Applications of Graph-based Codes in Networks: Analysis of Capacity and Design of Improved Algorithms

A Thesis
Presented to
The Academic Faculty

by

Badri Narayanan Vellambi

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2008

Applications of Graph-based Codes in Networks: Analysis of Capacity and Design of Improved Algorithms

Approved by:

Professor Faramarz Fekri, Chair, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Steven McLaughlin
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Raghupathy Sivakumar
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Prasad Tetali
School of Mathematics
Georgia Institute of Technology

Professor Geoffrey Li
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 19 August 2008

To my mother, father and my sister,

without whom this would not be a reality.

ACKNOWLEDGEMENTS

Honestly, this is the part I was hoping to write for a very long time. Apparently, fate wanted it to be the last part to be penned in my thesis. I have spent six long years in Atlanta. However, had my life not crisscrossed that of some of those I mention below, it would have been definitely longer, and certainly more arduous. In the first three years, I was more in company of my Indian friends. Though they are now “settled”, prosperous (now, we all know what that means, don’t we?) and relatively dormant, there are still moments that make me cherish my stay with them. The next three is what I would term the best part of my stay here. I had opened up to people of all cultures and races and have since come to meet so many wonderful people here. Some have left, some leave with me, and then there are some that I shall leave as I depart. Whatever said and done, graduate life is something that has come to provide me not only with multiple degrees behind my name, but also with a bunch of people that I have become fond of and have changed the way I look at life. In what follows, I briefly mention a few words thanking each one of them.

First of all, I would like to express my appreciation to my advisor, Dr. Faramarz Fekri, without whom this journey would have been everything but a mirage. Had I not received your call six years ago, I would not have had the chance to have so much fun academically and otherwise. Thank you!

I would also like to express my gratitude to the many professors that I have come to appreciate over the years here. Dr. Christopher Heil and Dr. Yang Wang of the mathematics department have been a subtle but a very strong driving force in propelling me towards a bright career. I would also like to thank Dr. Steven McLaughlin and Dr. Raghupathy Sivakumar for their support, help, suggestions and encouragement.

Shyam Sivakumar a.k.a. *Pops* – the one person who has constantly been with me during the ups and downs of my life and I hope it will be so for the rest of the journey too. Its hard not to think of David Boivin when I am writing this part. He is one of the nicest blokes I have had the pleasure of meeting! I shall always miss my tea sessions with him. Another gem of a person that I have had the pleasure of knowing is Kevin Chan. Call it good luck or fate, the past year would have been hellishly boring without him (*oui? non?*). I wish him the very best at the Army Research Laboratory.

I would also like to thank my colleagues and friends at CSIP, specially Nazanin Rahnavard, Matthieu Bloch, Arumugam Kannan and Mina Sartipi, without whom life would have been a little less spicy around work. My life at CSIP would not have been pleasant without a group of ladies who perform the various administrative duties on the floor. Patricia Dixon, Tammy Scott and Cordai Farrar have always been the there to count on whenever I landed myself into a dilemma. My many thanks to them! I have to add that in addition to all the help I have received, my inability discuss with Patricia at will is something that I shall sorely miss.

My life outside work in Atlanta has been equally fulfilling and meaningful. This would not have been possible if I had not met these awesome souls. It was just serendipitous that I met Venkat Ramachandran through whom I had the privilege of more than acquainting myself with Shiva and Subhashini and their six-month old princess, Srividya. They have been the biggest pillars of support for me, providing me with encouragement, happiness, satisfaction, and in a lighter vein, the food of the highest quality. I shall really miss being around them. I shall also miss my dear CRC-mates, Ramesh and Ramanan, who have become a part and parcel of my daily life. I am really fortunate to have met these contrasting but very nice people. The countless 3:00 PM coffee rituals that we performed together shall be deeply missed, to say the least. Of course, as I begin my next step in life, I shall also miss many friends whom I had the good fortune to interact, specially Chitra and Suresh Kodandaraman (and their children Smriti and Archit “Badri”), Kalpana and Rengarajan (I shall certainly be listening to karṇāṭic kacēris, and in each one of them I shall be reminded of the *joie de vivre* of Kalpana and the queer humor of Rengarajan).

I am certain that there have been many more people that have left an indelible mark over me during the six years than those I have had the joy of acknowledging here. To those, I apologize for my half-baked attempt at conveying my gratitude. Last but not the least, I am really thankful to God for giving me these invaluable memories. I am awestruck at the grace that He has decided to shower on me. I pray to Him to offer me the mind, the strength and the goodwill to offer to others at least an iota of what some of the people I have mentioned have offered me.

TABLE OF CONTENTS

PART I

Practical Coding Schemes with LDPC Codes

PART II

Reliable and Efficient Packet Delivery in Networks Via Modern Codes

PART III

Markov Chain Approach to Capacity of Finite-buffer Networks

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The conception of turbo codes by Berrou *et al.* has created a renewed interest in modern graph-based codes. Several encouraging results that have come to light since then have fortified the role these codes shall play as potential solutions for present and future communication problems.

This work focuses on both practical and theoretical aspects of graph-based codes. The thesis can be broadly categorized into three parts. The first part of the thesis focuses on the design of practical graph-based codes of short lengths. While both low-density parity-check codes and rateless codes have been shown to be asymptotically optimal under the message-passing (MP) decoder, the performance of short-length codes from these families under MP decoding is starkly sub-optimal. This work first addresses the structural characterization of stopping sets to understand this sub-optimality. Using this characterization, a novel improved decoder that offers several orders of magnitude improvement in bit-error rates is introduced. Next, a novel scheme for the design of a good rate-compatible family of punctured codes is proposed.

The second part of the thesis aims at establishing these codes as a good tool to develop reliable, energy-efficient and low-latency data dissemination schemes in networks. The problems of broadcasting in wireless multihop networks and that of unicast in delay-tolerant networks are investigated. In both cases, rateless coding is seen to offer an elegant means of achieving the goals of the chosen communication protocols. It was noticed that the ratelessness and the randomness in encoding process make this scheme specifically suited to such network applications.

The final part of the thesis investigates an application of a specific class of codes called network codes to finite-buffer wired networks. This part of the work aims at establishing a framework for the theoretical study and understanding of finite-buffer networks. The proposed Markov chain-based method extends existing results to develop an iterative Markov chain-based technique for general acyclic wired networks. The framework not only estimates the capacity of such networks, but also provides a means to monitor network traffic and packet drop rates on various links of the network.

CHAPTER I

INTRODUCTION AND RELATED WORK

Today, efficient and reliable transmission and storage of data have become an important part of our lives. On one hand, the developments in systems engineering and electronics have effected a significant increase in the efficiency of our communication systems. On the other, developments in the areas of material science and physics furnish us with communication channels and media superior than previously available. However, the presence of noise and other unwanted phenomena in almost all physical channels of communication is a reality we must accept. Such unwanted phenomena distort and attenuate the signals transmitted over these channels. Noise and other such phenomena are simply the *raison d'être* for the dynamic and challenging subfield of communication theory – *error control coding* (or *channel coding* or just *coding*). The idea behind channel coding is the controlled addition of redundancy to the transmitted signal by the process of *encoding*. This added redundancy is exploited at the receiver during the process of *decoding*. The aim of intelligent encoding and decoding is to yield at the receiver a near-replica of the signal that was transmitted by the source.

The foundation of channel coding was laid by Shannon in his pivotal work on communication theory [105, 106]. Shannon established the fundamental limits on the rate of information that could be transmitted over noisy channels. In his work, he proposed random codebook constructions to also show that these information-theoretic limits were achievable over a family of channels. However, this proof rested on the impractical supposition that the computational complexity of the decoding procedure is not of any concern. It was therefore natural for the researchers to study the construction of practical encoding and decoding techniques that almost achieve this theoretical limit. Since random code construction of Shannon did not provide sufficient structure to the encoding procedure and the code itself, the design of mathematically tractable methods for code construction were explored.

Algebraic techniques of encoding and decoding were the first to be explored systematically. *Linear block* codes such as Hamming codes [44], Reed-Muller (RM) codes [72], *cyclic* codes such as Bose, Ray-Chaudhuri, and Hocquenghem (BCH) codes and Reed-Solomon (RS) codes [72] were examples of exemplary code constructions that resulted from algebraic constructions.

In 1963, Gallager proposed a class of graph-based codes called *low-density parity-check* (LDPC) codes that were ignored almost entirely because of the computational complexity of the operations involved. However, in 1993, the invention of *turbo* codes by Berrou *et al.* [12] prompted a second glance at LDPC codes. The advances made in computer engineering and electronics in the three decades that had passed after the conception of LDPC codes were sufficient for their encoding and decoding processes to be considered feasible and practical. This led to what can be termed a renaissance in the field of error control coding. Algebraic techniques have taken a backseat since then and graph-based techniques of code construction and iterative modes of decoding have become an active area of research.

Extensive research on the efficient encoding and decoding of LDPC codes have yielded significant improvements [102, 67, 101, 99]. Recent research on these codes have established the design of families of codes that perform very close to the fundamental limits over certain channels. More specifically, LDPC codes have been shown to be capacity-achieving over the *binary erasure channel* [111, 84, 110]. While LDPC codes have several traditional features of codes like fixed length and rate, a newer class of iterative graph-based codes such as *fountain* (or *rateless*) codes have relinquished the idea of rate that has been common to codes discovered hitherto. These codes were introduced by Luby *et al.* [66, 65] and are suited for channels wherein feedback from the receiver is available at the source. It was shown that the design of jointly optimal rateless codes for parallel independent channels with differing channel erasure losses is possible. Another interesting feature of these codes is that over the erasure channel with feedback, optimality can be guaranteed, immaterial of the knowledge of channel erasure probability.

Broadly, this work investigates certain practical and theoretical aspects of the design and analysis of modern coding techniques and its application to practical networks. The

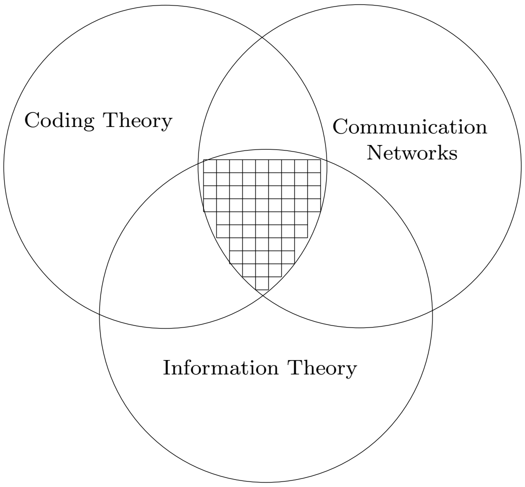

Figure 1.1: Areas pertinent to this work.

sub-fields of communication theory pertinent to this work are illustrated in Figure 1.1. The body of work can be divided into three sections. The first part focusses on the design and analysis of practical LDPC coding schemes that are universally applicable, whereas the second part concerns with the application of graph-based codes specifically for improving communication strategies in multihop wireless and delay-tolerant networks. The final part of the work deals with the estimation of information-theoretic limits in simple wired networks that are constrained in buffer size. Relevant past work in these areas and an outline of our contribution are presented below.

1.1 Practical Coding Schemes with LDPC Codes

Since the conception of turbo codes, LDPC codes have gained wide acceptance not only in the research community but also in the telecommunication industry. The design of practical coding schemes that minimize computational cost and ensure good performance has attracted a lot of attention in the research community. Some of the several relevant problems that have been explored include the analysis of finite-length LDPC codes [25], the design of efficient encoders [102], the design of improved decoding schemes, and the design of rate-compatible coding systems for varying channel conditions. In each of these aforementioned problems, practical considerations translate to either reducing the computational complexity or maintaining it within the acceptable limits. In this work, we focus on two problems, namely, 1. the design of improved decoding algorithm for the *binary erasure channel* (BEC), and 2. the design of good rate-compatible family of codes.

The *message-passing* (MP) decoder [101] for LDPC codes is a simple algorithm that decodes by passing suitable messages over the edges of the Tanner graph of the LDPC code. It has been shown that the MP decoder guarantees error-free performance as the codelength approaches infinity at code rates arbitrarily close to the channel capacity [84]. However, in short-length LDPC codes, the performance of the MP decoder and that of the optimal *maximum-likelihood* (ML) decoder differ substantially. In fact, a difference of several orders of magnitude in the bit-error rate offered by the MP decoder and the ML decoder can be noticed in many LDPC codes. This stark degradation in the performance when the codelength is short is attributed to the presence of a class of sets called stopping sets [25]. In order to mitigate the effect of stopping sets, it is desirable that the Tanner graph possesses a large girth. Considerable research on combinatorial constructions that guarantee large girth and techniques that eliminate short cycles in Tanner graphs is available [58, 119, 27].

Another common approach to this problem is the design of improved MP decoders that employ auxiliary parity-check equations or performing additional operations such as bit-guessing. Bit-guessing has been affirmed as a practical means of improving the MP decoder and providing a huge improvement in the bit-error rate (BER) [92]. In [93], P. Nik *et al.* introduce schemes that perform bit-guessing after the MP decoding of the codeword comes to a halt. The number of bits guessed is limited to keep the additional computational cost incurred within acceptable limits. Although [93] has established the merit of bit-guessing, certain questions on the number and an intelligent sequence of bits to be guessed were left unanswered. Our work in this area answers these questions convincingly. Our contributions and improvements in this area are summarized as follows.

1. We provide a complete matrix characterization of ML-decodable codewords that are also MP-decodable. More specifically, we present a structural characterization of all the stopping sets that can be decoded by guessing a fixed number of bits.

2. We observe that at certain channel conditions, a considerable number of bivalent check nodes remain in the stopping set after the message passing is completed. This observation is analytically shown to be true for the general class of left-regular LDPC codes when the codelength is made arbitrarily large.

3. We propose the *contraction-based message-passing* (CMP) decoder that exploits the presence of bivalent check nodes using graph contraction. The CMP decoder also provides a naturally efficient rule for selection of bits that have to be guessed to ensure fast decoding. The bit-error rate of the proposed scheme is noted to be several orders of magnitude lower than that of the MP decoder and other practical decoders of interest.

The second problem that we investigate is the design and analysis of rate-compatible puncturing of finite-length LDPC codes. While communicating over channels that have losses whose characteristics vary with time, it is necessary that the coding system employed exploits the temporal change in the loss phenomenon. For example, in channels where information about the average channel loss rate is available and is varying with time, it is necessary that the rate of the code employed is varied according to the changes in the capacity of the channel. This is usually achieved by modifications to a single code by techniques known as puncturing and extending [37, 83, 14, 126, 60]. These techniques selectively modify the *vector space* of codewords to effect a change in the rate of the code.

When such channel conditions exist, it is desirable to design a rate-compatible coding system that employs a single encoder-decoder pair operating at different rates at different channel conditions. Several approaches to the problem of rate-compatible puncturing of LDPC codes exist in literature. [60] considers the problem of designing a good rate-compatible ensemble of LDPC codes for a specific range of rates. In [14], authors consider the problem of searching and designing a rate-compatible sequence of codes (using the differential evolution procedure [108]). The codes of the designed sequence operate at different rates and each of them offers performs well at its rate. In [126], the authors choose a mother code at a rate that is in-between the range of required rates so as to minimize the excessive degradation caused by puncturing at high rates. To achieve a higher rate, a random puncturing technique is used. Lower rates are achieved by extension using a novel progressive edge-growth based scheme [49]. Unlike others, Ha *et al.* investigate the problem of puncturing a given LDPC mother code [37, 41, 83]. They have shown by simulations that over the *additive white Gaussian noise* (AWGN) channel, the codes constructed by their puncturing scheme is superior to those obtained from random puncturing schemes. Their puncturing scheme utilizes a pair of algorithms known as *grouping* and *sorting* algorithms that selects the bits to be punctured in a particular order [41]. We present a novel criteria for puncturing bits of a given parent code that results in an improved rate-compatible family of LDPC codes. Our approach is similar to that taken in [41, 37]. However, the criterion for the selection of the bits is loosely based on a metric of distance between punctured nodes. Additionally, we also investigate the average degradation in performance (bit-error rate) due to the random puncturing of a given LDPC code. Our contributions to this problem are summarized as follows.

1. We propose a new puncturing scheme that selects bits based on the average distance between the previously selected nodes in the Tanner graph of the LDPC code. The proposed scheme performs well for a wide range of punctured rates outperforming similar schemes of interest.

2. We propose a novel idea of estimating the average bit-error rate of randomly punctured codes obtained from a given mother code. The average performance of the family of punctured codes at a particular channel condition is estimated using the performance of the mother code at a different channel condition. It is noticed that such an estimation offers a good prediction of the behavior of punctured codes.

1.2 Reliable and Efficient Packet Delivery in Networks Via Modern Codes

Coding techniques have been pivotal in establishing reliability in network applications. Traditional algebraic codes have been used in networks in the form of *cyclic redundancy checks*

[90, 117] to detect errors during transmission. Error control codes have been well established as a means of guaranteeing both reliability and low latency. Recently, considerable attention has been devoted the design of communication strategies incorporating coding mechanisms for not only combating losses, but also enhancing performance metrics such as reliability and latency [98, 122, 50, 71, 69, 68]. Our work in this area is motivated by the huge benefits in performance measures such as latency, energy-efficiency and reliability these coded schemes offer. Broadly, our work concerns the design and analysis of communication strategies employing modern codes in wireless multihop and delay tolerant networks and is outlined below.

The first main problem that we study is the broadcasting of a large volume of data in a lossy wireless multihop network. By the wireless nature of these networks, any transmission made by a node in the wireless channel is heard by all nodes within its communication range. Therefore, even in the presence of losses, a single transmission can potentially benefit several neighboring nodes. This property necessitates starkly different approaches for the design of broadcasting schemes in wireless networks and wired networks.

If the nodes are allowed to only forward packets, then the optimal broadcasting in a lossless wireless network reduces to the problem of finding a *minimum-connected dominating set* (MCDS) for the corresponding network graph. Unfortunately, determining an MCDS for a general network is an NP-hard problem [31] even if a centralized algorithm utilizing the full knowledge of the graph topology is employed. Some heuristic algorithms for attacking this problem have been proposed in [79, 42, 116, 87, 124, 96, 63]. Several other approaches for reliable, energy-efficient, scalable schemes for broadcasting in wireless networks have also been proposed in literature [42, 79, 124, 116, 87, 82]. While some of these schemes assume complete knowledge of the network, others assume partial information of the network alone, and certain other schemes assume no knowledge of the network.

Among the schemes that assume no knowledge of the network, the *flooding* mechanism [81] and its natural extension, the *probabilistic broadcast* (PBcast) scheme have been explored in detail [55, 103, 127, 46]. In the flooding scheme, each node broadcasts every packet that it receives whereas in the PBcast scheme, each node that receives a packet decides to forward the received packet to each of its neighbors with a probability $p < 1$. On one hand, probabilistic forwarding of packets effects a reduction in the total number of transmission made by all the nodes, and increases both the bandwidth- and the energy-efficiency in comparison to flooding. On the other, probabilistic forwarding does not guarantee 100% reliability since there is a strictly non-zero probability that all neighbors of a node decide not to transmit a particular packet. The reliability of the PBcast scheme for random deployment networks exhibits a threshold phenomenon very similar to that of its radius of connectivity [35]. However, the variation of the reliability offered by the PBcast scheme in grid networks follows a different trend due the difference in the geometry of the node neighborhoods in the two network models. We analyze the reliability of the PBcast scheme in the specific setting of Manhattan grid networks by employing algebraic and combinatorial path-enumerative techniques. The derived bounds enable us to estimate the probability of forwarding that is required to guarantee any given probability of receipt at any node of the grid.

Another problem that this work discusses is the design of energy-efficient broadcasting via modern codes inspired by developments such as *network coding* [56, 62, 69, 71]. Recently, the idea of allowing intermediate nodes to selectively process (i.e., decode and or re-encode) the data that is routed through them has opened up a new area of research known as network coding. While the results from network coding are promising, the model of networks assumed in the bulk of the literature is fairly ideal. However, there are practical schemes such as CODEB [61] that employ network coding to increase reliability but assume only local knowledge of the network topology unlike the schemes in [69, 71]. Similarly, schemes such as CRBcast [97] employ rateless codes to effect reliability, energy efficiency and scalability. It is noted that while both network coding and rateless coding achieve the same goal of reliability, the latter offers significantly lower decoding costs than the former albeit with a marginal increase in the cost of encoding. We introduce a novel multi-stage broadcasting scheme known as *fractional transmission scheme* (FTS) that employs the *decode, re-encode and forward* paradigm with rateless codes. Our contributions to the above problem are summarized below.

1. We propose the FTS that uses of knowledge of the neighborhood of a node to determine the required fraction of data that must be transmitted so that each node in its neighborhood receives sufficient number of packets. Rateless code is employed in the scheme for three reasons: 1. to ensure that different packets sent by different nodes are innovative and hence increase the efficiency in the utilization of bandwidth, 2. to ensure high reliability and to minimize coding overhead, and 3. to ensure that the decoding complexity is acceptable.

2. We analyze the FTS to derive bounds that accurately predict (up to a scaling factor) the asymptotic variation of the average cost of transmission per packet per node with the number of nodes. Further, we note from simulations that the scheme performs very well in comparison to network coding when the natural sense of direction is assigned to different communication links.

The second type of network that this work focusses is the design and analysis of unicast schemes in *delay tolerant networks* (DTNs). DTNs are characterized by intermittent connectivity, the lack of end-to-end feedback, mobility of nodes, and opportunistic communication. The lack of permanent paths from the source to the destination and the dynamics of nodes in such networks make the study of communication strategies like unicast and multicast very challenging. Traditional approaches to unicast and multicast in networks such as the design of efficient routing schemes cannot be directly applied to DTNs. Several efficient schemes tailored to these networks were devised recently [51, 122, 78, 129, 19, 123]. These schemes predominantly use the *store, carry, and forward* paradigm for message delivery. Certain other approaches use partial or complete knowledge of the network dynamics to effect efficient routing [51, 53, 78]. Message-ferry based schemes have been suggested in [129, 130, 118], where some mobile nodes that move in a predictable fashion, known as ferries, provide communication service in the region of deployment. Epidemic routing, aimed at minimizing latency, has also been suggested as a viable solution to the problem of message delivery in DTNs [120, 64]. In epidemic routing, multiple *identical* copies of messages are injected into the network, and node mobility is relied upon to transfer all the requisite data packets to the destination. An intermediate node (other than the source and destination, called a *relay* node) transfers a copy of its packets to a node that it is in contact with if the latter does not already have a copy of them.

