from the buffer.

This scheme is between the FIFO and LIFO schemes mentioned above.

As mentioned in Section 7.2, these schemes are independent of the forwarding algorithm. We consider these policies in conjunction with our algorithms defined earlier (Frequency, Destination Frequency and their delegation versions). In addition, we also focus on the two schemes we developed which exploit the information provided to them by the forwarding algorithm.

High (H): Upon a buffer overflow, drop the message which has the highest delegation number $d_i = \max_{\forall j}(d_j)$. This is analogous to dropping the message which is closest to the destination as well the message which has been forwarded many times.

Low (L) Upon a buffer overflow, drop the message which has the smallest delegation number $d_i = \min_{\forall j}(d_j)$. This is analogous to dropping the message which is relatively young in the network or the message which is farthest from the destination.

We vary the buffer size (100, 200, 300,400, 500 and 1000) messages on every node. We study drop policies independently of bandwidth constraints, hence we assume unbounded contact duration in these experiments.

The results are shown in Fig. 7.2. The plots show success rate, cost and delay versus increasing buffer size. The top plot shows results for destination frequency and the bottom set of plots correspond to frequency. Figs.7.2 (a,d) show success rate versus buffer size. We first note that the all schemes perform similarly at high buffer sizes. Presumably, epidemic forwarding which does not do well under the presence of small buffers (as has been reported

before [RHB⁺07]), will perform the best with higher buffer sizes. All delegation schemes have a higher success rate than the non-delegation versions. This is because delegation schemes have been shown to reduce overall cost (Chapter 6) in the network, so they utilize the scarce resources more efficiently. Figs. 7.2 (b,e) show cost versus increasing buffer sizes. Clearly the cost will increase as buffer sizes increase, but for all algorithms (excluding once again, epidemic), large buffers do not help in increasing the success rate. All the delegation schemes achieve a very low cost; similar results have been reported before Chapter 6. Figs. 7.2 (c,f) show average delays as a function of message buffer sizes. The average delays decrease for epidemic as buffer sizes increase. However for the other schemes, the increase in buffer sizes do not have a large effect on delay. Across different schemes, however we note that while LIFO does the best, delegation schemes have higher delays - though they are not prohibitively high. The surprising result is that delegation-H policy(drop the message with the highest delegation number) gives the best success rate(with low cost) amongst all policies; for low buffer sizes it is an improvement by a factor of almost 2 over LIFO. The delegation-H policy is opposite of the policy employed by PREP and DLE in that we drop the message which is closest to its respective destination.

As discussed before, dropping a message close to its destination might increase delays. However we see from the results that the delays do increase, but they are not prohibitive. Overall it would appear that delegation schemes (H,RND) perform well under small buffer sizes, and the policy of dropping the message which is closest to its intended destination offers the best trade-off between performance and cost.

7.4.3 Message Transmission Policies

In order to impose constraints on message transmission, we limit the number of messages which a node can transmit during an encounter to at-most 1. We study transmission policies independent of buffer constraints, therefore we assume infinite buffers on nodes.

The schemes we study are the same as the ones dropping schemes studied above. We briefly describe the schemes:

FIFO: Upon a node encounter, node \mathcal{N}_i forwards the oldest message in the buffer. Once the message has been forwarded it is reinserted to the back of the buffer.

LIFO: Upon a node encounter, forward the youngest message in the buffer. We reinsert the message to the beginning of the buffer after it has been forwarded.

Random: Forward a random packet from the buffer. This scheme is between the FIFO and LIFO schemes mentioned above.

We consider these policies in conjunction with our algorithms (Frequency, Destination Frequency and their delegation versions). In addition we consider two additional schemes which make use of the delegation number:

High (H): Upon a node encounter, forward the message which has the highest delegation number $d_i = \max_{\forall j}(d_j)$. This is analogous to forwarding the message which is closest to the destination as well the message which has been forwarded many times.

Low (L) Upon a node encounter, forward the message which has the smallest delegation number $d_i = \min_{\forall j}(d_j)$. This is analogous to forwarding the message which is relatively young in the network.