Although considerable research is available for efficiently routing messages in DTNs, most of them employ simple replication of packets and multiple transmissions to ensure higher reliability and lower latency. Recently, hybrid schemes, wherein both replication of messages and simple fixed-rate erasure-coding are employed, were shown to have great potential as robust, efficient solutions for DTNs [122, 50]. It was shown in [122] that such hybrid routing strategies employing both coding and replication are more robust than simple replication. However, these works are either very simplistic or are based not unrealistic assumptions. We introduce the *rateless coding based* (RCb) scheme that ensures a significantly better reliability and delivery delay performance compared to existing schemes. Several aspects of rateless codes make them apt for such applications. First, their rateless nature does away with issues regarding a good choice of rates even in the presence of varying channel loss conditions. Second, they are packet-level codes that have low complexity of encoding and decoding and require very low coding overhead to recover the message [109]. Our contributions to the above problem are summarized below.

1. We propose the RCb scheme that utilizes rateless coding at the source to enhance both the reliability and latency profile at minimal overheads of energy. The improvement in performance metrics such as reliability, latency, energy-efficiency offered by our scheme is substantial over that of other coded and uncoded schemes such as [122].

1.3 Markovian Approach to Capacity Estimation in Networks

In this work, we focus on the study of capacity in two different class of networks using Markov chains. In the first half of this part, we focus on the computation of unicast throughput between a given pair of nodes. We consider a simple grid network over which nodes with finite buffers perform a random walk. Similar work for random deployment network models are available [34, 26]. Also, some results on the design of good routing schemes can be found in [113, 114], wherein the authors study the average latency that a fixed number of injected packets in controlled buffer conditions. The common approach to this problem has been the Poisson approximation [33, 1, 45, 52]. However, we show that for the model of random walk on grid, the Poisson approximation does not capture the statistics of the buffers. As a result, the throughput prediction becomes inaccurate. Our work in this area fills some of the inadequacies of this approach. We model the network using Markov chains [32, 54] and reduce the problem to computation of the steady-state probability of the chain. The modeling for this problem is fairly generic to accommodate the effects of both finite buffer of relay nodes and the event of contention of nodes. Our contributions to this problem are as follows.

1. We propose a Markov-chain based approach that incorporates realistic network settings such as finite buffer sizes and contention of nodes to analyze the unicast throughput of grid-based DTNs. The proposed approach models the steady-state buffer conditions more accurately than the Poisson approximation. The approach also provides a means for the exact computation of the throughput of a two-hop delivery scheme when the source-destination pair is immobile.

2. Using a suitable independence assumptions, an approximate Markov chain for the mobile source-destination case is studied. Our model offers considerable improvement in the accuracy of throughput estimation over the Poisson approximation model.

The second problem that we investigate is the computation of the information-theoretic capacity of networks. Its study is important for not only designing efficient coding schemes, but also for efficient buffer allocation. The problem of computing capacity and designing efficient coding schemes for lossy wired and wireless networks has been widely studied [24, 23, 85, 56, 62]. However, the study of capacity of networks with finite buffer sizes has been limited. This can be attributed solely to the fact that the analysis of finite buffer systems are generally analytically harder to track. With the advent of network coding [62, 71, 69, 68] as an elegant and effective tool for attaining maximum network performance, the interest in the study of the effect of finite buffers on the capacity has been increased.

The problem of studying capacity of networks with finite buffers has also been visited in queueing theory in the form of stochastic networks. The problems can be seen to be similar in the sense that the packets can be viewed as customers and the delay due to packet loss in the link as the arbitrary service time. Furthermore, there is a subtle difference in the packet-customer equivalence when the network has nodes that have more than three neighbors. When this is the case, the node can choose to duplicate packets on both the links to maximize throughput, an event that cannot be captured directly in customer-server based queueing model. Also, the modeling of packet overflow in the network is achieved by a type II blocking (also known as *blocking after service*) in stochastic networks. Therefore, the problem of finding capacity in certain networks is then seen to be identical to determining the arrival/departure rates of certain random processes in an open stochastic network of a given topology [4, 3, 16, 112, 22, 104]. However, most relevant works in the field of queueing theory consider the continuous-time model for arrival and departure of packets in the network. In [70], Lun *et al.* consider the discrete-time analogue of the arrival process by lumping the time into epochs wherein each node can transmit and receive a packet. However, the authors analyze the information-theoretic capacity of a simple two-hop lossy network. While our approach employs a model of network similar to that in [70], we extend their results to derive bounds and estimates for the capacity of line networks of any hop-length and intermediate node buffer size. Further, we extend our ideas on line networks to general wired acyclic networks assuming that the network is directed. Our contribution to this area of research is summarized below.

1. We present a Markov-chain based modeling of line networks using that we exploit to derive bounds and good estimates for the capacity of line networks. It is specifically noted that the estimate tracks the variation of capacity as a function of buffer size and channel loss parameters closely.

2. We combine the results of line networks with traditional flow interpretation of information [21, 15] to derive estimates on the capacity of general wired acyclic directed networks.

The rest of the dissertation is organized in the following way. Chapter 2 presents a list of the notations used in this dissertation. Chapter 3 describes a brief introduction to the various coding schemes that are of interest to the body of work. Part I presents our contribution to the analysis and design of improved algorithms for decoding and puncturing of LDPC codes. Part II presents our contributions to the design of energy-efficient and reliable schemes of communication in wireless multihop and delay tolerant networks. Part III presents our contributions to the study of capacity of networks using Markovian techniques. Finally, Chapter 12 concludes the dissertation.

CHAPTER II

NOTATIONS

In this chapter, we present an appropriately classified list of all notations that are used in the thesis.

2.1 Logic, Elementary Set Theory and Analysis

Here, upper case letters $A, B, \ldots$ denote sets and lower case letters $a, b, \ldots$ denote the set elements.

$\wedge$	Logical *and*.
$\vee$	Logical *or*.
$\oplus$	Logical *xor*.
$\emptyset$	The empty set.
2^A	The set of all subsets of A (also known as the *power set* of A)
A^c	The complement of the set A.
$A \times B$	The Cartesian product of the set A with B.
$A \subsetneq B$	A is a subset of B but $B \neq A$.
$\mathbb{N}$	The set of natural numbers $\{1, 2, \ldots\}$.
$\mathbb{Z}$	The set of integers $\{1, 2, \ldots\}$.
$\mathbb{Z}_{\geq 0}$	The set of whole numbers $\{1, 2, \ldots\}$.
$\mathbb{Q}$	The set of rational numbers $\{1, 2, \ldots\}$.
$\mathbb{R}$	The set of real numbers $\{1, 2, \ldots\}$.
$\mathbb{I}_A$	The indicator function of the set A defined as $\mathbb{I}_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$

2.2 Probability and Statistics

$\mathbb{G}(t)$ $(t \in \mathbb{R})$	The geometric distribution with mean inter-arrival time t.
$B(n,p)$ $(n \in \mathbb{N}, p \in (0,1))$	The binomial distribution with n independent trials each with a success probability p.
$\langle f \rangle$ (f: a p.d.f.)	A shorthand for $\int\limits_{-\infty}^{\infty} x f(x) dx$.

2.3 Vectors and Matrices

Here, upper case letters $A, B, \ldots$ usually denote matrices and bold case letters $\mathbf{a}, \mathbf{b}, \ldots$ denote vectors.

$\lVert A \rVert$	The determinant of A.
$\Re(B)$	The vector of row sums of the B.
$A \sim B$	A and B are row and/or column permutations of each other.
$\mathbb{1}_{(k_1,k_2)}$ $(k_1, k_2 \in \mathbb{N})$	The $k_1 \times k_2$ all-one matrix.
$\mathbf{a}_i$	The i^{th} component of the vector $\mathbf{a}$.

2.4 Special Functions

$\sigma(k)$ $(k \in \mathbb{Z})$	$\mathbb{I}_{\mathbb{Z}_{\geq 0}}(k)$
$\delta[k]$ $(k \in \mathbb{Z})$	$\sigma(k) - \sigma(k-1)$.
$(x)^+$ $(x \in \mathbb{R})$	$\frac{x + \lvert x \rvert}{2}$.
$\overline{x}$ $(x \in \mathbb{R})$	A shorthand for $(1-x)$.
$h(x)$ $(x \in [0,1])$	The binary entropy function $h(x) = \left\{ \begin{array}{cc} -x \log_2 x - \overline{x} \log_2 \overline{x} & x \in (0,1) \\ 0 & \text{otherwise} \end{array} \right.$.

2.5 Graph Theory and Combinatorics

Here, upper case letters $G, H, \dots$ represent graphs on the set of vertices $V_G, H_G, \dots$, respectively. The lower case letters $u, v, w \dots$ represent vertices.

$N_G(u)$	The neighbors of the vertex u in the graph G.
$\partial_G(v)$	The degree of v in the graph G.
$\Delta(G)$	The girth of the graph G.
$C_2(N_G(w))$	The set of bivalent vertices in the neighborhood of the vertex w.
$G[S] \quad (S \subset V_G)$	The subgraph induced by the subset S of vertices.
$\mathfrak{S}_k \quad (k \in \mathbb{N})$	The set of all permutations on k objects.
$\mathscr{C}(\sum_{i=1}^{k} \alpha_i x^i, x^j) \quad (k \in \mathbb{N},\, \alpha_i \in \mathbb{R})$	The j^{th} coefficient α_j in $\sum_{i=1}^{k} \alpha_i x^i$.

2.6 Miscellaneous Notations

$a \propto b$	variable a is directly proportional to variable b.
$a \approx b \ (a, b \in \mathbb{R})$	a is approximately the same as b.
$a \vert b \ (a, b \in \mathbb{Z})$	a divides b.

CHAPTER III

A BRIEF INTRODUCTION TO MODERN CODES

In this section, we present the background details pertinent to the work that is presented in the sections that follow. In Section 3.0.1, we introduce the binary erasure channel. In the two sections that follow thereafter, LDPC and fountain codes are introduced. In Section 3.0.4, the decoding of graph-based codes, the class within which both LDPC and fountain codes fall, is described for the binary erasure channel.

3.0.1 Binary Erasure Channel

The *binary erasure channel* (BEC) was first introduced by Elias in 1955 [28] and was deemed a theoretical channel until the emergence of the Internet. Today, most data networks can be safely assumed to be packet erasure networks. The basic notion in a binary erasure channel is that of an erasure. In a BEC, a bit is either received as is or is erased. Mathematically, a BEC is a memoryless channel that accepts as input any binary sequence $\mathcal{I} = \{i_n\}_{n\in\mathbb{N}}$. However, the sequence of outputs $\mathcal{O} = \{o_n\}_{n\in\mathbb{N}}$ is from an alphabet of size three, i.e., $o_n \in \{0, 1, ?(\text{erasure})\}$, $\forall n \in \mathbb{N}$. The following table presents the channel matrix that characterizes the channel precisely. Also, it can be verified that the BEC is parameterized by exactly one variable, the erasure probability $\varepsilon \in [0, 1]$, and that the information-theoretic capacity of this channel is given by $C_{\text{BEC}} = 1 - \varepsilon$. Further, it is worth noting that a BEC with zero erasure probability is noiseless and that with unit erasure probability is useless.

Table 3.1: The channel matrix of the BEC.

Output bit (x)	$P[o = x\|i = 0]$	$P[o = x\|i = 1]$
0	$1-\varepsilon$	0
1	0	$1-\varepsilon$
?	ε	ε

3.0.2 Low-Density Parity-Check Codes

Discovered by Gallager [30] in 1963, *low-density parity-check* (LDPC) codes were almost forgotten till turbo codes came into existence in 1993 [12]. Loosely speaking, these codes are defined by low-density parity-check matrices, i.e., the number of non-zero entries in the matrix forms a small fraction of the total number of entries. Central to the concept of LDPC codes is that of a Tanner graph. A Tanner graph is a bipartite graph $G(V, E)$, where the set of vertices is the disjoint union of two parts denoted by V and C. The elements of V and C are called as bit nodes (or variable node) and check nodes, respectively. Every edge in G has an end that is a node in V and the other that is a node in C. Moreover, the degree of a node in the Tanner graph is defined to be the number of edges that are incident on it. The Tanner graph of an LDPC code defined by a parity-check matrix H is constructed in the following manner. Each column of H corresponding to each bit in the codeword is assigned to a bit node in V and each row (equation) in H is assigned to a check node in C. The edges in the Tanner graph connect a bit node to a check node if and only if the bit corresponding to the bit node participates in the equation corresponding to the check node. Throughout the dissertation, as an abuse of notation, we would use bit node and bits interchangeably. The same is the case with check nodes and parity-check equations. It must be noted here that a code has multiple Tanner graphs, depending on the parity-check matrix used for defining the code. As an example, consider the following parity-check matrix.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{3.1}$$

The Tanner graph for the parity-check matrix of (3.1) is given in Figure 3.1. The bit nodes are all shown by circles whereas the check nodes are shown by shaded squares. Clearly, there are 8 bit nodes in V corresponding to the 8 bits in the codeword and in the same order. Similarly, the number of check nodes and equations match. It can be verified that

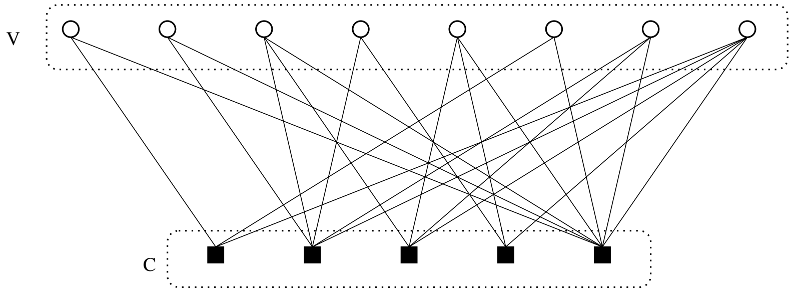

Figure 3.1: Tanner graph illustration of the parity-check matrix H of (3.1).

the number of edges in the graph equals the number of ones in H and that each check node is connected to only those bit nodes that participate in the check equation that the check node represents.

In the literature, LDPC codes are more often defined by their (edge) degree distributions than by the parity-check matrices or Tanner graphs. A family of LDPC codes is defined by a pair of polynomials (λ, ρ), the former is called the bit node degree distribution and the latter is called the check node degree distribution. An LDPC ensemble $\mathcal{G}^n(\sum_i \lambda_i x^{i-1}, \sum_i \rho_i x^{i-1})$ represents the set of all codes of length n that have parity-check matrices in whose Tanner graph representations λ_i represents the fraction of edges that are connected to a bit node of degree i. Similarly, in this ensemble, ρ_i denotes the fraction of edges that are connected to check nodes of degree i. For example, the above example can be seen to be one of the codes in the ensemble $\mathcal{G}^8(\lambda^*, \rho^*)$, where $\lambda^*(x) = \frac{4}{11}x + \frac{9}{22}x^2 + \frac{5}{22}x^4$ and $\rho^*(x) = \frac{3}{11}x^2 + \frac{2}{11}x^3 + \frac{5}{22}x^4 + \frac{7}{22}x^6$. For an ensemble $\mathcal{C}^n(\lambda, \rho)$, it can be shown that the number of edges in the Tanner graph and the *design rate*[1] of the code are given by $E(\lambda, \rho) = \frac{n}{\int_0^1 \lambda(x)\, dx}$ and $r(\lambda, \rho) = 1 - \frac{\int_0^1 \rho(x)\, dx}{\int_0^1 \lambda(x)\, dx}$, respectively. An ensemble $\mathcal{C}^n(\lambda, \rho)$ is constructed via the *socket method* in the following fashion. Partition V randomly into subsets V_i, $i = 1, 2 \ldots$ such that $|V_i| = E(\lambda, \rho)\frac{\lambda_i}{i}$. Similarly, divide C into subsets C_i with $|C_i| = E(\lambda, \rho)\frac{\rho_i}{i}$ check nodes. Assign i distinct sockets to each node in V_i and C_i. In total, there would be $\sum_i i|V_i| = \sum_i i|C_i| = E(\lambda, \rho)$ sockets in V and C alike. Now, inductively connect a random unconnected bit node socket to an unconnected check node socket. Finally, the Tanner graph is constructed by collapsing all sockets of a node into one vertex. Each construction,

[1] The actual rate of the code can be higher because of redundant equations in the parity-check matrix.

because of the randomness, potentially yields a different Tanner graph in the given ensemble and every Tanner graph in the ensemble can be constructed in this manner eventually. Here, it must be mentioned that the collapsing the sockets may not yield a simple graph. Therefore, to construct a proper code, the process can be repeated multiple times. Finally, the codes that are used for simulations and verification of the proposed results are notated in Table 3.2.

Table 3.2: Notation for the ensembles used in this work.

Notation	Degree Distributions ($\lambda(x)$, $\rho(x)$)
C1	$(x^2,\, x^5)$
C2	$(x^3,\, x^4)$
C3	$(0.0796x + 0.6923x^2 + 0.2308x^5,\, 0.4615x^5 + 0.5385x^6)$
C4	$(0.4706x^2 + 0.2353x^7 + 0.2941x^{29},\, 0.7843x^9 + 0.2157x^{10})$
C5	$(0.2223x + 0.3884x^2 + 0.1934x^7 + 0.1959x^{14},\, 0.78x^6 + 0.22x^7)$

3.0.3 Fountain (Rateless) Codes

Fountain codes are a new class of robust and promising class of graph-based codes first introduced in [66]. Since then, several classes of rateless codes such as LT codes, raptor codes [109], and online codes [73] have been conceived. From a given set of input symbols x_i, $i = 1, \ldots, k$, the fountain encoder can virtually generate an infinite stream of output symbols that are output sequentially. Commonly, the symbols are either bits or data packets (binary vectors). Central to the encoding process is the generator polynomial Ω that defines the packet selection process. Let $\{\Omega_i \geq 0\}_{i=0}^{k}$ denote a probability distribution on $\{1, \ldots, k\}$, *i.e.*, $\sum_i \Omega_i = 1$. The generator polynomial is defined by this probability distribution is given by $\Omega(x) = \sum_i \Omega_i x^i$. Define a random variable X_Ω that selects randomly a subset of $\{1, \ldots, k\}$ such that the probability of selecting a subset of size l is $\frac{\Omega_l}{\binom{k}{l}}$. To generate each output (encoded) packet, one generates a statistically independent instance of X_Ω. The packets whose indices are in the selected subset are then added (XORed) and the information of the indices used to generate this packet are appended to the packet. Since the random encoding is performed in a real-time fashion, it is necessary to append this information to the output packet for the decoder to perform decoding. Figure 3.2 presents an illustration of the encoding process where a sequence of encoded packets is generated from the input packets and the details of the packets are appended in the small shaded portion of each output packet.

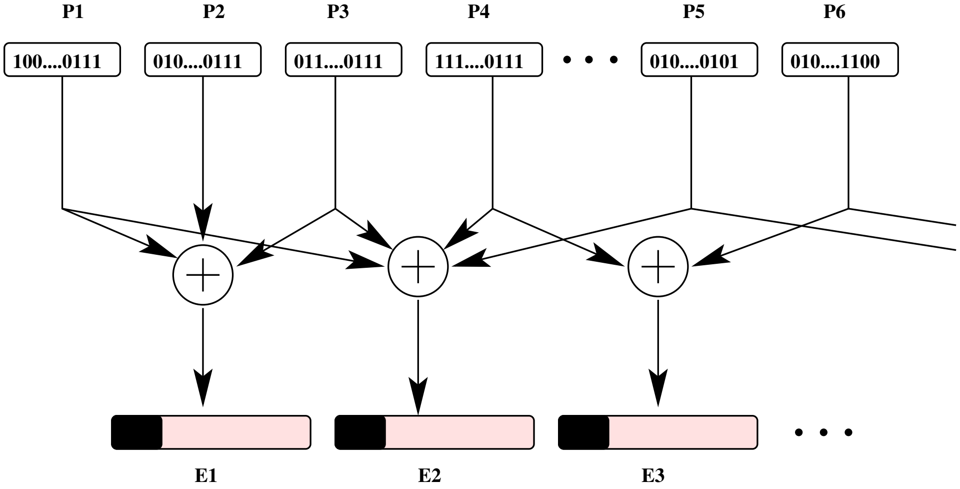

Figure 3.2: An illustration of the encoding process of rateless codes.

Encoded packets are injected into the channel as and when they are generated. At the receiver, as and when a packet arrives, the decoder constructs a Tanner graph with k bit nodes by adding a check node. The edges incident on the newly added check node connect this check node to exactly those bit nodes that were used to generate the presently received (encoded) packet. The decoder waits for a sufficient number of packets for the message-passing procedure outlined in Section 3.0.4 to complete successfully. It can be shown that the required number of received encoded packets for successful completion of the decoding process is about $k\gamma_{k,\Omega}$, where $\gamma_{k,\Omega} \gtrsim 1$. This constant $\gamma_{k,\Omega}$ is called the rateless overhead. The generator polynomial can be optimized such that $\gamma_{k,\Omega} \longrightarrow 1$ as $k \to \infty$. Another overhead that is incurred is that of appending the packet information caused by the online generation of packets at the encode. It can be shown that the expected number of packets used for generating an encoded packet is $\sum_i i\Omega_i = \Omega'(1)$ packets. Since representing the index of a packet will require $\lceil \log_2 k \rceil$ bits, the expected number of bits in the appended information overhead is about $\Omega'(1)\lceil \log_2 k \rceil$. This overhead is caused because of appending the packet information onto the packet itself. This overhead will be negligible of the packet size L satisfies $L \gg \Omega'(1)\lceil \log_2 k \rceil$.

3.0.4 Decoding Graph-Based Codes over BEC

The most common decoding algorithm for codes represented by Tanner graphs is the message-passing (MP) algorithm. As the name suggests, it involves passing messages to-and-fro over edges of the Tanner graph in an iterative fashion. In general memoryless binary-input output-symmetric (MBIOS) channels [101], the messages that are sent over the edges are log-likelihood ratio (LLR) values. Consider a binary random variable X transmitted over an MBIOS channel with channel output Y whose instance corresponding to the transmitted instance of X is r. Then, the LLR of X is defined to be

$$\mathrm{LLR}(X) \triangleq \log \frac{P[Y = r|X = 1]}{P[Y = r|X = 0]}. \tag{3.2}$$

Over the BEC, it can be seen that the values Y can take are $0, 1$, and ? and that LLR can take only values $\pm\infty$ and 0. An LLR of ∞ means that the transmitted bit was a one and an LLR of $-\infty$ affirms that the transmitted bit was a zero. However, a zero LLR implies erasure and that the bit could equally likely be either a zero or a one. The MP algorithm for the Tanner graph $G(V \cup C, E)$ for the BEC proceeds as follows.

1. Each bit node computes the LLR using the received channel outputs.

2. Each bit node forwards on each of its incident edges its LLR values.

3. If the magnitude of all messages received by a check node is ∞ excepting that of one, it sends $(-1)^{t-1}\infty$ over the edge on which an LLR of zero was received. Here, t is the number of messages that equal ∞.

4. If a node whose LLR was zero receives a non-zero message, it updates its LLR to the message received.

5. Repeat 2, 3, and 4 until no check node satisfying the condition in 3 exists.

6. If the LLR of each bit node is non-zero, declare the bits using the map $\infty \mapsto 1$ and $-\infty \mapsto 0$. Otherwise, declare *decoder failure.*

Figure 3.3 presents an illustration of the aforementioned decoding procedure for a code of length 5 and rate $\frac{2}{5}$. The arrows represent the non-zero messages sent over the edges that

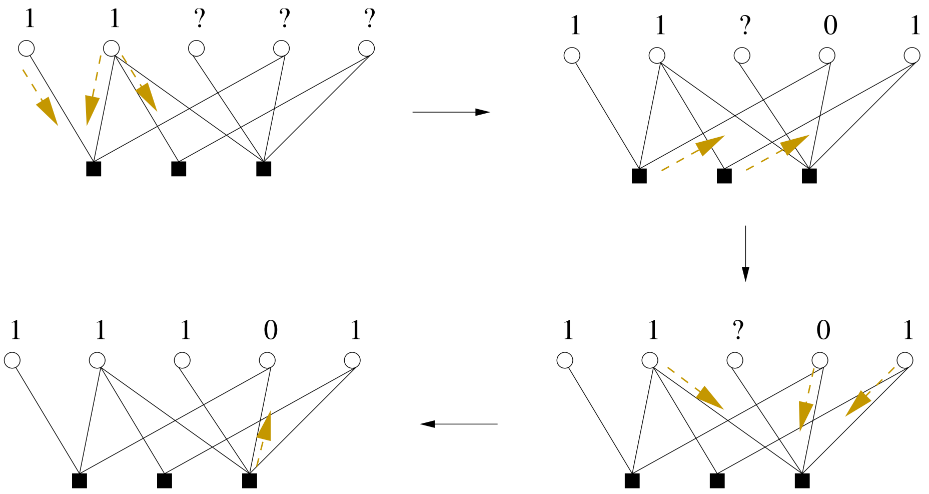

Figure 3.3: An illustration of message passing decoder over the BEC.

result in successful decoding of at least a bit. It can be seen that after two rounds of message passing between the bit nodes and check nodes and vice versa, the entire codeword will be recovered.

In the case that the decoding does not succeed, one can show that the subset of nodes that remain unidentified is a *stopping set.* A stopping set $S \subset V$ is a subset of nodes with the property that every node c in the neighborhood $N_G(S)$ has at least two neighbors in S, i.e., $|N_G(c) \cap S| > 1$. For codes of short length, the presence of stopping sets has an adverse effect on the performance of the code. Hence, the design of codes that do not have small stopping sets is key for achieving good performance over the BEC. Also, in many cases, it may be possible to successfully decode the received word using maximum likelihood (ML) decoder even if the MP decoder fails. To improve the performance of the MP decoder under such circumstances, several improved MP decoders have been proposed in literature. In specific, [93] presents a simple guessing alternative to proceed after MP decoder has halted at a stopping set. The algorithm proposed by P. Nik *et al.* performs the standard MP algorithm and then guesses a pre-determined number of bits from the bits of the stopping set. This procedure is described in detail in the following subsection.

3.0.4.1 Improved Message Passing Algorithm

The improved message-passing algorithm (Algorithm A) in [93] uses random bit-guessing after the failure of message passing to decode the received codeword as a means to improve the performance of the code. Algorithm A is the same as MP decoding when the maximal stopping set contained in the set of erasures is, trivially, the empty set. When that is not the case, the MP decoder terminates at the maximal stopping set, whereas decoding continues therefrom when Algorithm A chooses an unknown variable node at random and guesses its value randomly and equiprobably from $\mathbb{F}_2\,(=\{0,1\})$. It then restarts the message passing with the newly guessed value until it either completes or stagnates at a smaller stopping set or reaches a contradiction. If it stagnates with no contradictions, a new bit node is picked and its value is then guessed and the message passing is recommenced. If the decoder finds a check node that cannot be satisfied (i.e., the check node is contradicted), then the value of the guessed bit is flipped and the procedure of message passing is restarted. This whole process of guessing and message passing is continued until a predetermined g bits are guessed and no check node is contradicted or if the decoding is completed. A decoding failure is declared after the guess of g bits if further message-passing routines result in no contradiction and if the value of certain bits still remain unknown. After g guesses, if a contradiction is attained, then a different pattern for the chosen g bits is guessed and message passing is restarted anew.

PART I

Practical Coding Schemes with LDPC Codes

CHAPTER IV

IMPROVED DECODER FOR LDPC CODES OVER THE BEC

Ever since the invention of turbo codes, several important aspects of LDPC codes that were previously ignored have been studied. These codes have been shown to perform very well over a wide range of channels including the binary erasure channel, the binary symmetric channel (BSC), and the additive white Gaussian noise channel (AWGN). In specific, the existence of a family of LDPC codes that approach the capacity of the BEC has been affirmed [84, 110]. However, the standard message-passing (MP) decoder that offers near-optimal performance for large codelengths performs poorly for codes of short lengths. This phenomenon is attributed to the existence of certain subsets of variable nodes in the Tanner graph of the LDPC code that are widely known as *stopping sets*. Amongst the many approaches that the coding community has pursued to mitigate the effect of stopping sets in short-length codes, two important directions that have been studied are: 1. the design of short-length LDPC codes with suitable stopping set distribution, and 2. bit-guessing schemes to extend the MP decoder. In this work, we focus on the latter approach of designing decoders that extend the MP decoder with the aid of bit-guessing. We first notice that a clear understanding of the bit-guessing approach is not possible without a structural understanding of the stopping sets. Our first part therefore focusses on the analysis and structural characterization of stopping sets that are solved by guessing a fixed number of bits. Next, we introduce some key observations regarding size of stopping sets that are motivated partly by extensive simulations and partly by theoretical arguments. Finally, we introduce a novel improved decoder that is based on an intelligent guessing criterion that exploits these observations.

The chapter is organized thus. Section 4.1 presents the formal statement of the problem and the issues that are addressed in this work. Section 4.2 outlines the MP decoding procedure and then presents the characterization and properties of stopping sets. The first half of Section 4.3 motivates some key observations regarding the distribution of the stopping sets. The latter half of this section explores the design of an improved decoding algorithm that exploits the characterization in conjunction with these observations. Finally, Section 4.4 presents the results of simulations for the proposed improved decoding algorithm.

4.1 Problem Statement

The analysis of stopping sets is key in designing practical decoders for short-length LDPC codes that bridge the gap between the performance of the standard message passing decoder and that of the optimal *maximum-likelihood* (ML) decoder. The reason for the sub-optimal performance of MP decoder is the existence of several ML-decodable stopping sets. Recently, several practical decoders that employ bit-guessing as a means of extending the MP decoders have been proposed [92, 93]. However, several important questions regarding these decoders remain unanswered. Our work in this area attempts to answer the following questions.

1. What are the structural properties of a stopping set that is ML-decodable, but is decodable only with the guess of, say, g bits when MP decoder is employed?

2. Given that MP decoder has halted at a stopping set after the decoding process and that guessing at most g bits is allowable, how do we intelligently select the bits that must be guessed? Also, what is the additional computational cost that we incur if an efficient selection criterion is identified?

These questions aim at understanding several issues regarding the stopping sets including the minimum number of bits that need to be guessed for successfully decoding an ML-decodable stopping set. Identifying this minimum number will help determine the bits that need to be guessed and hence make the bit-guessing computationally efficient.

4.2 Improved Decoding and Stopping Sets

In the first half of this work, we will be concerned with the former of the two improved decoding algorithms (Algorithm A) proposed in [93]. To improve the performance of the decoder, Algorithm A employs random bit-guessing after the message passing fails to decode the received codeword. Algorithm A is the same as MP decoding when the maximal stopping set contained in the set of erasures is, trivially, the empty set. When that is not the case, the MP decoder terminates at the maximal stopping set. However, decoding proceeds in Algorithm A with the random selection of an unknown variable node whose value is guessed equiprobably from $\mathbb{F}_2 = \{0, 1\}$. It then restarts the message passing with the new guessed value until it either completes or stagnates at a smaller stopping set or reaches a contradiction. If it stagnates with no contradictions, then a new bit node is picked and its value is then guessed and the message passing is recommenced. If the decoder finds a check node that cannot be satisfied (i.e., the check node is contradicted), then the value of the guessed bit is flipped and the procedure of message passing is restarted. This whole process of guessing and message passing is continued until a pre-determined g bits are guessed and no check node is contradicted or if the decoding is completed. We declare a decoding failure if we guess g bits and reach no contradiction but have bits whose values are yet unknown. After g guesses, if we reach a contradiction, then we try a different pattern for the chosen g bits and start the message passing afresh. Here, we are primarily interested in characterizing the stopping sets based the minimum number of guesses that has to be made to complete the decoding process.

4.2.1 Free Number and Free Set of a Stopping Set

The expected complexity of Algorithm A can be seen to grow exponentially with the maximum number of guesses ($O(2^g n)$) when the length of the code n is kept fixed. Therefore, it is mandatory that we keep g as small as possible. In this section, we investigate the problem of characterizing stopping sets that can be completely decoded with the knowledge of g bits. Without loss of generality, we may also assume, henceforth, that by a stopping set we mean an ML-decodable stopping set.

It can be easily shown that every stopping set S has a minimum number $\mathcal{F}_S > 0$ and a set $B_S \subset S$ of size $\mathcal{F}_S$ such that the knowledge of the values of the bit nodes in B_S is sufficient to complete the process of decoding via further message-passing routines. We shall term this minimum number of bits as the '***free number***' ($\mathcal{F}_S$) of S and any set of cardinality $\mathcal{F}_S$ whose knowledge completes decoding to be a '***free set***' (B_S). Alternatively, we say that a stopping set S is '$\boldsymbol{g}$**-solvable**' if $\mathcal{F}_S \leq g$. Also, if for a parity equation of a check node c involving variables $u_1, \ldots, u_n$, all but one variable u_j are known, we can find the value of u_j from this check node. When this is the case, we say that the check node c '***frees***' variable u_j (or equivalently, we also say u_j is '***freed***' by c).

As an example, let in the graph G presented in Figure 4.1, $S = \{2, 3, 4, 5\}$ be the set of indices of erased bits. Also, let the bits at positions $1, 6$ be $b_1 = b_6 = 0$. It can be seen that S is an ML-decodable stopping set of G.

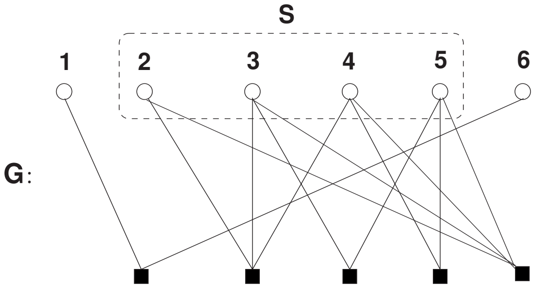

Figure 4.1: A simple example to illustrate the concept of free set and free number.

Simple calculation reveals that guessing correctly the bit at position 3 or 4 or 5 is sufficient to decode the stopping set S. Therefore, $\mathcal{F}_S = 1$ and $B_S = \{3\}$ or $\{4\}$ or $\{5\}$. This can be mathematically verified in the following way.

$$H\mathbf{c}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{bmatrix} = \mathbf{0}_{5\times 1} \Rightarrow \left[\begin{array}{c|cc|c} 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 0 \\ \cline{1-3} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array}\right] \begin{bmatrix} b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} = \mathbf{0}_{4\times 1}. \quad (4.1)$$

Since the last two equations are of weight 2 and since they share a variable b_5, guessing any one of b_3, b_4, or b_5 renders the vector MP-decodable. From this example, we note a stopping set can possibly have multiple free sets. In fact, it can be shown that for a stopping set with free number g has at least $g + 1$ free sets.

4.2.2 Matrix Characterization of g-solvable Stopping Sets

A stopping set S is successfully decoded by the ML decoder if and only if the '**residual parity-check matrix**' H_S (submatrix of H that contains only those columns which correspond to bits of S) has full-rank. Here, we prove an analogous statement for characterizing g-solvability. In this section, we present the result that MP decoding completes after g guesses if and only if there exist permutations of rows and columns that endow H_S an upper diagonal structure. The following theorem mathematically characterizes the criterion for g-solvability [121].

Theorem 4.2.1. (*Matrix characterization theorem for g-solvability*) *Let E be the set of erased bits and $S \subseteq E$ be a stopping set in an LDPC code. Then, $\mathcal{F}_S \leq g$ if and only if $\exists\, t \in \mathbb{N},\ s_1, \ldots, s_t \in \mathbb{N}$ s.t. $s_1 + \ldots + s_t = |S| - g$ and*

$$H_S \sim \left(\begin{array}{c|c|c|c|c|c|c|c}
\multicolumn{7}{c|}{X_t} & \\
\cline{1-7}
I_{s_t} & \multicolumn{6}{c|}{X_{t-1}} & \\
\cline{1-7}
\mathbf{0} & I_{s_{t-1}} & \multicolumn{5}{c|}{X_{t-2}} & \\
\cline{1-7}
\mathbf{0} & \mathbf{0} & I_{s_{t-2}} & \multicolumn{4}{l|}{X_{t-3}} & X_0 \\
\cline{1-7}
\multicolumn{7}{c|}{\vdots} & \\
\cline{1-7}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & I_{s_2} & X_1 & \\
\cline{1-7}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & I_{s_1} &
\end{array}\right), \tag{4.2}$$

where each submatrix represented by X_i, $i = 0, 1, 2, \ldots, t$ is an arbitrary binary matrix of the appropriate size.

The proof of the above result can be found in Section A.1 of Appendix A. The above theorem places a more stringent structural condition than the full-rank criterion for successful decodability for ML decoding. This can be attributed to the fact that the MP decoder attempts to solve the system of check equations without performing addition of check equations that the ML decoder potentially performs. From the theorem, we observe that only those systems H_S having a unique solution and requiring only row permutation and column permutation operations to obtain an upper triangular structure will be solvable by the MP decoder. The following result characterizes this structural requirement precisely.

Corollary 1. *A stopping set S is g-solvable if and only if*

$$H_S \sim \left[\begin{array}{c|c} \multicolumn{2}{c}{X^{(1)}_{(n-nR-|S|+g)\times|S|}} \\ \hline U_{(|S|-g)\times(|S|-g)} & X^{(2)}_{(|S|-g)\times g} \end{array} \right], \tag{4.3}$$

where U is an upper triangular matrix with an all-one (principal) diagonal, and the matrices denoted by $X^{(i)}$, $i = 1, 2$ are almost "don't care" binary matrices (i.e., have no algebraic constraints on their structure except that they must have at least one non-zero entry in each row). Additionally, if g is the smallest such number, then the set of bits indexed by the last g columns is a free set.

Revisiting the example in the previous subsection, we see that (4.1) agrees with the structure prescribed by Theorem 4.2.1 and that the stopping set S is indeed 1-solvable and that $\{5\}$ is a free set of S. Theorem 4.2.1 does not guarantee explicit uniqueness of the parameters t and $\{s_i\}_{i=1}^{t}$. However, for each free set, we can make the characterization canonical by picking the parameters with the property described in the proof. With the same notations as in (4.2), one can show that in a canonical representation, the submatrix of X_j corresponding to columns of I_{s_j} has no all-zero rows whenever $j \in \{1, \ldots, t\}$. A similar constraint can be shown to exist for X_t. In such a representation, one can note that the parameter t represents the number of iterations taken to completely recover all the bits of the stopping set when the bits of the corresponding free set are guessed. The final result in this subsection uses the representation of Theorem 4.2.1 to derive a lower bound on the number of iterations required for complete decoding of stopping sets.

Lemma 4.2.1. *Consider an LDPC ensemble whose edge distribution polynomial is $\lambda(x) = \sum_{i=2}^{d_{v_{\max}}} \lambda_i x^{i-1}$ with $d_{v_{\max}} > 1$. Let S be a stopping set of a code of length n from this ensemble with $\mathcal{F}_S = g$. Then the number of iterations t needed to complete the decoding when the g bits of a free set of S are guessed is at least*

$$t = \begin{cases} \lceil \frac{\log(|S|-g)+\log(d_{v_{\max}}-1)-\log(d_{v_{\max}})}{\log(d_{v_{\max}}-1)} \rceil & \textit{if } d_{v_{\max}} > 2 \\ \lceil \frac{|S|-g}{g} \rceil & \textit{if } d_{v_{\max}} = 2 \end{cases}. \tag{4.4}$$

Proof. Let S_0 be a free set of S. Then, $|S_0| = g$. From Theorem 4.2.1, we see that there exist permutations of bit and check nodes that endow H_S the canonical structure described above. As in the proof of Theorem 4.2.1, let us denote the set of bits freed in the i^{th} iteration as S_i and denote $|S_i| = s_i$. Then by definition, for every bit node in S_j with $j \in \{1, \ldots, t\}$, there is a check node that expresses the value of the bit as a linear combination of the values of bits in the set $\cup_{k=0}^{j-1} S_k$. There are in total at most $gd_{v_{\max}}$ equations involving bits of S_0. Hence, $|S_1| \leq gd_{v_{\max}}$. For each bit in S_k for some $k > 1$, the aforementioned linear combination must involve at least one bit from the set S_{k-1}, else the value of the bit will be found at the end of $(k-1)^{\text{th}}$ iteration or earlier, thereby contradicting its placement in S_k. After the k^{th} iteration, the number of equations that involve bits of S_{k-1} is bounded above by $s_{k-1}(d_{v_{\max}} - 1)$. Hence $s_k \leq s_{k-1}(d_{v_{\max}} - 1)$. Thus, one sees that

$$|S| - g = \sum_{i=1}^{t} s_i \leq gd_{v_{\max}} \sum_{i=0}^{t-1} (d_{v_{\max}} - 1)^i. \tag{4.5}$$

Estimating t from the above equation then completes the proof. ❑

4.2.3 Bounds on the Free Number of Stopping Sets

In this section, we derive bounds for the free number of a stopping set of an LDPC Code.

Lemma 4.2.2. *(Lower bound for $\mathcal{F}_S$) Let S be a stopping set. Then,*

$$(1 + \mathcal{F}_S) \geq \min\{\partial_{G[S \cup N_G(S)]}(c) : c \in N_G(S)\}. \tag{4.6}$$

Proof. If we are to free the check node $c \in N_G(S)$, then we must know at least all but one of the neighbors of c that are yet unknown. But the number of unknown neighbors of c is given by $\partial_{G[S \cup N_G(S)]}(c)$. This guarantees the freedom of the nodes in $N_G(\{c\})$ but not necessarily all bits of S. Therefore, to free an unknown bit of S we must necessarily know/guess at least $\min\{\partial_{G[S \cup N_G(S)]}(c) : c \in N_G(S)\} - 1$ bits which will free an unknown variable via a check node of minimum degree in $G[S \cup N_G(S)]$. Since further iterations of message passing may yield a smaller stopping set S' such that $\emptyset \subsetneq S' \subsetneq S$, the claim follows. ❑

Lemma 4.2.3. *(Upper bound for $\mathcal{F}_S$) Let S be a stopping set from a code of length n and rate R. Let Z be the number of check nodes c such that $N_G(\{c\}) \cap S = \emptyset$. Let $0 < w_1 \leq w_2 \leq w_3 \leq \ldots \leq w_{[n(1-R)-Z]}$ be an ordering of the degrees of check nodes of $N_G(S)$. Let $l = \max\{i \in \mathbb{N} : \sum_{j=1}^{i} w_j < |S|\}$. Then, $\mathcal{F}_S \leq |S| - l - 1$.*

Proof. First, we can arrange the rows of H_S such that the degree of the check nodes (equivalently, the row-sum) decreases as the row-index increases. We can then group all all-zero rows of H_S at the top by a simple cyclic permutation. To get an upper bound on the free number of S, we must ensure that there is a permutation of rows and a permutation for columns that will guarantee an all-zero triangular area below a diagonal of H_S (Corollary 1). Since the last row of H_S has weight w_1, we can find a permutation of columns of H_S that clusters all the ones to the last w_1 columns on the right bottom corner of H_S and therefore we are guaranteed to have a block of $|S| - w_1$ zeros at the left bottom corner. By a similar argument, we can find a permutation of the first $|S| - w_2$ columns that groups all the ones of the penultimate row to the right. After such a permutation, we are guaranteed that at least the first $|S| - w_1 - w_2$ columns are zeros at this row. Proceeding similarly, we see that we can order exactly l rows in this fashion. Since each $w_i \geq 1$, $i = 1, \ldots, [n(1-R) - Z]$ we see that there is a triangular block of at least $\frac{l(l+1)}{2}$ zeros cornered at the left bottom of the matrix. Finally, a simple permutation of the columns then renders H_S the structure defined in Theorem 4.2.1, which then guarantees that S is $(|S| - l - 1)$-solvable. ❑

Lemma 4.2.4. *(Upper bound for $\mathcal{F}_S$) Let S be a stopping set from a code of length n and rate R. Let $0 < w_1 \leq w_2 \leq w_3 \leq \ldots \leq w_{|S|}$ be an ordering of the degrees of variable nodes of S. Let $l' = \max\{i \in \mathbb{N} : \sum_{j=1}^{i} w_j < n(1-R)\}$. Then, $\mathcal{F}_S \leq |S| - l' - 1$.*

Proof. The proof is on similar lines as that of Lemma 4.2.3. ❑

To illustrate the idea in the proof of Lemma 4.2.3, consider H_S as given below in Figure 4.2 where $w_1 = w_2 = w_3 = 2$, $w_4 = 3$, $w_5 = 4$, and $w_6 = 6$. The proof of the lemma can be broken down exactly into three steps that are outlined in the Figure 4.2. Firstly, the rows are arranged in decreasing order of weights, which can be done in multiple ways. Then, the columns are clustered one by one from the last to the first row so that we get a triangular area of zeros at the bottom left corner. Then a re-arrangement of the columns is done to ensure that the structure is as prescribed in Theorem 4.2.1. For this example, we see from Lemma 4.2.3 that $l = 2$ and $\mathcal{F}_S \leq 6 - 2 - 1 = 3$. Actually, we find that the free number of this stopping set is 2 and that $\{3, 6\}$ is a free set. The idea for the proof of Lemma 4.2.4 is identical to that discussed above but with the roles of columns and rows interchanged.

$$\begin{tikzcd}
H_S = \begin{array}{c|cccccc|c} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \\ & 1 & 1 & 1 & 1 & 1 & 1 & \mathbf{1} \\ & 1 & 0 & 0 & 1 & 0 & 0 & \mathbf{2} \\ & 0 & 1 & 1 & 0 & 0 & 0 & \mathbf{3} \\ & 0 & 1 & 0 & 1 & 1 & 0 & \mathbf{4} \\ & 0 & 0 & 0 & 1 & 0 & 1 & \mathbf{5} \\ & 1 & 1 & 1 & 1 & 0 & 0 & \mathbf{6} \end{array} \arrow[r, "\text{Order rows}"'] & \begin{array}{c|cccccc|c} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \\ & 1 & 1 & 1 & 1 & 1 & 1 & \mathbf{1} \\ & 1 & 1 & 1 & 1 & 0 & 0 & \mathbf{6} \\ & 0 & 1 & 0 & 1 & 1 & 0 & \mathbf{4} \\ & 1 & 0 & 0 & 1 & 0 & 0 & \mathbf{2} \\ & 0 & 1 & 1 & 0 & 0 & 0 & \mathbf{3} \\ & 0 & 0 & 0 & 1 & 0 & 1 & \mathbf{5} \end{array} \arrow[d, "\text{Cluster ones}"] \\
\begin{array}{c|cccccc|c} & \mathbf{5} & \mathbf{1} & \mathbf{2} & \mathbf{4} & \mathbf{3} & \mathbf{6} & \\ & 1 & 1 & 1 & 1 & 1 & 1 & \mathbf{1} \\ & 0 & 1 & 1 & 1 & 1 & 0 & \mathbf{6} \\ & 1 & 0 & 1 & 1 & 0 & 0 & \mathbf{4} \\ & 0 & 1 & 0 & 1 & 0 & 0 & \mathbf{2} \\ & 0 & 0 & 1 & 0 & 1 & 0 & \mathbf{3} \\ & 0 & 0 & 0 & 1 & 0 & 1 & \mathbf{5} \end{array} & \begin{array}{c|cccccc|c} & \mathbf{5} & \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{6} & \\ & 1 & 1 & 1 & 1 & 1 & 1 & \mathbf{1} \\ & 0 & 1 & 1 & 1 & 1 & 0 & \mathbf{6} \\ & 1 & 0 & 1 & 0 & 1 & 0 & \mathbf{4} \\ & 0 & 1 & 0 & 0 & 1 & 0 & \mathbf{2} \\ & 0 & 0 & 1 & 1 & 0 & 0 & \mathbf{3} \\ & 0 & 0 & 0 & 0 & 1 & 1 & \mathbf{5} \end{array} \arrow[l, "\text{Re-arrange columns}"]
\end{tikzcd}$$

Figure 4.2: A simple example to illustrate the proof of the upper bound on free number.