The results are presented in Fig. 7.3. The results for destination frequency are presented on top, and the results for frequency are presented below. Fig. 7.3 (a,c) display success rate versus cost. All delegation based schemes perform similarly. Epidemic does not perform well. We can therefore conclude that the epidemic algorithm does not perform well under resource constraints, as also observed before[RHB⁺07]. It would appear that dropping policies have more impact than transmission policies.

7.5 Relation to Path Explosion

Taken as a whole, it would appear that the delegation schemes perform much better under resource constraints. This is primarily because they reduce overall cost in the network, thereby reducing contention.

In addition, the augmented delegation schemes which use the delegation number to

prioritize message dropping/transmission perform better as they use the extra information available to them. In particular it would appear that the joint policy of dropping the message with the highest delegation number performs better than all schemes.

We interpret this result in light of the path explosion phenomenon first discussed in Chapter 4. The basic observation is that there is an exponential increase in the number of temporal paths between nodes in such networks. This means that for a given node pair, finding the optimal delay path is hard, but finding one of the many sub-optimal paths is a relatively easier task. This has implications on the performance of forwarding algorithms. However the question is how does one initiate a path explosion? While a comprehensive study of the necessary and sufficient conditions for the explosion to take place is beyond the scope of this work (some factors have been identified in Chapter 4), we posit that prioritization of messages play an important role for such a phenomenon to take hold.

By assigning higher priority to messages with a low delegation number, we are in essence ensuring that such messages do not get dropped, and have a chance to be replicated and propagate in the network. Therefore by prioritizing transmission of messages with a low delegation number, this increases the likelihood that explosion takes place for those messages. The use of delegation schemes further ensure that the message is replicated and forwarded to the best nodes at the minimum cost.

Likewise, when a message with a high delegation number is dropped, more likely than not path explosion for that message has occurred; there are other replicas of the message which can potentially be delivered to the intended destination. In addition, dropping a message which is ostensibly close to its destination does not increase the delay by a large factor, as due to the path explosion phenomenon, many near-optimal (in terms of delay) paths to the destination exist.

7.6 Summary

Efficient forwarding in mobile opportunistic networks is a challenging problem. While work has been done in the design of forwarding algorithms, little work has been done on studying forwarding under the presence of short contact durations and finite buffers. Under such circumstances, the problem of prioritizing messages for transmission or dropping becomes important.

In this chapter, we provide an empirical study of different schemes. In order to facilitate our study, we design a scheme based on earlier results of delegation forwarding. The main idea is to add a replication-count to delegation schemes. We design a message prioritization scheme based on this number.

We evaluate different schemes using real encounter traces. For ease of analysis we study forwarding and dropping policies separately. This allows us to isolate the effects of the two policies. Since our results show that forwarding policies have much less effect than dropping policies, we are able to conclude that careful selection of dropping policy is of primary importance. More specifically, we find that the scheme which assigns high priority to messages with low delegation number and lower priority for high delegation number performs best in terms of balancing success rate, delay and cost. We interpret this result in light of previous results related to the path explosion phenomenon: i.e. favoring young messages facilitates path explosion to occur for those messages, leading to the results we observe.

Chapter 8

Conclusions

The last decade has seen a shift from a “tethered” computing environment to completely untethered environment. The emergence of cheap yet powerful smart-phones and personal computing devices as well as advancements in communication technology have enabled this shift. Within this space of mobile computing, our focus is on mobile opportunistic networks, where all available communication opportunities including Internet, cell-phone technologies (GSM/CDMA) as well as physical mobility of nodes are exploited for passing messages.

The key problem facing MONs is that of routing of messages between mobile nodes. Nodes in MONs are highly mobile, have relatively short battery lives and have limited bandwidth. All these factors contribute in making the problem difficult. The main contribution of this thesis is the systematic attempt to design and study forwarding algorithms for MONs. We now summarize the main contributions of the dissertation.