We would like to mention that the upper bounds derived here are superior to that presented in [93], since the latter only uses the information of check nodes with degree 2. Hence, in stopping sets where bivalent check node are absent, the upper bound of [93] equals the size of the stopping set itself. Our bound, on the other hand, can be clearly seen to be superior in such cases. Another important feature of these bounds is that they can be easily computed for any particular stopping set.

Our bounds seem reasonably tight for codes of small length ($n \approx 100$). As length increases, we note by simulations that the discrepancy between the actual free number and the bounds seems to increase. The discrepancy between the former upper bound and the actual value of the free number is due to the fact that the proof of Lemma 4.2.3 assumes that the position of ones in any two rows from the set of l rows of least weight are pairwise disjoint, which may not be the case in general. In fact, the proof assumes the worst-case scenario. Similar is the case with the proof of the other upper bound.

4.3 Improved Decoding Algorithm

In this section, we concentrate on the design of an improved decoder that takes into consideration the results of the previous section. We begin this section with some observations that concern the size and structure of stopping sets. These observations in conjunction with the characterization of the structure of stopping sets are exploited in the design of our improved decoder discussed in the later part of this section.

4.3.1 Background Facts and Observations

The first observation that we would like to highlight is as follows.

Observation 4.3.1. *Consider a code C of length n. Let this code be used over a BEC of erasure probability ϵ. Let S_ϵ be the random variable representing the bits of the stopping set after decoding the received word using the MP decoder. Then*

$$g(\epsilon) = E[|S_\epsilon| \mid |S_\epsilon| > 0] \tag{4.7}$$

is a non-decreasing function of $\epsilon \in (0, 1)$.

A detailed proof of Observation 4.3.1 can be found in Section A.2 of Appendix A. Intuitively, Observation 4.3.1 reveals that, on the average, the probability of incomplete decoding and the sizes of the stopping sets when the decoder cannot decode the received word both decrease as the channel erasure probability is decreased. However, it is possible to get qualitative information regarding the distribution of the stopping sets from $\frac{dg}{d\epsilon}$, the derivative of the function g defined in Observation 4.3.1. As an illustration, Figure 4.3 gives the variation of the expected size of stopping sets given that the MP decoder does not complete decoding, as a function of channel erasure probabilities for codes of length 1000 from ensembles C1, C2, and C3. At an erasure probability of 0.37, C3 exhibits lesser number of erasures after decoding in comparison to ensemble C1 and C2. Thus, we can conclude that, unlike ensembles C1 and C2, there are not many "small" stopping sets in the codes from ensemble C3. Another important observation of stopping sets obtained after decoding received codewords from LDPC codes using the MP decoder is the following.

Observation 4.3.2. *In a typical stopping set obtained when decoding received codewords from LDPC codes at (coded) bit-error rates of about 10^{-1} or smaller, the set of bivalent check nodes ($C_2(N_G(S))$) form a considerable fraction of the set of unsatisfied check nodes ($N_G(S)$).*

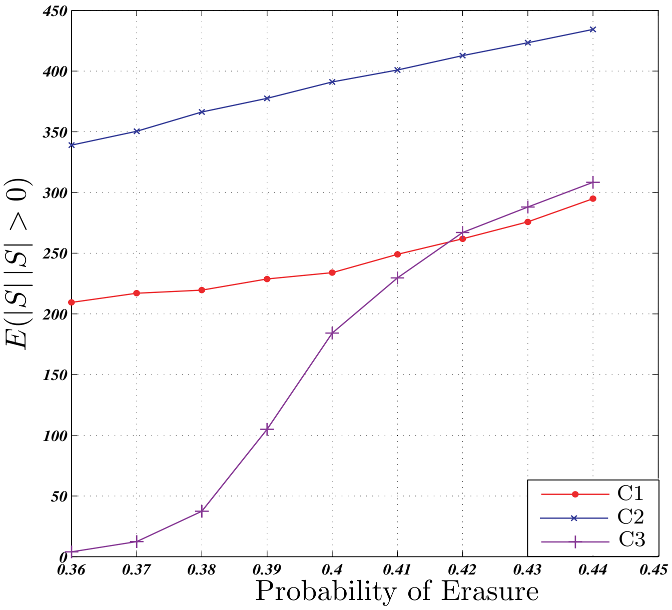

Figure 4.3: The variation of g (defined in Observation 4.3.1) with the channel parameter for various ensembles.

Figure 4.4 illustrates the distribution of the fraction of bivalent check nodes for a code of length 1000 from ensemble C1 when the channel erasure probability $\epsilon = 0.42$. At $\epsilon = 0.39$, where the code exhibits a bit-error rate of about 10^{-2}, the expected fraction of bivalent check nodes increases to about 0.63 (with $\sigma^2 = 0.003$). Table 4.1 provides the ratio of bivalent check nodes to the number of unsatisfied check nodes for various ensembles simulated in this work from Table 3.2 of Chapter 3.

Table 4.1: Mean and variance of the ratio of bivalent check nodes to unsatisfied check nodes for various ensembles.

Ensemble	(μ, σ^2) at BER=10^{-1}	(μ, σ^2) at BER=10^{-2}
C1	$(0.5562, 3.88 \times 10^{-3})$	$(0.6260, 2.78 \times 10^{-3})$
C2	$(0.2622, 2.69 \times 10^{-3})$	$(0.2990, 2.36 \times 10^{-3})$
C3	$(0.4949, 4.74 \times 10^{-3})$	$(0.5419, 4.04 \times 10^{-3})$
C4	$(0.1985, 6.47 \times 10^{-3})$	$(0.2161, 4.52 \times 10^{-3})$

A qualitative explanation of the above observation, at least for regular ensembles, can be given using the combinatorial problem of estimating the number of binary matrices with a given row- and a column-sum vector that is explored in [75, 10, 76]. In [10], it was shown that the number $M(\underline{r}, \underline{c})$ of $m \times n$ binary matrices with row-sum vector $\underline{r} = (r_1, r_2, \ldots, r_m)$

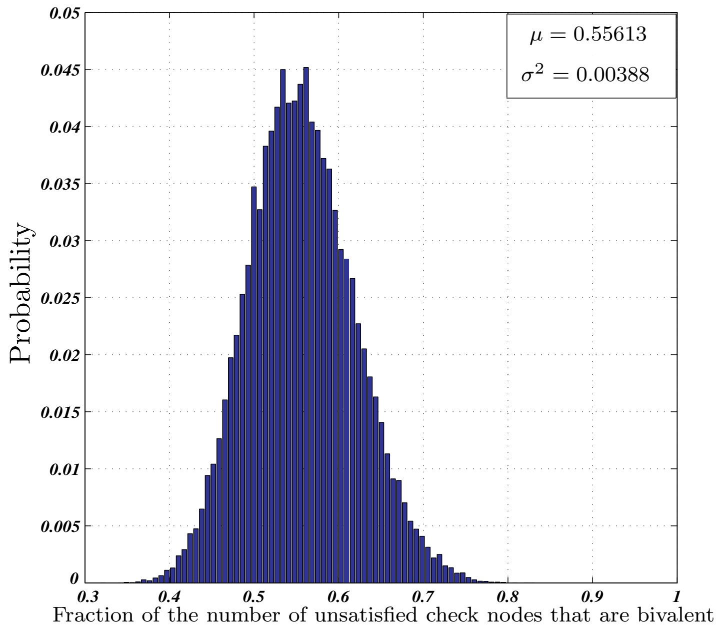

Figure 4.4: Discretized probability distribution of the fraction of check nodes that are bivalent for a code of length 1000 from ensemble C1 at a bit-error rate of 10^{-1}.

and column-sum vector $\underline{c} = (c_1, c_2, \ldots, c_n)$ is given by

$$M(\underline{r}, \underline{c}) \approx \frac{(\sum_{k=1}^{m} r_k)!}{\prod_{i=1}^{m} r_i! \prod_{j=1}^{n} c_j!} e^{-\Delta} \tag{4.8}$$

where

$$\Delta = 2\frac{(\sum_{i=1}^{m} \binom{r_i}{2})(\sum_{j=1}^{n} \binom{c_j}{2})}{(\sum_{i=1}^{m} r_i)(\sum_{j=1}^{n} c_j)}. \tag{4.9}$$

Consider a regular ensemble of LDPC codes of length n and rate R whose degree distributions are such that parity-check matrices of codes from this ensemble have a row-sum vector[1] $\underline{r} = d_c \mathbb{1}_{(1,\overline{R}n)}$ and a column-sum vector $\underline{c} = d_v \mathbb{1}_{(1,n)}$, where $d_c \triangleq \frac{d_v}{\overline{R}}$. Consider the scenario when a code from this ensemble is selected. Suppose that after decoding a codeword from this code, one arrives at a stopping set of size αn. By regularity, one can assume that the stopping set corresponds to the first αn vertices. Then, one sees that

$$\begin{aligned} Pr[\Re(H_S) = \underline{r}' \,|\, |S| = \alpha n] &\propto M(\underline{r}', d_v \mathbb{1}_{(1,\alpha n)}) M(\underline{r} - \underline{r}', d_v \mathbb{1}_{(1,\overline{\alpha} n)}) && (4.10) \\ &\approx \frac{[\alpha n d_v]! e^{-\Delta_1}}{[d_v!]^{\alpha n} \prod_i r_i'!} \frac{[\overline{\alpha} n d_v]! e^{-\Delta_2}}{[d_v!]^{\overline{\alpha} n} \prod_j (d_c - r_j')!} && (4.11) \\ &\propto \frac{e^{-\Delta_1 - \Delta_2}}{\prod_i (r_i')!(d_c - r_i')!} && (4.12) \end{aligned}$$

[1] Generally, the row-sum and column-sum vectors are unique only up to a permutation.

$$\propto \prod_i \binom{d_c}{r'_i} = \prod_{i=0,2,\ldots,d_c} \left[\binom{d_c}{i}\right]^{\overline{R}n\tilde{\lambda}^{(i)}_{r'}}, \tag{4.13}$$

where (a) the column vector $\underline{r}'$ is subject to the constraints that $r'_i \in \{0, 2, \ldots, d_c\}$ for each $i = 1, 2, \ldots, \overline{R}n$ and $\sum_i r'_i = \alpha n d_v$, and (b) $\tilde{\lambda}^{(i)}_{r'}$ is the fraction of rows in H_S that have a row-sum of i. Here, we would like to reiterate that our definition of an LDPC ensemble described by a pair of degree distribution (λ, ρ) differs from that of [101] in that we consider only the simple graphs from the ensemble constructed by the socket technique. This is visible from (4.11) since we use the factor $e^{-\Delta}$ that represents the probability that a random code from the standard LDPC ensemble has a Tanner graph that is simple, i.e., no parallel edges between nodes. Also, the approximation simplifying (4.12) to (4.13) is not strictly valid but is acceptable for codes of moderate lengths (of the order of 10^3), since the following bounds hold.

$$\max\left\{0, \frac{\alpha d_c - 1}{2}\right\} \leq \quad \Delta_1 \quad \leq \frac{(d_v - 1)(d_c - 1)}{2}. \tag{4.14}$$

$$\max\left\{0, \frac{(1-\alpha) d_c - 1}{2}\right\} \leq \quad \Delta_2 \quad \leq \frac{(d_v - 1)(d_c - 1)}{2}. \tag{4.15}$$

The approximation simplifies (4.12) to a mathematically tractable term in (4.13) from which qualitative inferences can be drawn. To find the most likely row-sum vector given the size of the stopping set, one has to maximize the term in (4.13) over all valid row-sums, yielding following optimization problem.

Maximize $\sum_{i=1}^{d_c} \tilde{\lambda}^{(i)}_{r'} \log \binom{d_c}{i}$ *subject to*

$$\begin{array}{llll} A. & \{\tilde{\lambda}^{(i)}_{r'}\}_{i=0}^{d_c} \in [0,1]^{d_c} & B. & \tilde{\lambda}^{(1)}_{r'} = 0 \\ C. & \sum_i \tilde{\lambda}^{(i)}_{r'} = 1 & D. & \sum_i i\tilde{\lambda}^{(i)}_{r'} = \alpha d_c \end{array} . \tag{4.16}$$

The node degree distribution polynomial $\tilde{\lambda}_{r'*}(x) = \sum_{i=1}^{d_c} \tilde{\lambda}^{(i)}_{r'*} x^i$ that is the solution to the above maximization problem can be shown to be

$$\tilde{\lambda}_{r'*}(x) = \begin{cases} (1 - \frac{\kappa}{2}) + \frac{\kappa}{2} x^2 & \text{if } (2|\alpha n d_v) \,\&\, (\kappa < 2) \\ x^{\lfloor \kappa \rfloor}[\kappa - \lceil \kappa - 1 \rceil + (\lceil \kappa \rceil - \kappa) x] & \text{if } \kappa \geq 2 \\ (\frac{2 - \kappa + \kappa x^2}{2}) + \frac{1 - 3x^2 + 2x^3}{2\overline{R}n} & \text{otherwise} \end{cases}, \tag{4.17}$$

where we have $\kappa \triangleq \alpha d_c$. Moreover, it can also be shown that for any other node degree distribution $\tilde{\lambda}_{\underline{r}''}$ corresponding to a row-sum vector $\underline{r}''$ (unique up to a permutation)

$$\frac{Pr[\text{row sum of } H_S = \underline{r}'' \,|\, |S| = \alpha n]}{Pr[\text{row sum of } H_S = \underline{r}'^{*} \,|\, |S| = \alpha n]} \leq \eta^{-n \sum_i (\tilde{\lambda}^{(i)}_{\underline{r}''} - \tilde{\lambda}^{(i)}_{r'^{*}})^{+}}, \tag{4.18}$$

where $\eta \in (0,1)$ depends only on d_c and $\tilde{\lambda}^{(i)}_{\underline{r}''}, \tilde{\lambda}^{(i)}_{r'^{*}}$ denote the coefficients of x^i in the respective node degree distributions. In other words, row-sum vectors that are very different from $\tilde{\lambda}_{r'^{*}}$ become rarer as n becomes large since they have a probability that is exponentially smaller than that of $\tilde{\lambda}_{r'^{*}}$.

One notes from Observation 4.3.1 that as the channel erasure probability decreases, the expected fractional size of the stopping set ($E[\frac{|S|}{n} \,|\, S \neq \emptyset]$) also decreases. Equivalently, as the channel erasure probability decreases, $\alpha \triangleq \frac{|S|}{n}$ decreases and hence $Pr[\lfloor \frac{\alpha d_v}{1-R} \rfloor < 2] = Pr[\alpha < \frac{2}{d_c}]$ increases. When this probability becomes reasonably high, it can be seen from the optimal solution of (4.16) and from (4.18) that a linear fraction of bivalent check nodes can be noticed. Thus, at lower bit-error rates, bivalent check nodes start to appear and become more predominant in stopping sets. In Section 4.3.2, we will exploit this guaranteed presence of bivalent check nodes in our proposed algorithm to improve the performance in comparison to the standard MP algorithm.

For irregular ensembles, we expect a similar but more detailed argument justifying the observation. This above explanation for finite-length codes can be extended to the asymptotic case, using techniques developed in [18, 83] to quantify the asymptotic existence of bivalent check nodes in stopping sets for regular and left-regular ensembles.

Theorem 4.3.1. *Consider the (d_v, d_c) regular LDPC ensemble. For a stopping set S, let T_S denote the ratio of number of bivalent check nodes ($|C_2(N_G(S))|$) to the number of check equations in the Tanner graph of the code. Then, for each $\alpha \in (0, 1 - \frac{d_v}{d_c}) \cap \mathbb{Q}$, there exists $\beta^* \in (0, 1 - \frac{d_v}{d_c})$ and a strictly increasing sequence of codelengths $\{n_k\}_{k \in \mathbb{N}}$ such that for any $\beta \in (0, \beta^*)$,*

$$\lim_{k \to \infty} Pr[T_S \leq \beta \mid |S| \in \big(\alpha(n_k - \sqrt{n_k}), \alpha(n_k + \sqrt{n_k})\big)] = 0. \tag{4.19}$$

Moreover, the rate of convergence of the above probability to zero is exponential in the size of the code.

Proof. Refer to Section A.3 of Appendix A. ❑

Theorem 4.3.2. *Consider the left-regular LDPC ensemble with $\lambda(x) = x^{d_v-1}$ and design rate R. For a stopping set S, let T_S denote the ratio of number of bivalent check nodes ($|C_2(N_G(S))|$) to the number of check equations in the Tanner graph of the code. Then, for each $\alpha \in (0, 1-R) \cap \mathbb{Q}$, there exists $\beta^* \in (0, 1-R)$ and a strictly increasing sequence of codelengths $\{n_k\}_{k\in\mathbb{N}}$ such that for any $\beta \in (0, \beta^*)$,*

$$\lim_{k\to\infty} Pr[T_S \leq \beta \mid |S| \in \big(\alpha(n_k - \sqrt{n_k}), \alpha(n_k + \sqrt{n_k})\big)] = 0. \tag{4.20}$$

Moreover, the rate of convergence of the above probability to zero is exponential in the size of the code.

Proof. The proof is an obvious generalization of that of Theorem 4.3.1 using ideas for irregular ensembles presented in [83]. ❑

In other words, for long codes, almost all stopping sets of size linear in the codelength have a sizeable fraction of bivalent unsatisfied check nodes after the termination of message passing.

4.3.2 The Proposed Decoding Algorithm

In Section 4.3.1, we noted that the average size of stopping sets obtained when the message-passing decoder fails decreases monotonically with the channel erasure probability. We also noted by simulations that typical stopping sets of LDPC ensembles obtained at moderate bit-error rates of smaller than 10^{-1} contain a reasonably high fraction of bivalent check nodes (Observation 4.3.2). Combining these two observations, we note that at lower channel erasure probabilities, typical stopping sets have a sizeable fraction of bivalent check nodes among the unsatisfied ones. Since for a bivalent check node, the knowledge of one node implies the knowledge of the other, it is possible to just keep one of the two and replace the other completely by the former bit variable. This possibility of bit replacement whose full advantage is taken in the proposed algorithm by the use of the equivalence relation $\equiv$ on the set of bit nodes in the stopping set S defined as follows.

$$v_1 \equiv v_2 \iff v_1, v_2 \text{ belong to the same component of } G[S \cup C_2(G[S \cup N_G(S))]. \tag{4.21}$$

The rationale behind using this equivalence is as follows. Any two nodes in an equivalence class are connected by a path wherein each check node that the path traverses through is bivalent. Thus, it follows that the knowledge of a bit in an equivalence class implies the knowledge of all the bits in the equivalence class after a series of message-passing rounds. This enables the decoder to pick a representative bit node $r_{\mathcal{E}}$ from each equivalence class $\mathcal{E}$ and represent every other bit node in the class $\mathcal{E}$ as either $r_{\mathcal{E}} \oplus 1$ or $r_{\mathcal{E}} \oplus 0$ as the case may be. As an example, consider the stopping set presented in Figure 4.5. The dotted lines tagged to each check node represent the value to which the neighbors of the check nodes add up. The second graph $G[S \cup C_2(G[S \cup N_G(S))]$ is the induced subgraph generated by the stopping set and all neighboring bivalent check nodes. This induced subgraph has three components and the representative for each component is denoted by the emboldened variables $\mathbf{b}_{r_i}$, $i = 1, 2, 3$. As mentioned before, the value of every other bit (with respect to the representative bit) in the stopping set is determined by the relation defined by the check nodes lying in a path between the two nodes in their corresponding component. Ascertaining the components of $G[S \cup C_2(G[S \cup N_G(S))]$, choosing a representative for each component, and finding the size of each component can all be achieved in a parallel fashion

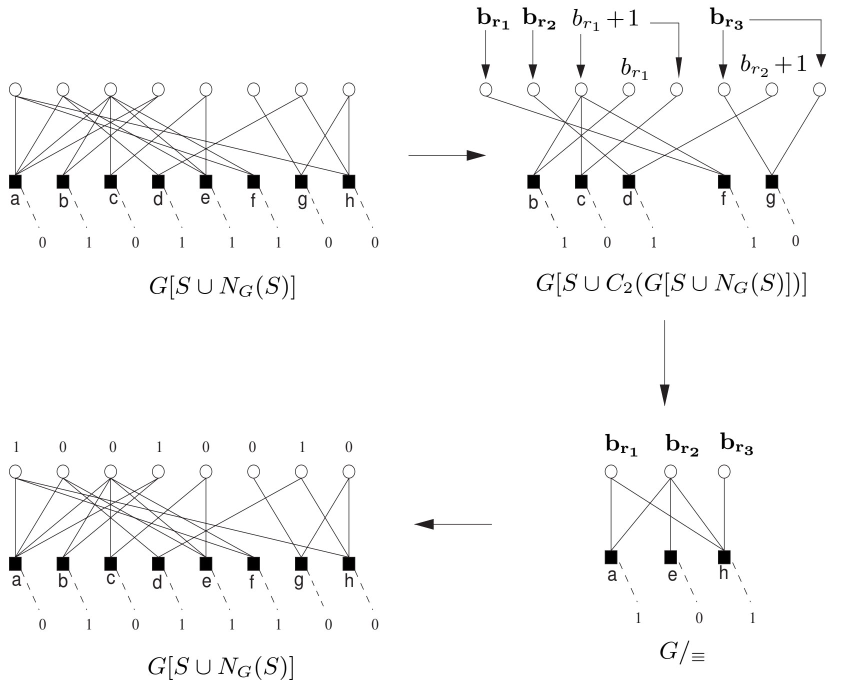

Figure 4.5: Example illustrating the proposed improved decoding algorithm.

by employing a *depth-first search* (DFS) algorithm [21] on the Tanner graph generated by the stopping set and its neighboring bivalent check nodes. Such a representation for all bits in terms of their representatives is lossless, and hence, one can replace the bits in every check equation (of weight 3 or more) by their representative bits and accordingly change the constants to which the check variables add up. What this replacement amounts to is the parallel contraction of bit nodes of each connected component of $G[S \cup C_2(G[S \cup N_G(S))]$ to its corresponding representative to construct the graph $G/_{\equiv}$. As mentioned above, while effecting the contraction, care must be taken to update the constants to which each check node adds up. Moreover, by definition of contraction, we delete all pairs of edges between a pair of nodes in the contracted graph, ensuring that the degrees of the check nodes after contraction are smaller than or equal to their respective degrees before contraction. In the example of Figure 4.5, it can be seen that the stopping set of eight bits is reduced to a Tanner graph of three bit and check nodes.

The advantage of this graph contraction is that, almost always, one recovers few bits. For example, at a BER of 10^{-2}, on the average, 8 and 15 representative bits get recovered (by just contraction and without any guessing) for codes of length 1000 from ensembles C1 and C3, respectively. Here, it should be noted that the actual number of bits that are released is much more. The strength of the algorithm lies in the contraction stage that effects the recovery of certain bits. These recovered bits can then be used to carry out another series of message-passing routines. This process of graph contraction, bit recovery, and message passing can be repeated a predetermined number (J) of times to achieve a suitable gain margin in the performance. In the example of Figure 4.5, it can be seen that after contraction, one recovers the bit $\mathbf{b}_{r_2}$ to be 0 from the check node e. Further iterations of message passing using this recovered bit results in complete decoding of this stopping set. The values of all the bits found thereby is given in the last graph of Figure 4.5.

However, at high BERs, the number of recovered bits decreases rapidly. This being detrimental to the performance of the decoder, one can resort to guessing a modest number N (say, 1 or 2) of bits just at the first round of contraction to start the recovery process. The bits can be chosen to be those corresponding to the N largest components in $G[S \cup$

$C_2(G[S \cup N_G(S))]$. Thus, the role of contraction is twofold − first to release some unknown bit of the stopping sets, and next, to provide a suitable rule for guessing bits. Further, this bit-guessing can be efficiently implemented by maintaining a binary N-tuple for each bit node that represents the relationship between the bit node and the N chosen bits and proceeding on lines similar to Algorithm C described in [93].

Algorithm 1 *Contraction-based Message Passing (CMP) Algorithm*

```
Require: Tanner Graph G
         Number of Contractions J
         Number of initial guesses to start initial commencement N
 1: Input: received word $\underline{c}$.
 2: Decode: $\underline{c}$ using MP decoder to obtain the stopping set $S$.
 3: if $|S| = 0$ then
 4:     Output the decoded word and go to 30.
 5: end if
 6: Construct graph $G' = G[S \cup C_2(G[S \cup N_G(S)])]$ from the residual Tanner graph.
 7: Perform a DFS algorithm on $G'$ to extract components (and their sizes), representative of each component
    and dependency of every node on the representative of the component that it lies in.
 8: Contract the bit nodes of residual graph preserving only the representative nodes to obtain $G''$.
 9: Pick $N$ bit nodes of $G''$ that represent the largest $N$ components of $G'$ and that are not univalent in $G''$.
10: Guess ← 1, Round ← J.
11: if Guess = $2^N$ then
12:     go to 29.
13: end if
14: Guess for each of the chosen $N$ bits, a binary pattern (that is not previously guessed) and Guess ←
    Guess+1.
15: Round ← Round−1.
16: Perform message passing on $G''$.
17: if a contradiction is reached then
18:     Round ← J and go to 11.
19: end if
20: Obtain smaller stopping set $S'$.
21: if $|S'| = 0$ then
22:     go to 28.
23: end if
24: if Round = 0 then
25:     go to 11.
26: end if
27: Set $S \leftarrow S'$ and do steps 3 through 6 and go to 15.
28: Decode the non-representative bits from their respective representatives by using the stored dependencies
    and output the decoded word and go to 30.
29: Output: “Decoding Faliure”.
30: Stop.
```

Furthermore, the improvement can be increased by appending to the parity-check, an auxiliary parity-check matrix whose rows are equations generated by the sum of check nodes of cycles of length 4 and 6 in the original Tanner graph. In some ensembles (e.g., C3), the average number of cycles of length 6 may be large. In such cases, one may choose to consider either fixing an upper bound on the size of the auxiliary matrix or adding only check nodes whose degree is bounded above by a fixed integer. Finally, this auxiliary matrix can be appended to the original parity-check matrix to reduce further the stopping set and to increase the number of bivalent check nodes. The proposed algorithm with two parameters denoted by J, N in the discussion above is summarized in Algorithm 1.

We would like to mention that our observations hold true for regular and irregular ensembles alike although all our analytical claims have been shown only for regular ensembles. We hope their extension to all LDPC ensembles will follow from similar but more detailed arguments. Finally, we would like to add that based on our observations, we have changed the message-passing decoder to perform intuitive modifications such as graph contraction to enhance the performance of the decoder. Since most of the steps are heuristic, the proposed algorithm has a lot of scope for minor tuning that can be specifically tailored to specific codes before implementation.

4.3.3 Running-Time Estimate of the CMP Algorithm

Let S denote the stopping set output by the MP decoder. With the help of the DFS algorithm, one can identify its components and their sizes, choose the component representatives, and identify the dependency of each bit node to its component representative in about $(2d_{v_{\max}} + 3)|S|$ operations. The operation of contraction can be done by contracting the neighbors of each check node with same representative, one after another. This can be achieved in $nR\binom{d_{c_{\max}}}{2}$ operations. The message passing at each step can be done in $\Theta(L_{\max}(|S| + nR))$ computations when we restrict the maximum number of decoder iterations to $L_{\max}$. Finally, an initial guess of N bits before graph contraction will take, on the average, 2^{N-1} guess patterns before the stopping set will be decoded and J iterations of the recovery, contraction and MP stages further result in a J-fold increase in the number of computations. Collectively, we note that the average estimated running time for a stopping set of size $|S|$ is of the order of

$$\left[J2^{N-1}\left((2d_{v_{\max}} + 3)|S| + nR\binom{d_{c_{\max}}}{2} + L_{\max}(|S| + nR)\right)\right]. \tag{4.22}$$

Furthermore, the exponential dependence on N can be reduced to a polynomial dependence by the same idea as that in [93]. However, as our results in the next section show, very small values of J, N such as $3, 2$, respectively, suffice to outperform the results of [93]. In comparison, the complexity of the improved decoding algorithm [93] with N' guesses is $\Theta(2^{N'-1} L_{\max}|S|)$. As we shall be seeing in the next section, our algorithm has an average running time similar to that of the improved decoding algorithm. This is because our our algorithm requires fewer guesses to obtain comparable performance. In other words, we obtain much gain in performance by guessing fewer bits and substituting the computations taken up by few guesses by those of contraction and recovery stages. In fact, by compressing the Tanner graph, one recovers not only some bits, but also gains insight as to which bits when guessed dissolves the stopping set faster.

4.4 Results of Simulation

First, the ensembles that were selected for simulated in this work are presented in Table 3.2 in Chapter 3. Next, we would like to mention that all decoding algorithms, whose results we provide in this section, were simulated using the extended parity-check matrices that were formed after appending the auxiliary parity-check equations (as described in Sec. 4.3.2) to the corresponding original parity-check matrices. This greatly increases the amount of contraction during the first stage. For example, for the ensemble C1 at a BER of 10^{-2} under MP decoding, using this auxiliary matrix we generate an additional 30 bivalent check nodes, which in turn increases the average number of recovered bits from 9 to about 35. In the case of ensemble C3 at the same BER, appending the auxiliary matrix has the effect of increasing the number of recovered bits from an average of 12 to 55.

Although the CMP algorithm is defined for all pairs of integers J, N, in practice, a reasonable choice for these parameters is $J = 3$ and $N = 2$ when codes of length 1000 are used. For this choice of parameters, the implementation is very efficient and fast. For example at a BER of 10^{-2}, our algorithm takes about 42% and 33% more time, on the average, than the standard message-passing algorithm for decoding codewords of length 1000 from ensembles C1 and C3, respectively. The above choice of parameters is also sufficient to provide improvement of several orders of magnitude, as shown in the following figures. Figure 4.6 provides the performance of a code of length 1000 from the regular $(3, 6)$

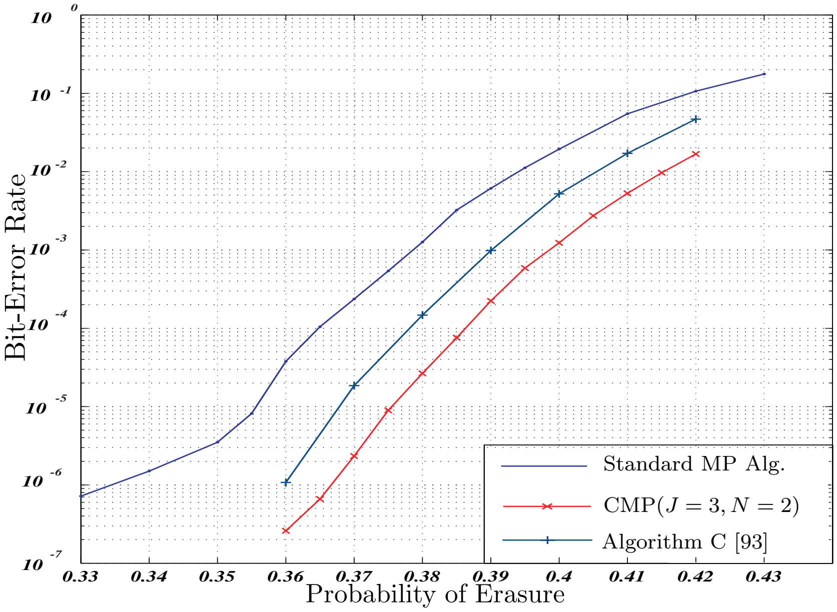

Figure 4.6: The performance of a code of length 1000 from ensemble C1 under message passing algorithm, CMP algorithm with $J = 3, N = 2$ and improved decoding algorithm [93] with $g_{\max} = 6$.

ensemble under three algorithms $-$ the standard MP algorithm, the improved decoding algorithm (Algorithm C) using six guesses [93], and our proposed algorithm with parameters $J = 3, N = 2$. All the algorithms were simulated using the parity check matrix appended with the auxiliary parity check matrix formed with equations of check nodes in cycles of length 4 and 6. Clearly, our proposed algorithm provides an improvement of an order of magnitude at a BER of 10^{-1} ($\varepsilon = 0.42$). At a BER of 10^{-4} ($\varepsilon = 0.365$), our algorithm provides an improvement of more than two orders of magnitude over the standard message-passing decoder, an order of magnitude more than Algorithm C proposed by P. Nik *et al.* Moreover, the expected running times of our proposed algorithm and Algorithm C (with $g_{\max} = 6$) are almost the same.

Figure 4.7 shows the performance of a code of length 1000 from the irregular ensemble C3 under the standard MP decoder, Algorithm C, and our proposed decoding scheme. Our algorithm provides more than one order of magnitude of improvement at a BER of 10^{-2} ($\varepsilon = 0.398$) and provides about three magnitudes of improvement at a BER of 10^{-4} ($\varepsilon = 0.369$). Although the performance for this code under our algorithm provides marginal gain over the performance of Algorithm C, it should be noted that our algorithm makes one-third the number of guesses as the other. Also to be noted is the fact that our algorithm performs much better than Algorithm C even at bit-error rates as high as 0.05, thereby showing the superiority of our algorithm over almost all bit-error rates of interest.

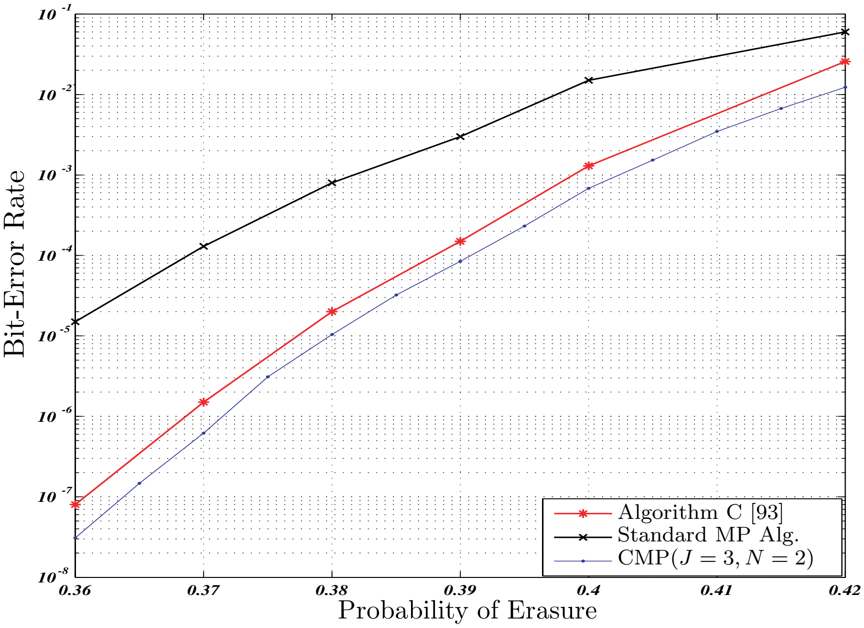

Figure 4.7: The performance of a code of length 1000 from ensemble C3 under message passing algorithm, CMP algorithm with $J = 3, N = 2$ and Algorithm C [93] with $g_{\max} = 6$.

CHAPTER V

RATE-COMPATIBLE PUNCTURING OF SHORT-LENGTH LDPC CODES

In certain time-varying channels where *channel state information* (CSI) is available, using a coding scheme with a fixed (code) rate is not optimal. To maximize throughput in such conditions, it is desirable to employ an error correcting scheme that is flexible in terms of its rate so that one can encode at a higher/lower rate when the channel becomes more/less reliable, respectively. Using several encoder-decoder pairs for achieving this flexibility in rate is, in most cases, undesirable. A practical technique for circumventing the above problem is *rate-compatible puncturing.* Puncturing is a good technique that has been studied extensively for various classes of error-correcting codes (and in specific, convolutional codes) [43, 9]. In puncturing, the number of parity bits that are sent over the channel vary depending on the condition of the channel. To send a smaller number of parity bits, a selected subset of the set of parity bits are transmitted. A family of punctured codes derived from a single parent code is said to be *rate-compatible* if puncturing is specifically done to ensure that the Tanner graph of a code of higher rate from the family is a subgraph of the Tanner graph of any code from the family that possesses a smaller rate. As a result, a single encoder-decoder pair suffices to decode any code from a rate-compatible family of punctured codes.

5.1 Related Work and Our Approach

The design and analysis of puncturing schemes for LDPC codes in the asymptotic setting (*i.e.*, as the codelength is made arbitrarily large) have been studied by many authors [38, 39, 40, 91]. However, the analysis of finite-length codes is much harder than that of its asymptotic counterpart. Some work in the area of puncturing schemes for finite-length LDPC codes can be found in [25, 100]. The issue of puncturing LDPC codes has been considered in [37, 83, 14, 126, 60] with each work considering a different approach. [60] considers the problem of designing a good rate-compatible ensemble of LDPC codes for a given range of rates. In [14], the authors consider the problem of designing a rate-compatible sequence of codes (using the differential evolution procedure [108]), where each code of the sequence is good at its respective rate. In [126], the authors choose a mother code at a rate that is in-between the range of required rates so that the excessive degradation because of puncturing to high rates is minimized. In their scheme, random puncturing is used to obtain codes of higher rates, whereas lower rates are achieved by extension using a novel progressive edge-growth based scheme [49].

Unlike most other works on puncturing LDPC codes, Ha *et al.* investigate the problem of puncturing a given LDPC mother code [37, 83]. They have shown by simulations that over the *additive white Gaussian noise* (AWGN) channel, the codes constructed by their puncturing scheme are superior to those obtained from random puncturing schemes. In [41], the authors observe a positive correlation between the probability of successful recovery of a bit and the number of unpunctured nodes in the recovery tree of a node. Therefore, in [41], the authors use a *grouping* algorithm to partition the set of bit nodes into various groups. Within each group, they identify the order for puncturing based on their *sorting* algorithm.

Just like in [41, 37], we investigate the design of a novel puncturing scheme from a *specific parent LDPC code* that results in a good rate-compatible family of LDPC codes. However, our approach to puncturing here is different from that of [41] in the following way. It neither uses the information regarding the degree of the survived check nodes, nor does it search for optimal nodes during the grouping algorithm. We can classify our puncturing scheme as a partitioning scheme that divides the set of bit nodes into groups. However, during the selection of nodes at any stage of the proposed puncturing scheme, the order in which the nodes are selected is unimportant. It is only important that we puncture the bits obtained from the first group before we puncture those of the next group.

5.2 Proposed Puncturing Scheme

In [41], Ha *et al.* approach puncturing finite-length LDPC codes based on their classification of punctured nodes into k *step recoverable nodes*, where k is a positive integer. The authors claim that, on the average, the magnitudes of the average *log-likelihood ratio*s (LLRs) for k step recoverable (k-SR) nodes go down monotonously as the index k increases. Based on this claim, they present a greedy algorithm to select the bits to be punctured. We suppose that the main shortcoming of this approach is the greedy selection of bits to be punctured. By this mode of selection, the reliability of nodes that are $(k+1)$-SR are, in a sense, critically dependent on the reliability of i-SR nodes for $i \leq k$. In other words, a non-greedy selection of the same number of nodes may guarantee higher probability of recovery for k-SR nodes with higher k than the greedy selection-based scheme, whereas the greedy scheme guarantees higher probability of recovery for k-SR nodes with lower k. However, the non-greedy selection may result in a better average probability of recovery when the probability of successful recovery of all punctured nodes are taken into consideration.

In this work, we take the following approach to rate-compatible puncturing. Consider the Tanner graph G of a parent LDPC code C. Suppose that the bits of $P \subseteq V(G)$ are punctured. Denote $\delta_P = \min\{d(u,v) : u, v \in P, u \neq v\}$ and $\eta_P(\mathcal{C}, l)$ to be the average probability of error after $l > 0$ rounds of message passing for an unpunctured node of a punctured code C that is sent through a channel $\mathcal{C}$. We call a puncturing pattern P to be l-optimal if P satisfies the following condition.

$$\eta_P(\mathcal{C}, l) = \min_{\substack{P' \subseteq V(G) \\ |P'| = |P|}} \eta_{P'}(\mathcal{C}, l). \tag{5.1}$$

The following result relates the concept of l-optimality with the minimum distance between the punctured bits over the Tanner graph.