8.1 Summary of Contributions

The starting point of the research documented in this thesis is the realization that in order to study forwarding, one should start by studying forwarding paths. By first understanding the nature of the paths, one can devise and study effective forwarding algorithms. Motivated by this realization, we proceeded to study forwarding paths using real measurements and all the results presented in this follow the realization. In particular, the dissertation makes

the following contributions:

- **Analysis of Forwarding Paths:** Using real measurements comprising contacts between individuals moving around in a conference environment, we enumerate forwarding paths between nodes in a MON. We observe that, on average, there exist many paths between a pair of nodes and soon after the first path arrives at the destination, the number of subsequent paths increase dramatically. We refer to this as the path explosion phenomenon. We use an analytical model to understand the reasons how and why path explosion occurs. While the analytical model is helpful in understanding different facets of path explosion, it falls short of explaining some outlier cases we observe in our data sets. In order to build a more comprehensive understanding of path explosion, we incorporate heterogeneity of contact rates in our study.
- **Study of Forwarding Algorithms:** The study of forwarding paths yielded results that inform the central question of this thesis: the study and design of forwarding algorithms. Towards answering this central question, we rely on trace-driven simulations to study the performance of algorithms that span different design choices. We find that most of the forwarding algorithms we study have very similar performance and we relate this result to the path-explosion effect. The algorithms we study tend to perform similarly because they typically find one of the many nearly-optimal paths that exist between node pairs.
- **Reducing Costs:** We found that, while the algorithms we study perform similarly in terms of success rate and average delay, they differ in terms of cost. This leads us to design algorithms with the explicit intent of reducing cost. Towards this end, we draw on the theory of optimal stopping to inspire *delegation forwarding*. We first show analytically that delegation forwarding reduces cost by a factor of $O(\sqrt{N})$, N being the number of nodes, over naive schemes. We study and compare delegation forwarding with other algorithms using real traces and show that the reduction in cost can be even higher. These results are presented in Chapter 6.

- Studying Forwarding under Resource Constraints: Our research thus far assumes infinite size buffers and infinite bandwidth. While these assumptions helped us in understanding the nature of forwarding and designing efficient forwarding algorithms, it is important to study forwarding without these assumptions, to make the leap towards more practical protocols. We attempt such a study in Chapter 7. We begin this study with the realization that under buffer and bandwidth constraints, the problem of forwarding shifts slightly to the problem of prioritizing of messages. We study different schemes of assigning priorities to messages, with the help of traces. Our main result is that the scheme which assigns high priority to messages which are young yields best results. We interpret this result in light of the path explosion effect.

Bibliography

- [AMW95] Miklos Ajtai, Nimrod Megiddo, and Orli Waarts. Improved Algorithms and Analysis for Secretary Problems and Generalizations. In *IEEE FOCS*, pages 473–482, 1995.
- [BAZ07] Vincent Borrel, Mostafa H. Ammar, and Ellen W. Zegura. Understanding the wireless and mobile network space: a routing-centered classification. In *CHANTS '07: Proceedings of the second ACM workshop on Challenged networks*, pages 11–18, New York, NY, USA, 2007. ACM.
- [BBC] BBC. Sales of smartphones to overtake those of laptops. <http://news.bbc.co.uk/2/hi/technology/7250465.stm>.
- [BBL05] Brendan Burns, Oliver Brock, and Brian Neil Levine. MV routing and capacity building in disruption tolerant networks. In *Proc. IEEE INFOCOM*, pages 398–408, March 2005.
- [BC03] Magdalena Balazinska and Paul Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 303–316, New York, NY, USA, 2003. ACM Press.
- [BCSW98] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A distance routing effect algorithm for mobility (dream). In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mo-*