Lemma 5.2.1. *Let G be the Tanner graph of a regular LDPC code C with girth $\Delta(G)$. Let $0 < l < \frac{1}{2}\Delta(G)$ be an integer and let $P \subseteq V(G)$ denote the set of bits to be punctured. Then $\delta_P \geq 2l + 1$ guarantees l-optimality of P.*

Proof. Suppose that $\delta_P \geq 2l + 1$. Then one sees that for every punctured bit node w, the decoding neighborhood [101] of depth $2l$ is a tree and that all bit nodes in the neighborhood are unpunctured. Also, the bit nodes in the decoding neighborhood of depth $2l$ of any two punctured nodes are (node-)disjoint. Since after l iterations, the probability of error for every node in the graph depends only on the decoding neighborhood of depth $2l$, by symmetry, one concludes that the average probability of error for every punctured node is the same. Finally, one can show that the probability of successful decoding of a bit after l iterations with no unpunctured bit in its decoding neighborhood of depth $2l$ is higher than a node whose decoding neighborhood of depth $2l$ contains other punctured nodes. Using the above fact, one can show that such a puncturing scheme P offers the least probability of decoding failure for every unpunctured node after l iterations, and hence is l-optimal. ❑

Thus, we see that the condition of l-optimality (for $l < \frac{1}{2}\Delta(G)$) is guaranteed if one chooses the punctured bits to be a distance of $2l$ away from each other. However, it is not clear as to how the concept of l-optimality relates to the universal definition of optimality that is based on average bit-error rate (BER). It should also be noted that finding a sufficient criteria guaranteeing l-optimality for $2l > \Delta(G)$ is mathematically intractable, since the decoding neighborhood for iterations $l > \lfloor \frac{\delta_P - 1}{2} \rfloor$ is no longer a tree. Hence, finding l-optimal puncturing patterns becomes a difficult problem. Albeit being optimal in the sense defined above, l-optimal puncturing schemes have several shortcomings. For example, for a regular LDPC code of length n from the (d_v, d_c) ensemble, choosing the set of punctured bits P with $\delta_P > 2l > 2$ implies that no more than $\lfloor \frac{n}{(d_v d_c)^{\lfloor \frac{l}{2} \rfloor}} \rfloor$ bits can be punctured. This poses a serious constraint on the range of achievable rates. Another serious issue with this approach is that of rate-compatibility. It is highly unlikely that there exists a nested family of subsets of bit nodes that are each l-optimal or almost l-optimal for some positive integer l. However, we can design puncturing patterns using the intuition that keeping the punctured bits far from each other ensures that decoding neighborhoods contain more unpunctured bit nodes, thereby guaranteeing a good performance under message-passing decoder. Presented below (Algorithm 2) is a puncturing method that incorporates loosely the above idea of distance while selecting the bits to be punctured. More specifically, each round of our proposed scheme returns a set of nodes that is 1-optimal.

The algorithm takes in a parity-check matrix of an LDPC code and returns the set of indices to be punctured (INDEX). To compute INDEX, the algorithm proceeds as follows. Initially, the algorithm starts with an empty INDEX. At each time step **3** is executed, the algorithm selects a bit node uniformly at random from the set of available indices (REMAIN) and appends the selected vertex to INDEX. It adds all the neighbors of the check nodes connected to the selected node to NEIBOR and again deletes all elements of NEIBOR from REMAIN. In this way, the algorithm guarantees that bit nodes that will be selected in any step is at least a distance of four hops (in the Tanner graph) from any previously selected bit node. Of course, as the algorithm proceeds REMAIN gets smaller and NEIBOR gets bigger until finally REMAIN becomes empty. At this point, NEIBOR is re-initialized and all those check equations that involve the bit nodes in INDEX are deleted from H. Similarly, all bit nodes that do not participate in the remaining equations are also deleted to obtain the smaller matrix H' (and its corresponding Tanner graph G'). The algorithm is then re-run with H' as input and the whole process is repeated until $H' = \emptyset$.

Algorithm 2 ***Puncturing Scheme***

```
Require: Tanner graph $G$ of an $m \times n$ matrix $H$
 1: (Initialization) Set INDEX= $\emptyset$, REMAIN= $\{1, ..., n\}$ and NEIBOR= $\emptyset$.
 2: If REMAIN$\neq \emptyset$, go to 3, else go to 6.
 3: Find a vertex $v$ uniformly at random from REMAIN.
 4: Set INDEX=INDEX$\cup \{v\}$ and NEIBOR=NEIBOR$\cup N_G(N_G(v))$.
 5: Set REMAIN=REMAIN$\setminus$NEIBOR and go to 2.
 6: (Re-initialization) Set NEIBOR= $\{1, 2, ..., n\} \setminus$INDEX.
 7: Set $J$ to be the set of equations involving only bit nodes in NEIBOR.
 8: Set $G' = G[N_G(J) \cup J]$.
 9: Set $H'$ to be the parity check matrix corresponding to $G'$.
10: If $H' \neq \emptyset$, then set $H = H'$, $G = G'$ and go to 1, else output INDEX and halt.
```

It must be noted here that the algorithm presented above is generic and numerous improvisations can be done to boost its performance depending on the ensemble considered. For example, step **3** may be modified for irregular LDPC codes to select bits based on the degree of the bit instead of a uniformly random selection. However, from our simulations, we find that the improvement, in general, is not drastic and can be left to the choice of the designer. Another important point to be noted is the following. At the first round of selection of the bits to be punctured, step **5** ensures that for each node v in INDEX, there is at least one neighboring check node c_v whose all neighbors excluding v are unpunctured. Therefore, if all the unpunctured neighbors of c_v are known with high accuracy after sufficient iterations of decoding, one can identify v with reasonable accuracy. In other words, the set of punctured bits constructed by the Puncturing Scheme does not contain a stopping set, and hence, one can expect codes of our scheme to perform better than those of random puncturing at least over the BEC.

Suppose that the Puncturing Scheme is completed after L rounds of bit selection and deletion and at the i^{th} round, the algorithm appends a subset S_i of bit nodes to INDEX. Then, it can be observed that if all the unpunctured nodes of the code are received with high accuracy, then the values of bits in the set S_i will be known with high accuracy after the $(L-i+1)^{\text{th}}$ iteration of message passing. Hence, by the notation of [41], these nodes are $(L-i+1)$-SR although the criteria for selection of these nodes follow an approach different from that in [41]. In [41], the criteria for bit selection in the grouping algorithm is based on the number of unpunctured nodes in the recovery tree of each node and a search for the optimal node is done at each instant of selection. Here, we perform much less work by simply guaranteeing just 1-optimality at each round of selection. Also, within each group, there is no sanctity in the order by which the nodes are selected. However, the amount of computational work done in the selection has a direct bearing on the limited range of rate-compatibility achievable by the Puncturing Scheme alone. In order to increase the range of rate-compatibility, the Additional Puncturing Scheme given below (Algorithm 3) may be implemented. In this scheme, nodes are selected one at a time from the set of nodes that participate the least in the equations that relate punctured bits to unpunctured bits. This ensures that by selection of an additional bit, the reduction in the probability of recovery of previously punctured bits is minimized. Also, care is taken at each step to ensure that stopping sets are not contained in the set generated by appending the newly selected bit node to the set of all previously selected nodes.

Algorithm 3 ***Additional Puncturing Scheme***

```
Require: Tanner graph $G$ of an $m \times n$ matrix $H$.
Generate INDEX and $S_i, i = 1, ..., L$ using the Puncturing Scheme.
Set $I$ to be the set of all rows of $H$ that involve bits nodes from exactly one set $S_j$ for some $0 \leq j \leq L$.
Set $B$ to be the set of all those bit nodes $v \in$INDEX$^c$ that are involved in the least number of equations represented in the rows $I$ of $H$ and such that no stopping set is contained in INDEX$\cup\{v\}$.
If $B = \emptyset$, go to 6, else go to 5.
Select $b \in B$ at random and set INDEX=INDEX$\cup\{b\}$, $B = B \setminus \{b\}$ and go to 3.
Output INDEX and halt.
```

5.3 Results of Simulations

In this section, we present the results of our simulation for the proposed scheme described in Section 5.2 over the erasure channel and the AWGN channel. Codes of length 1000 were simulated from ensembles of codes notated in Table 3.2 of Chapter 3. It must be mentioned here that the design rates of codes from ensembles C1 and C3 are both 0.5, whereas those from ensembles C2 and C4 are 0.2 and 0.38, respectively. To simulate our proposed rate-compatible scheme, the Puncturing Scheme was simulated in conjunction with the Additional Puncturing Scheme. Since our scheme involves a random selection of a subset of vertices, during every run of simulation, we obtain a different subset of indices to be punctured (INDEX). However, for the chosen ensembles and codelengths, we find that the average size of INDEX after completion of the Puncturing Scheme alone for multiple independent runs of the scheme to be 229, 322, 230 and 232 for C1, C2, C3 and C4, respectively. Not using the Additional Puncturing Scheme would then mean a limited range of achievable rates. For example, the maximum rate attainable by this scheme for ensemble C1 will be only about $\frac{500}{1000-229} \approx 0.65$. For each ensemble, for more that 95% of the runs, the Punctured Scheme terminates after two rounds of selection and graph contraction. Also, approximately 60%, 72%, 63%, and 61% of the bits selected are from the first round for codes C1, C2, C3, and C4, respectively. However, using the Additional Puncturing Scheme, we can increase the range of rate-compatibility significantly. For example, we can extend the range of rate-compatibility to 0.885 and 0.87 for ensembles C1 and C3, respectively.

In the following, we summarize the results of simulation for the performance of the punctured codes over BEC and AWGN channels. For the sake of convenience, in the figures that follow, we index codes constructed from our proposed punctured scheme by X and by that from Ha *et al.* [41] by Y. Also, we do not present the results of simulations for ensembles C2 or C4, since inferences similar to those made for the ensembles C1 and C3 also hold for ensembles C2 and C4.

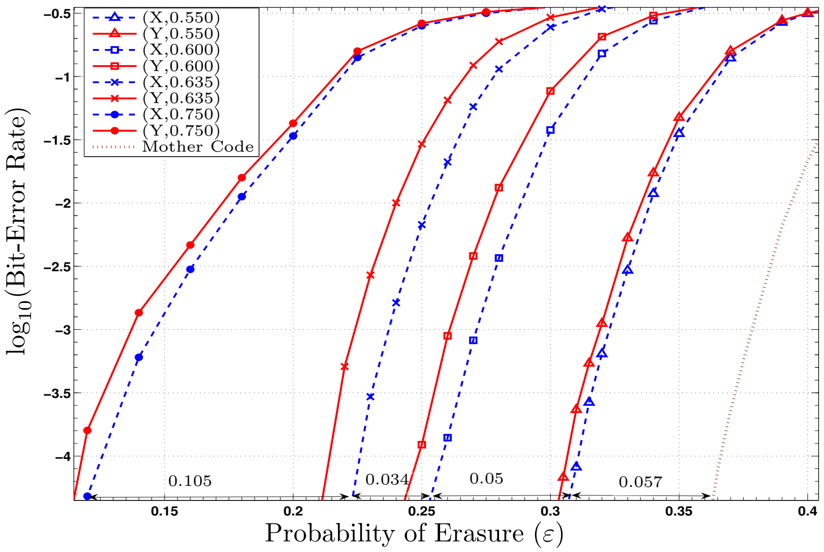

Figure 5.1: BERs of punctured codes of various rates obtained from a code from the ensemble C1 over the BEC.

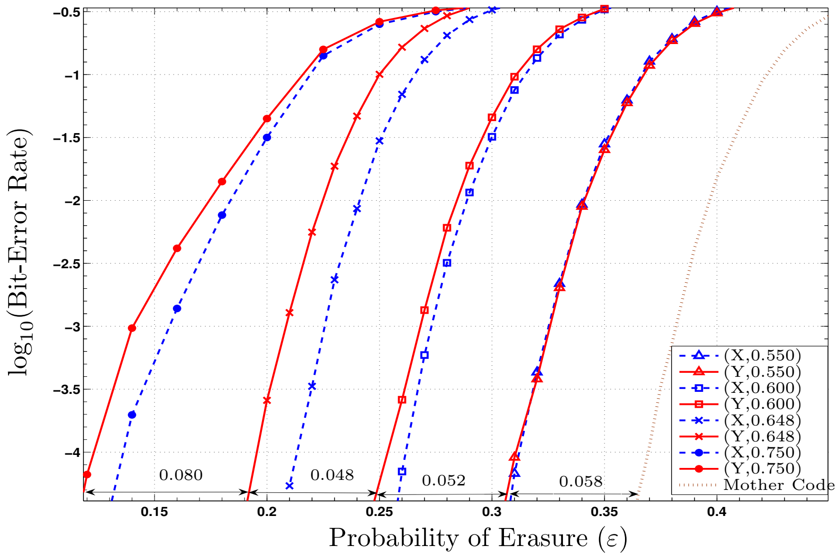

Figure 5.2: BERs of punctured codes of various rates obtained from a code from the ensemble C3 over the BEC.

The bit-error rates of the punctured codes obtained from ensemble C1 and C3 are presented in Figures 5.1 and 5.2, respectively. The rates of the punctured codes are presented as the second argument in the legend of the figures. It must be noted that punctured codes with lower rates employ only the bit nodes that are selected using the Puncturing Scheme, whereas codes with higher rates, specifically, those with a rate of 0.75 uses the bits selected from the Additional Puncturing Scheme also.

From the figures, it is clear that punctured codes derived from scheme X perform better than those of scheme Y at all simulated rates. It is interesting to note that under our scheme the difference in erasure probabilities required for a BER of 3×10^{-5} for the parent and the daughter codes (generated by using only Puncturing Scheme) is almost identical to the difference in their rates. Therefore, the punctured codes are as good at the lower rate as the parent code is at its rate. It is also interesting to note that the difference between our codes and those from scheme Y become stark when the puncturing fraction is higher therefore showing that codes derived from scheme Y deteriorate more due to puncturing than ours. In fact, our scheme offers an improvement of up to two orders of magnitude at higher SNRs (lower ε). However, the substantial benefits in BER gained by using the bits of the Puncturing Scheme start to diminish when the bits obtained from the Additional Puncturing Scheme are used. This can be explained by the following argument. The set of bits to be punctured (INDEX) that is returned by the Puncturing Scheme is maximal in the sense that there is no equation in the parity-check matrix that solely relates only nodes in NEIBOR=INDEX^{c}. Once this limit is reached, further selection of bits seems to create a critical dependence of the accuracy of the newly punctured bits on those of the bits obtained from the Puncturing Scheme. As a result, the degradation is accelerated from this point on. However, the benefits obtained by the punctured nodes obtained by the Puncturing Scheme is sufficient to guarantee that our scheme remains superior for a greater range of rates including those that are achievable by puncturing the bits obtained from the Additional Puncturing Scheme.

Figures 5.3 and 5.4 present the BERs of punctured codes from ensembles C1 and C3 for schemes X and Y over the AWGN channel. As before, we see that at higher rates,

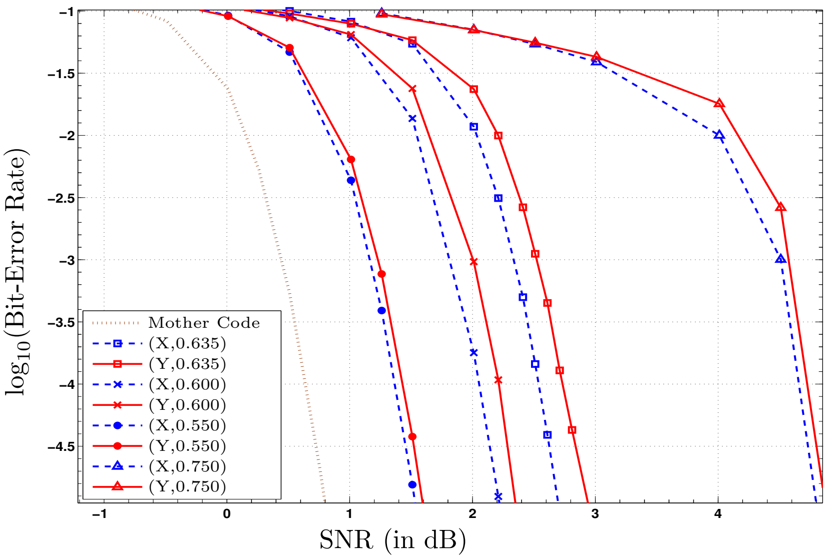

Figure 5.3: BERs of punctured codes of various rates obtained from a code from the ensemble C1 over the AWGN channel.

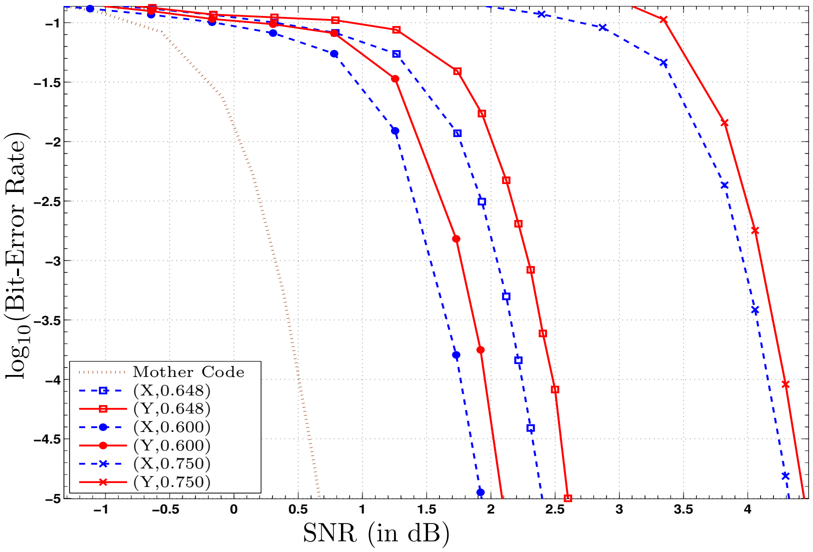

Figure 5.4: BERs of punctured codes of various rates obtained from a code from the ensemble C3 over the AWGN channel.

punctured codes from scheme X outperform those of scheme Y by about two orders of magnitude. Again, it is noticed that the improvement in performance obtained by using our puncturing scheme starts to diminish when bits from the Additional Puncturing Scheme are punctured. Finally, over both the channels that were simulated, it was observed that our scheme provided a *word-error rate* (WER) improvement of about an order of magnitude in comparison to the scheme in [41].

CHAPTER VI

PERFORMANCE OF RANDOMLY PUNCTURED LDPC CODES

Puncturing is an established approach to constructing good families of codes for time-varying channels whose channel state information is available. Puncturing has been studied extensively for various classes of error-correcting codes (and in specific, convolutional codes) [43, 9]. For the class of LDPC Codes, the analysis of the effects of puncturing has been studied by many authors for both asymptotic and short-length scenarios [38, 39, 40, 94, 91, 36]. However, finite-length codes and their puncturing are lot more difficult to analyze due to their mathematical intractability. Urbanke *et al.* have studied finite-length LDPC codes in [25, 100]. However, substantial work concerning the degradation of performance experienced due to puncturing is not available. This information on the amount of degradation due to puncturing would be useful in the design and selection of good ensembles for puncturing. In this work, we present bounds on the performance of punctured LDPC codes derived from a given finite-length LDPC (parent) code over the BEC. These bounds use a qualitative fact about the distribution of stopping sets in the parent LDPC code employed to relate the performance of the punctured code to that of the parent code at a different channel parameter. The basic idea behind the bounds is the following. Given the performance (*i.e.,* the bit-error rate $\eta(\varepsilon)$) of the mother code for all channel parameters ε of interest, we identify functions $\omega_p, \Omega_p : [0,1] \longrightarrow [0,1]$ such that the average bit-error rate η_p (over a BEC with erasure probability ε) achievable by randomly puncturing p bits of the mother code is bounded in the following manner.

$$\eta(\omega(\varepsilon)) \leq \eta_p(\varepsilon) \leq \eta(\Omega(\varepsilon)). \tag{6.1}$$

The problem therefore reduces to identifying the functions ω_p and Ω_p for various channel erasure probabilities.

The codes used for simulations to evaluate our results in this work are notated in Table 3.2 of Chapter 3. Some of the codes (namely C1 and C3), also used in [93], are such that their degree distributions satisfy $\lambda'(0)\rho'(1) < 1$ and hence have favorable stopping set distributions. In such ensembles, the size of the minimum stopping set (and the minimum distance) of a typical code grows linear in the codelength asymptotically. Conversely, codes from an ensemble that does not satisfy the condition and have many small stopping sets and exhibit error floor behavior.

The organization of this work is as follows. The following section presents the bounds on the average bit-error rates (BERs) of codes generated by random puncturing codes in terms of the bit error rate of the parent LDPC code. Section 6.2 presents our results of simulations that evaluate our derived bounds.

6.1 The Expected Performance of Randomly Punctured Family of Codes

It is well understood that the performance of LDPC codes over the BEC when decoded using iterative message passing is directly related to the distribution of the stopping sets. The analysis of this distribution becomes rather cumbersome and intractable for finite lengths [25]. Here, we take a practical approach in deriving bounds on the average performance of randomly punctured ensembles derived from a given parent code $\mathcal{C}$. These bounds derived in this section rely not on the exact distribution of the stopping sets of the parent code, but on certain qualitative facts concerning the distribution. Using the qualitative facts, we then relate the average performance of a punctured code over the BEC to the performance of the parent code over an erasure channel with higher erasure probability. Such a relation will be practically very significant since one has to in any case know that the performance of the parent code to decide whether its performance is satisfactory before one attempts to construct punctured codes from it. Moreover, the information of the distribution of stopping sets is intrinsically woven into the performance measure (such as bit-error or word-error rates) of the parent code.

Before we proceed to the bounds, we recall some properties regarding binomial random variables. Let for $X \sim B(n, s)$, $f(k, n, s) \triangleq P[X = k]$ and $F(k, n, s) \triangleq P[X \leq k]$. For notational convenience, we extend the two definitions in the following obvious manner.

$$f(k,n,s) = \begin{cases} 0 & k < 0 \text{ or } k > n \\ \binom{n}{k} s^k (1-s)^{n-k} & 0 \leq k \leq n \end{cases} \tag{6.2}$$

$$F(k,n,s) = \sum_{-\infty < j < k} f(j,n,s). \tag{6.3}$$

With the above notations, the following results hold.

Lemma 6.1.1. *Let $n \in \mathbb{N}$, $0 < p < n$, and $\varepsilon_1, \varepsilon_2 \in (0,1)$. Define: $g_1, g_2 : \{0, \ldots, n\} \longrightarrow \mathbb{R}$ by $g_1(i) = f(i,n,\varepsilon_1)$ and $g_2(i) = f(i-p, n-p, \varepsilon_2)$. Then, $g_1 - g_2$ changes sign at most twice. More specifically, if means are matched,* i.e., *$n\varepsilon_1 = p + (n-p)\varepsilon_2$ and $\frac{n\varepsilon_1}{p} > 1 + \frac{\log(\pi n e^{\frac{1}{3}})}{2p}$, then the sign of $g_1 - g_2$ alternates exactly twice.*

Proof. For a proof, refer Section A.4 of Appendix A. ❑

Corollary 2. *In the case of matched means, when $\frac{n\varepsilon_1}{p} > 1 + \frac{\log(\pi n e^{\frac{1}{3}})}{2p}$, $F(i,n,\varepsilon_1) - F(i-p, n-p, \varepsilon_2)$ changes sign exactly once for $i \in \{0, \ldots, n\}$. For $\frac{n\varepsilon_1}{p} \leq 1 + \frac{\log(\pi n e^{\frac{1}{3}})}{2p}$, the argument in Section A.4 of Appendix A can be extended to show that $F(i,n,\varepsilon_1) - F(i-p, n-p, \varepsilon_2)$ either has exactly one change of sign from positive to negative or remains non-negative for $i \in \{0, \ldots, n\}$.*

Lemma 6.1.2. *Suppose that $R \in (0,1)$ and $0 \leq \varepsilon \leq \frac{\overline{R} - n^{-1}p}{1 - n^{-1}p} - \alpha$ for some $\alpha > 0$ and positive integer $p < \overline{R}n$. Then,*

$$\sum_{i=0}^{\overline{R}n} (F(i,n,\varepsilon(1-\frac{p}{n}) + \frac{p}{n}) - F(i-p, n-p, \varepsilon)) \leq Rne^{\alpha^2 n(1-n^{-1}p)^2}. \tag{6.4}$$

Proof. For a proof, refer Section A.5 of Appendix A. ❑

The rest of this section is divided into two subsections. The first presents the upper bound and the second derives lower bounds.