- bile computing and networking*, pages 76–84, New York, NY, USA, 1998. ACM Press.
- [BdAF05] V. Borrel, M. Dias de Amorim, and S. Fdida. A preferential attachment gathering mobility model. *IEEE Communications Letters*, 9(10):900–902, 2005.
- [BGJL06] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. Max-Prop: Routing for Vehicle-Based Disruption-Tolerant Networks. In *Proc. IEEE INFOCOM*, April 2006.
- [BHT⁺03] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. Delay-tolerant networking: an approach to interplanetary internet. *IEEE Communications Magazine*, 41(6):128–136, 2003.
- [BKK⁺08] A.Z. Broder, A. Kirsch, R. Kumar, M. Mitzenmacher, E. Upfal, and S. Vassilvitskii. The Hiring Problem and Lake Wobegon Strategies. In *ACM-SIAM SODA 08*, 2008.
- [BLV07] Aruna Balasubramanian, Brian Levine, and Arun Venkataramani. DTN Routing as a Resource Allocation Problem. In *ACM SIGCOMM '07*, pages 373–384. ACM, 2007.
- [CB05] David R. Choffnes and Fabián E. Bustamante. An integrated mobility and traffic model for vehicular wireless networks. In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 69–78, New York, NY, USA, 2005. ACM.
- [CBD02] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing (WCMC) Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

- [CE07] Han Cai and Do Young Eun. Crossing over the bounded domain: from Exponential to Power-law inter-meeting time in MANET. In *ACM MobiCom '07*, pages 159–170. ACM, 2007.
- [CHC⁺05] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott. Pocket Switched Networks: Real-world mobility and its consequences for Opportunistic Forwarding. Technical Report UCAM-CL-TR-617, University of Cambridge, 2005.
- [CHC⁺07] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of Human Mobility on Opportunistic Forwarding Algorithms. *IEEE Trans. on Mobile Computing*, 6(6):606–620, 2007.
- [CLF06] Vania Conan, Jeremie Leguay, and Timur Friedman. The heterogeneity of inter-contact time distributions: its importance for routing in delay tolerant networks, 2006.
- [Dav00] V. Davies. Evaluating mobility models within an ad hoc network. *Master's thesis, Colorado School of Mines*, 2000.
- [DFGV03] Henri Dubois-Ferriere, Matthias Grossglauser, and Martin Vetterli. Age matters: efficient route discovery in mobile ad hoc networks using encounter ages. In *ACM MobiHoc '03*, pages 257–266, 2003.
- [DFL01] James Davis, Andy Fagg, and Brian Neil Levine. Wearable Computers and Packet Transport Mechanisms in Highly Partitioned Ad hoc Networks. In *Proc. IEEE Intl. Symp on Wearable Computers (ISWC)*, pages 141–148, October 2001.
- [DH07] Elizabeth M. Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant MANETs. In *ACM MobiHoc '07*, pages 32–40, New York, NY, USA, 2007. ACM.

- [ECCD07] Vijay Erramilli, Augustin Chaintreau, Mark Crovella, and Christophe Diot. Diversity of forwarding paths in pocket switched networks. In *ACM/SIGCOMM IMC*, pages 161–174, October 2007.
- [ECCD08] Vijay Erramilli, Augustin Chaintreau, Mark Crovella, and Christophe Diot. Delegation forwarding. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc)*, pages 251–259, May 2008.
- [EP05a] N. Eagle and A. Pentland. Reality mining: Sensing complex social systems. *Journal of Personal and Ubiquitous Computing*, 10(4):255–268, 2005.
- [EP05b] Nathan Eagle and Alex (Sandy) Pentland. CRAWDAD data set mit/reality (v. 2005-07-01). Downloaded from <http://crawdad.cs.dartmouth.edu/mit/reality>, July 2005.
- [Fal03] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–34, New York, NY, USA, 2003. ACM.
- [GGL07] Michele Garetto, Paolo Giaccone, and Emilio Leonardi. Capacity scaling in delay tolerant networks with heterogeneous mobile nodes. In *Proceedings of ACM MobiHoc '07*, pages 41–50, 2007.
- [GT02] Matthias Grossglauser and David N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Trans. Netw.*, 10(4):477–486, 2002.
- [GV06] Matthias Grossglauser and Martin Vetterli. Locating mobile nodes with ease: learning efficient routes from encounter histories alone. *IEEE/ACM Trans. Netw.*, 14(3):457–469, 2006.
- [Hag] Huggle. Huggle FET Project. <http://www.huggleproject.com>.