6.1.1 The Upper Bound

For an LDPC code $\mathcal{C}$ of length n and $0 \leq i \leq n$, let $S_i(\mathcal{C})$ be the average size of the stopping set given that i erasures have occurred. Then, the BER $E_{\mathcal{C},0}(\varepsilon)$ of the unpunctured code $\mathcal{C}$

over the erasure channel with erasure probability ε is given by

$$E_{\mathcal{C},0}(\varepsilon) \triangleq \sum_{0\le i\le n} \binom{n}{i} \frac{S_i(\mathcal{C})}{n} \varepsilon^i (\overline{\varepsilon})^{n-i}. \tag{6.5}$$

When a randomly punctured code of length $n-p$ is used over a BEC of erasure probability ε, the decoder sees a distribution of erasures that is binomial *i.e.*, $B(n-p, \varepsilon)$ over and above the random set of erasures (of size p) corresponding to the bits punctured. Note that the binomial distribution arises from the bits that are erased during transmission. The performance of the punctured code over the BEC averaged over all realizations of the channel and the puncturing pattern (denoted by $E_{\mathcal{C},p}$) can be seen to be

$$E_{\mathcal{C},p}(\varepsilon) \triangleq \sum_{p\le i\le n} \binom{n-p}{i-p} \frac{S_i(\mathcal{C})}{n} \varepsilon^{i-p} (\overline{\varepsilon})^{n-i}. \tag{6.6}$$

Before we proceed to derive the upper bound,we present the following theorem characterizing this monotonicity of the sequence $\{S_i(\mathcal{C}) : i = 1, \ldots, n\}$.

Lemma 6.1.3. *For any code $\mathcal{C}$ of length n, the sequence $\{\frac{S_i(\mathcal{C})}{i} : i = 1, \ldots, n\}$ is an increasing sequence.*

Proof. For a proof, refer to Section A.6 of Appendix A. ❑

The fact that one cannot exactly characterize the sequence in any greater detail stems from the fact that the approximation in (A.50) of Appendix A cannot, in general, be made tighter. Lemma 6.1.3 can only be extended to the following observation that has been verified by simulations.

Observation 6.1.1. *For an LDPC code $\mathcal{C}$ of length n and rate R, the variation of the sequence $\{S_i(\mathcal{C}) : i = 1, .., n\}$ is characterizable in three regions based on the index i. There exist $m \lesssim \overline{R}n$ and $M \gtrsim \overline{R}n$ such that the sequence $S_i(\mathcal{C})$ is convex on $\{0, 1, \ldots, m\}$. For indices $i > M$, $S_i(\mathcal{C}) \approx i$. The set of indices sandwiched in the middle ($\{m+1, \ldots, M\}$) is a transition phase where the property of the sequence changes from convex to almost linear.*

Figure 6.1 shows clearly the three regions described in Observation 6.1.1 for an LDPC code of length 500 from the ensemble C1. We shall make use of this observation to derive

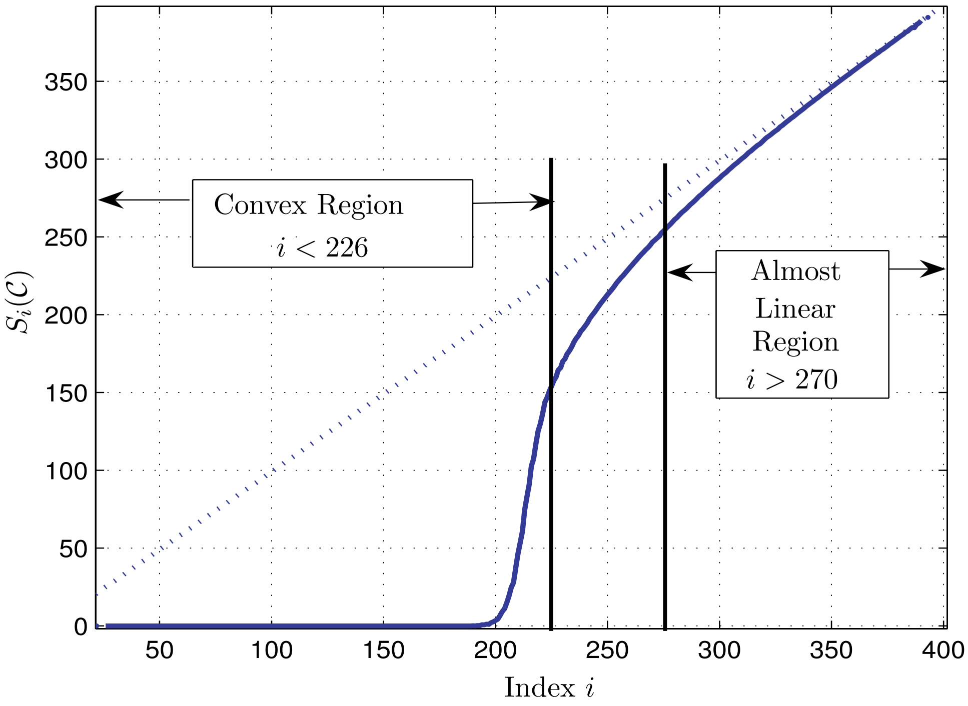

Figure 6.1: $S_i(\mathcal{C})$ vs i for a code of length 500 from ensemble C1.

bounds on $E_{\mathcal{C},p}$ in terms of $E_{\mathcal{C},0}$. However, we shall make the simplifying assumption that the region of convexity ranges all the way up to $\overline{R}n$ (i.e., $m = \overline{R}n$). We describe the consequences of this assumption while presenting the results of our simulation in Section 6.2. Before we present the upper bound, we present a discussion outlining the underlying idea.

Observation 6.1.1 will be used in conjunction with properties of binomial random variables to estimate the contribution to the BER by the three regions of the set of indices. Particularly, the contributions to $E_{\mathcal{C},p}, E_{\mathcal{C},0}$ from the two non-convex regions are estimated to be both bounded by $Rne^{-2\frac{\alpha^2(n-p)^2}{n}}$ (from Lemma 6.1.2). The contributions of the convex region to $E_{\mathcal{C},0}$ and $E_{\mathcal{C},p}$ are then estimated for appropriate choices of the channel parameters to derive the following bound.

Theorem 6.1.1. *(The Upper Bound) For a code $\mathcal{C}$ of rate R,*

$$E_{\mathcal{C},p}(\varepsilon) \leq E_{\mathcal{C},0}(\varepsilon(1-\frac{p}{n})+\frac{p}{n}), \qquad 0 < \varepsilon \lesssim \frac{\overline{R}n-p}{n-p}. \tag{6.7}$$

Proof. For a proof, refer to Section A.7 of Appendix A. ❑

Here, we would like to briefly concentrate upon a certain aspect of the proof of the above bound. The contribution of the non-convex regions to the both the BERs in the inequality of (6.7) are exponentially small due to Lemma 6.1.2. However, as the channel erasure probability approaches the upper limit of the range prescribed in Theorem 6.1.1, the neglected terms get larger and hence the bound may be inaccurate or invalid. We shall investigate the effect of the same via simulations in Section 6.2.

6.1.2 The Lower Bound

In this subsection, we derive lower bounds on the performance of the punctured code using that of the parent code at different channel conditions. To derive the same, we shall make use of the upper bound derived in the previous subsection.

Before we present a detailed lower bound, we present a simple lower bound that is based on *single-sign change* condition. For a given p and ε, we say ε' satisfies the single-sign change condition if and only if $[f(i-p,n-p,\varepsilon)-f(i,n,\varepsilon')]$ changes sign at most once from negative to positive as i increases from 0 to n. The following intuitive lower bound exploits this condition.

Lemma 6.1.4. (*Single-Sign Change Bound*) *For a code $\mathcal{C}$ of rate R,*

$$E_{\mathcal{C},p}(\varepsilon) \geq E_{\mathcal{C},0}((\varepsilon)^{1-\frac{p}{n}}), \qquad 0<\varepsilon<\overline{R}. \tag{6.8}$$

Proof. By an argument similar to that of the proof of Theorem 6.1.1 to show that for any ε' that satisfies the single-sign change condition (*i.e.*, $[f(i-p,n-p,\varepsilon)-f(i,n,\varepsilon')]$ changes sign only once from negative to positive in the entire range of $i=0,\ldots,n$), the following also holds.

$$E_{\mathcal{C},p}(\varepsilon)-E_{\mathcal{C},0}(\varepsilon') = \sum_{i=0}^{n} \frac{S_i(\mathcal{C})}{n}[f(i-p,n-p,\varepsilon)-f(i,n,\varepsilon')] \geq 0. \tag{6.9}$$

One can see that the maximum possible choice of ε' for a given ε is when the second intersection is allowed to happen at $i=n$. Equivalently,

$$f(n-p,n-p,\varepsilon)-f(n,n,\varepsilon')=0 \Rightarrow \varepsilon'=\varepsilon^{1-n^{-1}p}. \tag{6.10}$$

❑

Although this bound is simple and easy to estimate, this bound is not very tight and hence not very useful in estimating the performance of the punctured code.

However, from the earlier subsection, we see that when the channel erasure probability ε' seen by the punctured code is suitably smaller than $\frac{\overline{R}n-p}{n-p}$, we can neglect the effect of stopping sets of size greater than $\overline{R}n$. If we restrict the channel parameter to the same suitable range, we can then ease the single sign-change condition further to derive a better lower bound in the following way.

Theorem 6.1.2. (*Modified Single Sign-Change Bound*) *For $R \in (0,1)$ and $\varepsilon \leq \frac{\overline{R}n-p}{n-p}$ denote $\varepsilon^*(\varepsilon)$ to be the unique solution for the variable ε' in*

$$h(\overline{R}) + \overline{R}\log(\varepsilon') - R\log(\overline{\varepsilon}') = (1-\frac{p}{n})h(\frac{n\overline{R}-p}{n-p}) + (\overline{R}-\frac{p}{n})\log\varepsilon + R\log(\overline{\varepsilon}). \qquad (6.11)$$

Then, for a code $\mathcal{C}$ with rate R,

$$E_{\mathcal{C},p}(\varepsilon) \geq E_{\mathcal{C},0}(\varepsilon^*(\varepsilon)), \qquad 0 < \varepsilon \lesssim \frac{\overline{R}n-p}{n-p}. \qquad (6.12)$$

Proof. Since the indices $i > \overline{R}n$ do not contribute significantly to the BER (in this chosen range of channel erasure probability), we can let $[f(i-p,n-p,\varepsilon) - f(i,n,\varepsilon')]$ to change sign twice provided the second change of sign happens between $\overline{R}n$ and n. Of course, the best lower bound will be obtained when we let ε to be the channel erasure probability that guarantees the second point of sign-change of $[f(i-p,n-p,\varepsilon) - f(i,n,\varepsilon')]$ to be at $i = \overline{R}n$, which in turn necessitates

$$\binom{n}{\overline{R}n}\varepsilon'^{\overline{R}n}(\overline{\varepsilon'})^{Rn} = \binom{n-p}{\overline{R}n-p}(\varepsilon)^{\overline{R}n-p}(\overline{\varepsilon})^{Rn} \qquad (6.13)$$

$$\Rightarrow h(\overline{R}) + \overline{R}\log(\varepsilon') + R\log(\overline{\varepsilon}') = (1-\frac{p}{n})h(\frac{n\overline{R}-p}{n-p}) + (\overline{R}-\frac{p}{n})\log\varepsilon + R\log(\overline{\varepsilon}). \qquad (6.14)$$

Finally, the unique solution $\varepsilon^*(\varepsilon)$ for the unknown ε' in (6.14) for each given ε can be found by the method of bisection [7], if needed. ❑

In both lower bounds presented above, we choose ε' such that the second point of sign-change for the function $[f(i-p,n-p,\varepsilon)-f(i,n,\varepsilon')]$ is fixed either at $i = n$ or at $i = \overline{R}n$. The assumption that the effect of stopping sets of size greater than $\overline{R}n$ on BER is insignificant becomes strongly valid for smaller erasure probabilities. However, this fact is not reflected on the improved lower bound. Thus, for very low BERs (very high SNRs), the modified single sign-change lower bound is similar to the one obtained from the single sign-change condition.

The key idea to strengthen the lower bound further for high SNRs is to adaptively change the second point of sign-change from $\overline{R}n$ to lower values based on the channel erasure probability ε. The bound hinges on the fact that for the aforementioned range of channel parameters, the effect due to stopping sets of size greater than $\overline{R}n$ play an insignificant role in determining the magnitude of $E_{\mathcal{C},p}(\varepsilon)$. However, as the channel erasure probability is reduced from the upper limit $\frac{\overline{R}n-p}{n-p}$, the sizes of stopping sets that significantly determine the magnitude of $E_{\mathcal{C},p}(\varepsilon)$ decreases. Therefore, we can potentially change the index of the modified sign-change bound with the channel parameter instead of fixing it at $\overline{R}n$. To do just that, we fix a "small" fraction $1 \gg \beta > 0$, which shall be the fraction of contribution to BER by tail indices that we shall neglect. For a given channel parameter ε for the daughter code and β, we identify the index J_β which shall play the same role as $\overline{R}n$ in the previous bound. The index J_β is identified as the solution to the following equation.

$$\sum_{k>J_\beta-p} \binom{n-p}{k-p} \frac{S_{k-p}(\mathcal{C})}{n} (\varepsilon)^{k-p} (\overline{\varepsilon})^{n-k} \leq F(n-J_\beta, n-p, \overline{\varepsilon}) < \beta E_{\mathcal{C},0}(\delta) \tag{6.15}$$

Choosing J_β to satisfy (6.15) is equivalent to identifying J_β such that the ratio of the contribution to $E_{\mathcal{C},p}(\varepsilon)$ from indices greater than J_β to that of the BER $E_{\mathcal{C},0}(\delta)$ is no greater than β. Applying Azuma-Hœffding's inequality [2] to (6.15), we get

$$J_\beta = p + (n-p)\varepsilon' + \sqrt{\frac{(n-p)}{2} \log(\frac{1}{\beta E_{\mathcal{C},0}(\delta)})}. \tag{6.16}$$

Using J_β from (6.16) as the index where the second change of sign occurs, we can calculate an improved channel parameter for the parent code using the following.

$$j_\beta \log(\delta) - \overline{j_\beta} \log(\overline{\delta}) = \overline{j_\beta} \log(\overline{\varepsilon}) + (j_\beta - \frac{p}{n}) \log \varepsilon + (1 - \frac{p}{n}) h(\frac{J_\beta - p}{n-p}) - h(j_\beta), \tag{6.17}$$

where we used $j_\beta = \frac{J_\beta}{n}$. The unique solution for $\delta \in (0, \varepsilon)$ in (6.17) (denoted by $\tilde{\varepsilon}(\varepsilon)$) can be found in the same way as $\varepsilon^*(\varepsilon')$ is determined in the modified single sign-change bound. We, therefore, have the following improved lower bound.

Theorem 6.1.3. *(Adaptive Sign Change Bound) With the notations as above, for every code $\mathcal{C}$ of rate R,*

$$E_{\mathcal{C},p}(\varepsilon) \geq E_{\mathcal{C},0}(\tilde{\varepsilon}(\varepsilon)), \qquad 0 < \varepsilon \lesssim \frac{\overline{R}n - p}{n-p}. \tag{6.18}$$

6.2 Results of Simulation

In this section, we present our simulation results for the bounds of expected performance of punctured codes described in Section 6.1. LDPC codes of length 1000 were used in all the simulations. It must be noted here that the design rates of C1, C3 and C5 are 0.5. Just as in [93], the ensembles were selected so that the bounds are compared in both regular (C1) and irregular codes (C3 and C5), and in codes with the error floor property (C5) and in those without it.

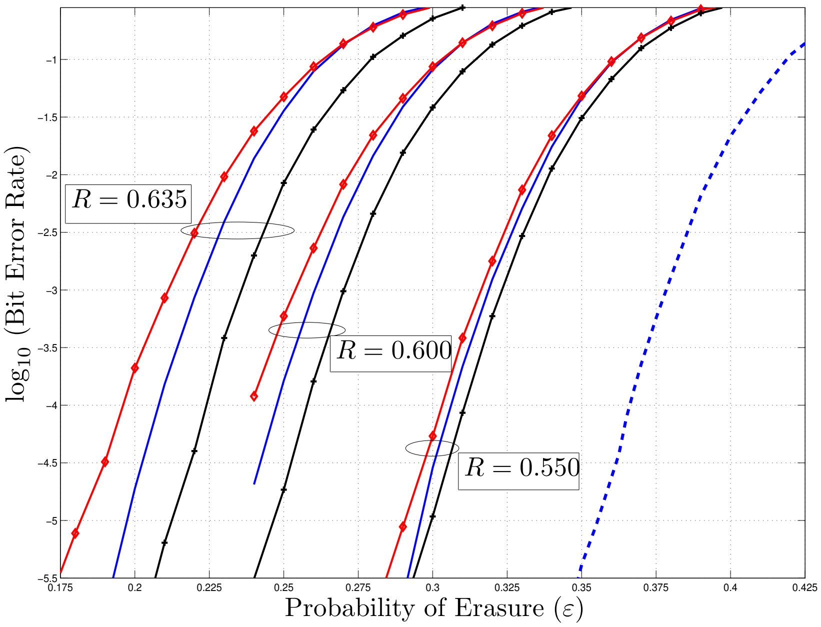

Figure 6.2: Results for the average BERs of randomly punctured codes of various rates derived from a code from the ensemble C1 over the BEC.

Figures 6.2 and 6.3 present the derived bounds for the performance over BEC for codes derived by a random puncturing of mother codes from C1 and C3 to rates ranging from 0.55 to 0.65. The performance of the mother code is presented using dotted lines at the far right of these figures. Also in these figures, the center curve at each rate represents

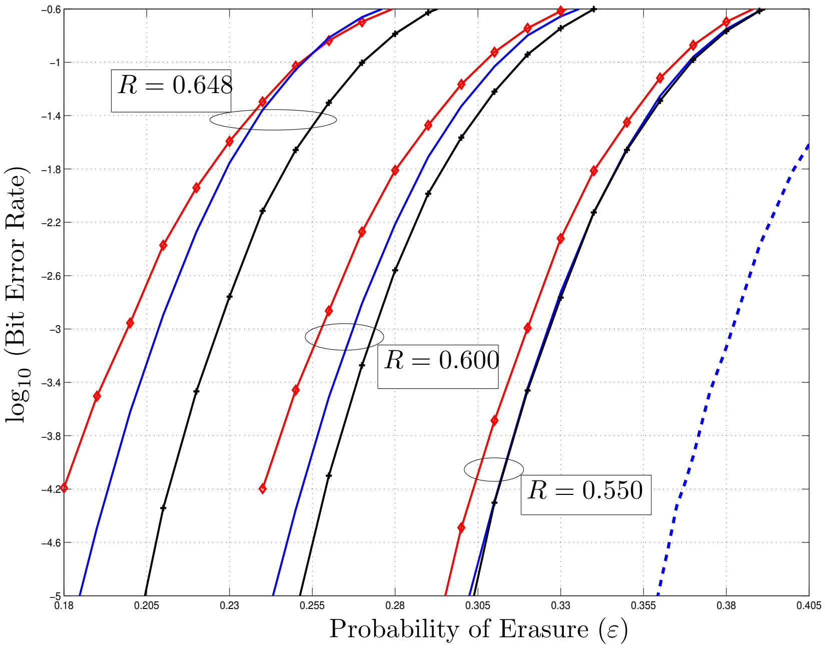

Figure 6.3: Results for the average BERs of randomly punctured codes of various rates derived from a code from the ensemble C3 over the BEC.

the simulation result for the expected performance of randomly punctured codes derived from the specific parent code of the corresponding ensemble. It can be noticed that at high erasure probabilities, the upper bound intersects the expected performance curve. This is attributable to the fact that the assumption that the convex region alone dominates the performance holds only for small erasure probabilities. Also, one notices that the effect of assumption of convexity all the way up to $\overline{R}n$ is a mere reduction in the range of validity of the upper bound. For example, the upper bound for a code of rate 0.635 from the ensemble C1 given by (6.7) is expected to be valid for channel erasure probability almost up to $\frac{\overline{R}n-p}{n-p} = 0.365$. However, by simulations we see that the region of validity for the upper bound is only up to 0.27. Further, it can be noted that for the actual channel parameter at which the upper bound and the actual performance curves intersect, the BER is of the order of 10^{-1}. Hence, one can safely use this bound at BERs smaller than 10^{-2}.

Figure 6.4 presents the comparison of the bounds and the simulations for the expected BER from random puncturing of a mother code from the ensemble C5 that exhibits an error floor. Notice that since the bounds use a scaled and shifted version of the mother code, the slope of the error floor for the bounds are different. However, the overall prediction and tracking using these bounds are fairly accurate. The lower bound presented in Figures 6.2, 6.4 and 6.3 is that of (6.18). Note that we use $\beta = 10^{-3}$ to generate our improved lower bound. It can be seen that both the upper and the lower bounds are tighter for higher channel erasure probabilities (say, those corresponding to BER of the order of 10^{-1}) than for lower channel erasure probabilities (say, those corresponding to the order of 10^{-3}). This can be attributed to the fact that, on the average, the performance of a punctured code obtained by randomly puncturing p bits over a BEC of very low erasure probability is very close to $\frac{S_p(\mathcal{C})}{n}$, the estimation of which is difficult.