- [HCS⁺05] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *WDTN '05*, pages 244–251, New York, NY, USA, 2005. ACM Press.
- [HGPC99] Xiaoyan Hong, Mario Gerla, Guangyu Pei, and Ching-Chuan Chiang. A group mobility model for ad hoc wireless networks. In *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–60, New York, NY, USA, 1999. ACM Press.
- [HH06] Wei-Jen Hsu and Ahmed Helmy. On nodal encounter patterns in wireless lan traces. In *Proceedings of WiNMee*, 2006.
- [HKA04] T. Henderson, D. Kotz, and I. Abyzov. The changing usage of a mature campus-wide wireless network. In *Proceedings of ACM MobiCom*, 2004.
- [HW05] Emir Halepovic and Carey Williamson. Characterizing and modeling user mobility in a cellular data network. In *PE-WASUN '05: Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, pages 71–78, New York, NY, USA, 2005. ACM Press.
- [JBRAS03] Amit Jardosh, Elizabeth M. Belding-Royer, Kevin C. Almeroth, and Subhash Suri. Towards realistic mobility models for mobile ad hoc networks. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 217–229, New York, NY, USA, 2003. ACM Press.
- [JFP04] Sushant Jain, Kevin Fall, and Rabin Patra. Routing in a Delay Tolerant Network. In *ACM SIGCOMM '04*, pages 145–158, 2004.

- [JLB05] Ravi Jain, Dan Lelescu, and Mahadevan Balakrishnan. Model t: an empirical model for user registration patterns in a campus wireless lan. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 170–184, New York, NY, USA, 2005. ACM Press.
- [JLW05] Evan P. C. Jones, Lily Li, and Paul A. S. Ward. Practical routing in delay-tolerant networks. In *WDTN '05*, pages 237–243, 2005.
- [JM96] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.
- [KBV07] Thomas Karagiannis, Jean-Yves Le Boudec, and Milan Vojnovic. Power law and Exponential decay of inter contact times between mobile devices. In *MobiCom '07*, pages 183–194. ACM, 2007.
- [Kes05] S. Keshav. Why cell phones will dominate the future internet. *SIGCOMM Comput. Commun. Rev.*, 35(2):83–86, 2005.
- [KKK06] Minkyong Kim, David Kotz, and Songkuk Kim. Extracting a mobility model from real user traces. In *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Barcelona, Spain, April 2006.
- [Kur71] T. G. Kurtz. Limit theorems for sequence of jump markov processes approximating ordinary differential processes. *Journal of Applied Probability*, pages 344–356, 1971.
- [LDS03] Anders Lindgren, Avri Doria, and Olov Schelen. Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20, 2003.

- [LFC05] Jeremie Leguay, Timur Friedman, and Vania Conan. DTN routing in a Mobility Pattern Space. In *ACM WDTN '05*, pages 276–283, 2005.
- [LH03] Ben Liang and Zygmunt J. Haas. Predictive distance-based mobility management for multidimensional pcs networks. *IEEE/ACM Trans. Netw.*, 11(5):718–732, 2003.
- [LPC⁺04] Anukool Lakhina, Konstantina Papagiannaki, Mark Crovella, Christophe Diot, Eric D. Kolaczyk, and Nina Taft. Structural analysis of network traffic flows. In *Proceedings of ACM SIGMETRICS / Performance 2004*, pages 61–72, June 2004.
- [LW07] Cong Liu and Jie Wu. Scalable routing in delay tolerant networks. In *MobiHoc '07*, pages 51–60. ACM, 2007.
- [MAZ04] S. Merugu, M. Ammar, and E. Zegura. Routing in space and time in networks with predictable mobility. Technical Report GIT-CC-04-7, Georgia Institute of Technology, 2004.
- [MM06] Mirco Musolesi and Cecilia Mascolo. A community based mobility model for ad hoc network research. In *REALMAN '06*, pages 31–38, New York, NY, USA, 2006. ACM Press.
- [MMC08] Liam McNamara, Cecilia Mascolo, and Licia Capra. Media sharing based on colocation prediction in urban transport. In *MobiCom '08: Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 58–69, New York, NY, USA, 2008. ACM.
- [MV03] Marvin McNett and Geoffrey M. Voelker. Access and mobility of wireless pda users. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(4):55–57, 2003.
- [NDGG07] Erik Nordstrom, Christophe Diot, Richard Gass, and Per Gunningberg. Experiences from measuring human mobility using bluetooth inquiring devices. In

MobiEval 2007: System Evaluation for Mobile Platforms, New York, NY, USA, 2007. ACM Press.

- [PB94] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. In *Proceedings of ACM Sigcomm 1994*, pages 234–244, 1994.
- [PFH04] Alex (Sandy) Pentland, Richard Fletcher, and Amir Hasson. Daknet: Rethinking connectivity in developing nations. *Computer*, 37(1):78–83, 2004.
- [PR99] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.
- [Res] Gartner Research. Gartner Research article on Smartphone Penetration. <http://www.gartner.com/it/page.jsp?id=754112>.
- [RHB⁺07] Ram Ramanathan, Richard Hansen, Prithwish Basu, Regina Rosales-Hain, and Rajesh Krishnan. Prioritized epidemic routing for opportunistic networks. In *MobiOpp '07*, pages 62–66, New York, NY, USA, 2007. ACM.
- [RMSM01] E. Royer, P. Melliar-Smith, and L. Moser. An analysis of the optimum node density for ad hoc mobile networks. *Proceedings of IEEE ICC*, 2001.
- [SDPG06] N. Sarafijanovic-Djukic, M. Pidrkowski, and Matthias Grossglauser. Island hopping: Efficient mobility assisted forwarding in partitioned networks. In *IEEE SECON '06*, Reston, VA, USA, 2006.
- [Shi08] Albert N Shiryaev. *Optimal Stopping Rules*. Springer, 2008.
- [SJ04] Amit Kumar Saha and David B. Johnson. Modeling mobility for vehicular ad-hoc networks. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 91–92, New York, NY, USA, 2004. ACM.

- [SK99] John Scourias and Thomas Kunz. An activity-based mobility model and location management simulation framework. In *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 61–68, New York, NY, USA, 1999. ACM Press.
- [SMO06] Vikram Srinivasan, Mehul Motani, and Wei Tsang Ooi. Analysis and implications of student contact patterns derived from campus schedules. In *ACM MobiCom '06*, pages 86–97, New York, NY, USA, 2006. ACM Press.
- [SPR04] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi Raghavendra. Single-copy routing in intermittently connected mobile networks. In *IEEE SECON '04*, pages 145–158, San Jose, CA, USA, 2004. IEEE.
- [SPR05] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pages 252–259, New York, NY, USA, 2005. ACM Press.
- [SPR08] T. Spyropoulos, K. Psounis, and C. Raghavendra. Efficient routing in intermittently connected mobile networks: The multi copy case. *IEEE/ACM Trans. Netw.*, 2(2):477–486, 2008.
- [TB02] Diane Tang and Mary Baker. Analysis of a metropolitan-area wireless network. *Wirel. Netw.*, 8(2/3):107–120, 2002.
- [VAN] VANET. Vehicular ad-hoc networks. <http://en.wikipedia.org/wiki/VANET>.
- [VB00] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks. Technical Report CS-200006, Duke University, 2000.
- [WSG⁺04] Randolph Y. Wang, Sumeet Sobti, Nitin Garg, Elisha Ziskind, Junwen Lai, and Arvind Krishnamurthy. Turning the postal system into a generic digital

communication mechanism. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 159–166, New York, NY, USA, 2004. ACM.

- [ZNKT06] Xiaolan Zhang, Giovanni Neglia, Jim Kurose, and Don Towsley. Performance modeling of epidemic routing. In *Proceedings of IFIP Networking*, 2006.

Vita

Vijay Erramilli defended his Ph.D. thesis in Computer Science from Boston University on Forwarding algorithms for Mobile Networks. Previously, Vijay spent five months at Thomson Paris, where he worked on problems related to mobile networks. He has also worked on problems related to intra-domain traffic matrices. Vijay's research is predominantly empirically driven and his current research interests include social networks, network management and mobile networks.