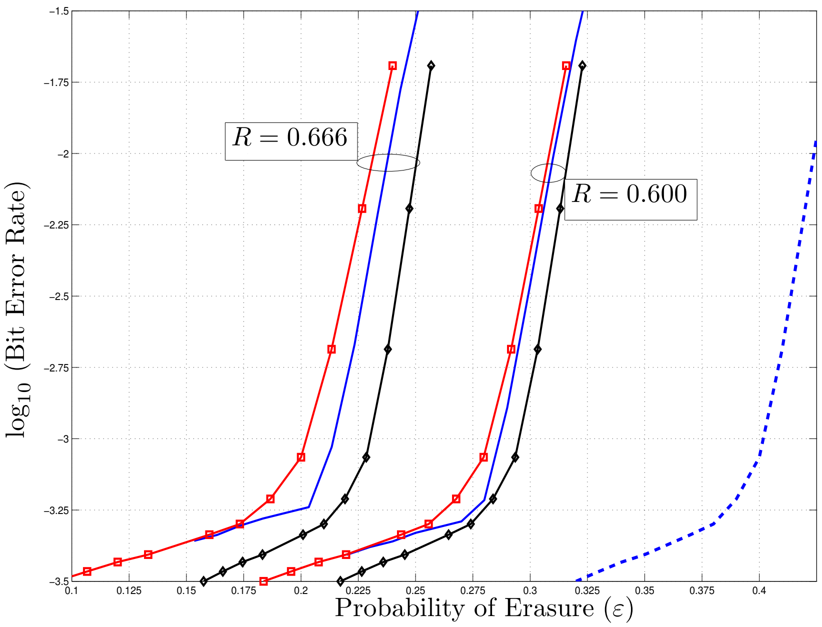

Figure 6.4: Results for the average BERs of randomly punctured codes of various rates derived from a code from the ensemble C5 over the BEC.

PART II

Reliable and Efficient Packet Delivery in Networks Via Modern Codes

CHAPTER VII

RELIABLE BROADCASTING IN MULTIHOP WIRELESS NETWORKS

The general approach to communication strategies in wireless networks is very different from that of wired network because the wireless property of the channels. Every packet that is transmitted by a node over a wireless link is heard by all the neighbors of the node. Therefore, the fundamental limits and the design of efficient algorithms in such networks are significantly different from that of wired networks. In this work, we consider the application of modern graph-based codes in the design of efficient broadcasting scheme in such networks[1].

This chapter is organized as follows. The following section presents the problem statement and the network model employed. The first half of Section 7.2 introduces the various stages proposed broadcasting scheme and then presents a discussion on the various overheads involved in the scheme. The latter half of the section discusses analytic bounds on the performance of the proposed scheme in grid and random deployment network settings. Finally, Section 7.3 presents the results of our simulations.

7.1 Problem Statement and Model Definitions

We investigate the design of an reliable, energy-efficient and low-complexity broadcasting algorithm that employs modern coding schemes in the setting of wireless multihop networks. We specifically focus on the design of a broadcasting algorithm when each node u knows its neighborhood information *i.e.*, the size and the members of its neighborhood. Throughout this work, we use the following model definitions.

[1] This work was done in collaboration with Dr. Nazanin Rahnavard.

7.1.1 Network Model

Consider a network of n static nodes with omnidirectional antennas and a transmission range of r units. Such a wireless network is modeled as a geometric undirected graph $G(V, E)$ with the set of nodes V and set of edges E. An edge connects two nodes u and v in G if and only if the Euclidean distance between nodes u and v is at most r. We also assume that r is large enough so that $G(V, E)$ is connected[2]. The neighborhood $N_r(u)$ of a node u is the set of all nodes that are within a distance of r units away from u. The wireless nature of the network is modeled by the assumption that any message that any node transmits is heard by all the nodes in its neighborhood. Also, we assume that the network modeled by the graph $G(V, E)$ has a unique source node s and $\mathbf{H} : V \longrightarrow \mathbb{Z}_{\geq 0}$ denotes the *hop-distance* function that maps each node v to the length of the shortest path connecting s to v. Finally, we assume that the network is equipped with a one-to-one node identifier function $id : V \longrightarrow \mathbb{N}$ such that each node v knows $id(v)$. For simplicity, we assume that id satisfies for any $u, v \in V$ and $u \in N_r(v)$, if u is closer to the source than v, then $id(u) < id(v)$.

To quantify the cost criterion for energy efficiency, we take an approach similar to [124]). We consider only the energy expended on RF transmissions. The cost for sending a packet from a node with transmission range r is taken to be r^2. We also denote the *number of transmissions per packet per node* by $\mathcal{N}_{/p/n}$ and the corresponding *energy consumption per packet per node* by $\mathcal{C}_{/p/n}$. Note that if m_v packets are losslessly sent by each node v with a transmission range r_v, then $\mathcal{N}_{/p/n} := \frac{1}{n} \sum_v m_v$, where as $\mathcal{C}_{/p/n} := \frac{1}{n} \sum_v m_v r_v^2$.

7.1.2 Simulation Setup

For simulations, we assume that the transmissions in networks to be subject to both distance attenuation and Rayleigh fading. Therefore, when a node u with a nominal transmission range r transmits, the signal-to-noise (SNR) of the signal received at a node v with distance d_{uv} from the node u is $\lambda r^2 / d_{uv}^{\alpha}$, where λ is an exponentially-distributed random variable

[2] Connectivity of a randomly deployed set of nodes occurs asymptotically almost surely if the radius of communication is chosen sufficiently larger than the threshold r^* described in [5, 89]

with unit mean and α is the attenuation parameter called *path loss*. The value of α is usually between 2 and 4 depending on the characteristics of the channel. However, in this work we set $\alpha = 2$. We assume that a packet transmitted by a node u is successfully received by a node $v \in N_r(u)$ if and only if the received SNR exceeds a threshold β, i.e., $\lambda r^2/d_{uv}^2 > \beta$. Also, we have $\beta = \frac{1}{2}$ for this whole volume of work.

In our simulations, we also consider a *medium access control* (MAC) scheme to prevent the collision of packets. Nodes that have packets to send contend to get access to the channel so that no collision occurs. By considering a MAC layer, we will also be able to compare the *latency* of the different broadcasting algorithms. Here, latency is defined as the time taken for all the nodes in the network to successfully decode all of the data sent by the source s. Here, a slotted CSMA scheme with mini-backoff similar to the one described in [8] is considered. In this scheme, each time unit has three parts – a contention period for *request-to-send* (RTS) transmissions, another for *clear-to-send* (CTS) transmissions and the other for actual packet transmission as described in Figure 7.1. The contention period is divided into M slots where M is greater than the average number of neighbors (this condition can be relaxed but will result in lesser nodes transmitting in each slot in different parts of the network). All nodes that want to transmit during the time period select a random slot from the RTS contention window to transmit their RTS. Each node transmits at a random slot in the CTS contention window in response to the first RTS it hears. The node that receives CTS from all its neighbors "wins" the right to transmit during the slot. Thus, one can generate a two-hop silence around each node that wins the slot.

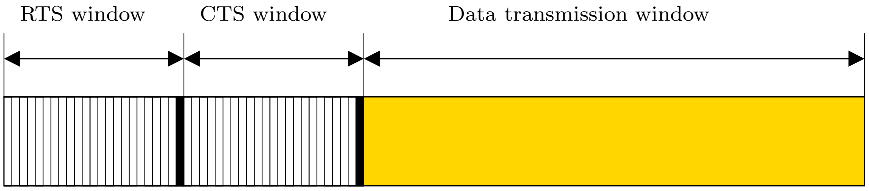

Figure 7.1: A time slot in slotted CSMA with mini-backoff.

For the purpose of both analysis and simulations, standard networks such as grid networks and randomly deployed networks are considered in this chapter. In grid networks, we consider two cases. In the first, the source is one of the four corners and in the second, the source is at the center of the grid. When the source is at a corner, the grid is assumed to be a square of l rows and l columns with neighbors spaced equally distant from each other. When the source is placed at the center, the network is assumed to be a square of $2l-1$ rows and $2l-1$ columns. In both scenarios, the transmission range r is chosen to be equal to the distance between any two neighboring nodes constraining the maximum degree of a node in the network to be four. In random deployment networks, the nodes are placed independently and uniformly at random in a field of $A \times A$ units with a transmission range that guarantees asymptotic connectivity with high probability.

7.2 FTS: Fractional Transmission Scheme

In this section, we propose a broadcasting scheme referred to as the *Fractional Transmission Scheme* (FTS). FTS is based on the idea that various neighbors of a node u can *share* the load of packet transmission to it. It suffices that each neighbor of a node just sends a fraction of the data such that the total sum of all fractions received by the node form its neighbors is enough for successful data recovery. However, packets from different neighbors should be innovative for this scheme to work. To ensure this, we employ rateless coding. First, the source encodes the data and forwards it. Other nodes, when they receive enough number of encoded data packets perform decoding and re-encoding to generate new packets.

It is assumed that each node is aware of its hop-distance from the source. Each node can also be aware of hop-distances of its immediate neighbors from the source by simple HELLO message exchanges. In FTS, between every pair of nodes that share a link, the one with the smaller hop-distance (from the source) transmits to the one with the higher hop-distance. In the case that both the nodes have the same hop-distance, the one with lower *id* transmits to the other. Therefore, a node v will expect to receive data from the nodes in its *parent set* $\mathcal{P}_r(v)$ that is defined as the following.

$$\mathcal{P}_r(v) = \{w \in N_r(v) : [\mathbf{H}(w) < \mathbf{H}(v)] \vee [(\mathbf{H}(w) = \mathbf{H}(v)) \wedge (id(w) < id(v))]\}. \tag{7.1}$$

On the other hand, each node w is responsible for providing a fraction of data to a set of nodes that are collectively called its *children* and are members of the set $\mathcal{C}_r(w) = N_r(w) \backslash \mathcal{P}_r(w)$. The method of determining the fraction α_w of data that each node w has to send to its children is described in the following subsection.

7.2.1 Description of FTS

FTS includes three phases: *Initial Fraction Exchange Phase*, *Fraction Reduction Phase*, and *Data Transmission Phase*. In the first two phases, each node determines the fraction of the data that it has to send, and the last phase is the actual phase where data is transmitted. In the *Initial Fraction Exchange Phase* (Algorithm 4), each node v determines the number of neighbors κ_v that will send data to v. In other words, $k_v = |\mathcal{P}_r(v)|$ represents the number of neighbors of v that either have a smaller hop-distance from the source or have the same hop-distance but a smaller *id*. Therefore, v expects a fraction of $\frac{1}{\kappa_v}$ of the required data from each of them. Once v determines the fraction it expects, it declares this fraction to nodes in $\mathcal{P}_r(v)$. Each node w collects all the expected fractions that it has to send to all its children $\mathcal{C}_r(w)$. It considers the maximum of these fractions as the sufficient fraction of data that it has to send. For example, if $\mathcal{C}_r(w) = \{u, v\}$ with $\kappa_u = 3$ and $\kappa_v = 4$, u and v expect one-third and one-fourth of the data from w, respectively. Then, $\alpha_w = \max\left(\frac{1}{3}, \frac{1}{4}\right) = \frac{1}{3}$. After this phase, the sum of the fraction of data that the neighboring parent nodes of a node u might be larger than one. Therefore, we can attempt to further reduce the fraction by employing the second phase called the *Fraction Reduction Phase* (Algorithm 5). In this phase, each node v asks its neighbors in $\mathcal{P}_r(v)$ to reduce their fractions equally by f_v that is

Algorithm 4 *Initial Fraction Exchange Phase*

```
Require: Graph $G(V, E)$ with source $s \in V$ where each node $v \in V$ knows its minimum hop-distance $\mathbf{H}(v)$ from
    source.
for $v \in V \setminus \{s\}$ do
    $\alpha_v = 0$.
end for
$\alpha_s = 1$.
for $v \in V$ do
    Transmit: $(\mathbf{H}(v), id(v))$
end for
for $v \in V$ do
    identify $\kappa_v = |\mathcal{P}_r(v)|$
    Transmit: $(v, \mathbf{H}(v), \kappa_v)$
end for
for $v \in V$ do
    for $w \in N_r(v) \setminus \mathcal{P}_r(v)$ do
        $\alpha_v = \max(\alpha_v, \frac{1}{\kappa_w})$
    end for
end for
for $v \in V$ do
    Transmit: $\alpha_v$
end for
```

Algorithm 5 *Fraction Reduction Phase*

```
for $v \in V$ do
    Transmit: $f_v = \max\left(\frac{1}{\kappa_v} \sum_{w \in \mathcal{P}_r(v)} \alpha_w - \frac{1}{\kappa_v}, 0\right)$
end for
for $v \in V$ do
    $\alpha'_v = \max(\alpha_v - \min_{w \in \mathcal{C}_r(v)} f_w, 0)$
end for
```

Algorithm 6 *Data Transmission Phase*

```
$send(s) = 1$
Transmit: $n_p\gamma$ encoded packets
for $v \in V \setminus \{s\}$ do
    $send(v) = 0, done(v) = 0$
end for
while $\exists\, v \in V$ with $send(v) = 0, done(v) = 0$ do
    $A = \{u \in V : send(u) = 0, done(u) = 0\}$
    for $w \in A$ do
        if $w$ has received $n_p\gamma$ encoded packets then
            Decode: $n_p$ data packets
            Transmit: $cACK(w)$
            Set $send(w) = 1$
        else
            if $w$ has received $\max(\alpha_z - f_w, 0)n_p\gamma$ packets from neighbor $z$ then
                Transmit: $pACK(w \to z)$
            end if
        end if
    end for
    $B = \{u \in V : send(u) = 1\}$
    for $t \in B$ do
        if all neighbors have either sent a $cACK$ or a $pACK$ directed at $t$ then
            Set $send(t) = 0$, $done(t) = 1$
        else
            generate an encoded packet $P$.
            Transmit: $P$
        end if
    end for
end while
```

calculated from the amount over and above the required fraction of data that v receives from its neighbors that send the fractions determined thus far. A node will reduce its fraction by the smallest of the requested reduction in fractions. The new fraction that node w will send is denoted by α'_w.

Figure 7.2 shows a small part of a network as an example. Suppose that $\mathcal{P}_r(u_1) = \{w_1\}$, $\mathcal{P}_r(u_2) = \{w_1, w_2\}$, and $\mathcal{P}_r(u_3) = \{w_2, w_3, w_4\}$, $\mathcal{C}_r(w_1) = \{u_1, u_2\}$, $\mathcal{C}_r(w_2) = \{u_2, u_3\}$, $\mathcal{C}_r(w_3) = \{u_3\}$, and $\mathcal{C}_r(w_4) = \{u_3\}$. Then, we have $\kappa_{u_1} = 1$, $\kappa_{u_2} = 2$, and $\kappa_{u_3} = 3$, and hence, $\alpha_{w_1} = 1$, $\alpha_{w_2} = \frac{1}{2}$, and $\alpha_{w_3} = \alpha_{w_4} = \frac{1}{3}$. By the fraction reduction phase, we have $f_{u_1} = 0$, $f_{u_2} = \frac{1}{4}$, $f_{u_3} = \frac{1}{18}$. The final fractions will therefore be $\alpha'_{w_1} = 1$, $\alpha'_{w_2} = \frac{1}{2} - \frac{1}{18} = \frac{4}{9}$, and $\alpha'_{w_3} = \alpha'_{w_4} = \frac{1}{3} - \frac{1}{18} = \frac{5}{18}$.

Once the first two phases are completed, we commence the *Data Transmission Phase*

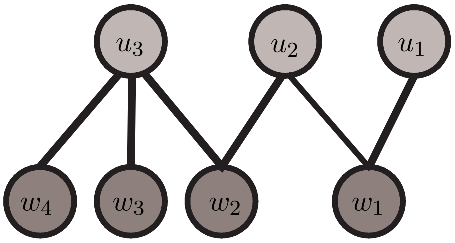

Figure 7.2: A small example.

(Algorithm 6). In this phase, once a node v receives fraction $\max(\alpha_w - f_v, 0)$ of packets from a neighbor w it sends acknowledgement of partial completeness ($pACK(v \rightarrow w)$), and when it becomes complete (receives enough number of encoded packets to decode), it sends acknowledgement of completeness ($cACK(v)$) to its neighbors. It then decodes and re-encodes the decoded data using a rateless code and sends the newly encoded packets until all its children are either complete or do not need any more packets from v.

7.2.2 Discussion of Various Overheads

The proposed scheme (FTS) has three overhead messages/signals whose effects on latency and transmission cost are described below. Since the network is assumed to be stationary, the setup overheads correspond to the *initial fraction exchange phase* and the *fraction reduction phase*, which have to be performed just once. In the former phase, the determination of the hop counts and the initial fractions can be initiated by the source by flooding a single packet. This can be completed in $\Theta(n)$ time units and by spending $\Theta(1)$ units of energy per node. In the latter, each node has to declare the amount by which its neighbors can deduct the fraction that it needs to send. This can again be performed in $\Theta(nr^2(n))$ time units by spending $\Theta(1)$ energy units per node.

The second overhead is that of coding. In general, the number of output packets required for giving a high probability of decoding n_p input packets is expressible as γn_p for a fraction $\gamma \geq 1$. The *coding overhead* γ corresponds to the number of additional packets that are required over and above the number of input packets so that successful decoding of the encoded packets happens with high probability of success. If one performs maximum likelihood (ML) decoding, one would have a smaller coding overhead. However, the computational complexity of ML decoding makes this option restrictive. On the other hand, using iterative decoding will enable lower decoding complexity but will have a higher coding overhead. Throughout this chapter, we set the coding overhead γ to be 1.03 for rateless codes for encoding $n_p = 2000$ packets as described in [73].

The third overhead is the bits that need to be appended to the header of an encoded packet to identify the index of the packets that were XORed to form the current packet. In rateless codes, at most $\Theta(\log_2 n_p)$ bits must be added to each output packet. The overhead imposed by these additional bits is considerably smaller than the corresponding overhead in random-sum coding, which is $n_p \log_2 q$ bits since in random-sum coding (also referred to as *network coding* (NC) in this chapter), each packet is multiplied by a coefficient chosen randomly from $GF(q)$ and these coefficients must be appended to the header of packets for decoding to commence at the decoder.

The last overhead and the most important overhead is that of acknowledgement of the completion or that of partial receipt from a particular neighbor. The former acknowledgement has to be transmitted exactly once. It must be noted that the second one is necessary only if we want a node to stop transmitting when all its children have received the required number of packets prior to finishing the transmission of its share of packets. In partial acknowledgment, we can achieve benefits in conserving energy while transmitting encoded data packets by allowing each node to transmit much smaller control packets. Note that $\Theta(nr^2(n))$ partial acknowledgements are sent totally and reduction in the size of each one of them results in considerable energy savings.

7.2.3 Analysis of FTS in Grid Networks

FTS is easily analyzed over grid networks and yields the following results on the average energy consumed. For simplicity, we first consider the case that all the links are lossless and then extend it to the case that considers signal attenuation over links and Rayleigh fading channel models.

7.2.3.1 Lossless Grid Networks

The following lemma presents the result on the average number of transmissions per packet per node for broadcasting over a grid network when the source is located at a corner of the grid.

Lemma 7.2.1. *For an $l \times l$ grid network with the source at a corner of the grid, the average number of transmissions per packet per node under FTS without the second stage and when a rateless code of overhead γ is used is given by*

$$\mathcal{N}_{/p/n} = \frac{(l^2 + 2l - 4)\gamma}{2l^2}.$$

Proof. Without loss of generality, we assume that the source node is in the bottom left corner of the grid. It is clear that the source node has to inject a fraction of one else no node in the network can perform decoding. Nodes at the bottom and left edges that are not corner nodes, which are $2(l-2)$ in number, have to send a fraction of one as their neighbors have $\kappa = 1$. The node at the top right corner transmits fraction of zero. All other nodes, which are $(l-1)^2 + 1$ nodes in number, transmit a fraction of $\frac{1}{2}$, since each of their neighbors have $\kappa = 2$. Therefore, the total number of transmissions per packet can be easily calculated as $\frac{(l^2+2l-4)\gamma}{2}$. ❑

One can verify that for such a grid network, the fraction reduction phase does not change the fraction of data for all but one of the nodes. This node (denoted as w) is at the position (1,1) assuming the source is at (0,0). The unique property of this node is its two children have other parents that will forward them all the data. Therefore, $\alpha'_w = \frac{1}{2} - \frac{1}{4} = \frac{1}{4}$. This change, however, is asymptotically negligible. Figure 7.3 illustrates the fractions of data that each node determines after the first phase of FTS in a 4×4 grid.

Assuming that each node has a transmission range r, the average cost of transmissions per packet per node is given by $\mathcal{C}_{/p/n} = \mathcal{N}_{/p/n} \times r^2$. From Lemma 7.2.1, we note FTS provides $\mathcal{C}_{/p/n} = \frac{\gamma r^2}{2}(1 + o(1))$.

Next, we would like to compare the cost of broadcasting with FTS and NC. If we fix the direction of each edge from the node with lower hop-count to the node with the higher

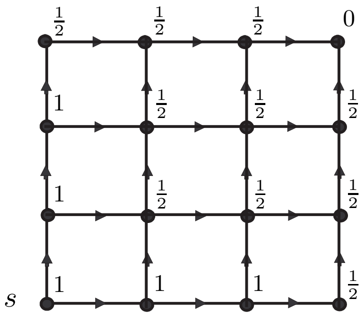

Figure 7.3: The fraction of data that each node sends to its neighbors after decoding and re-encoding in FTS. Source is at the bottom left corner.

hop-count, as is depicted in Figure 7.3, we see from the following result that FTS and NC offer the same asymptotic energy cost.

Lemma 7.2.2. *For an $l \times l$ grid network with the source at a corner of the grid, under NC (with the default direction that the node closer to the source always transmits to its neighbors that are farther away) we have $\mathcal{N}_{/p/n} = \frac{(l^2+l-2)}{2l^2}$.*

Proof. With the default direction, it can be seen that each node can potentially get data from at most two of its neighbors and transmit to potentially at most two neighbors. It can be seen that for the set of nodes in any diagonal[3] below the main diagonal of the grid, the fractions determined satisfy the conditions of Lemma B.1.1 of Appendix B. Similarly, for the set of nodes in any other diagonal, the fractions transmitted satisfy the hypotheses of Lemma B.1.2 of Appendix B. Combining both, one sees that, for an $l \times l$ grid,

$$\mathcal{N}_{/p/n} = \frac{1 + T_2 + \ldots + T_{l-1} + T_l^{'} + T_{l-1}^{'} + \ldots + T_2^{'}}{l^2} = \frac{(l^2 + l - 2)}{2l^2}.$$

❑

We can easily extend this directed grid analysis to the $(2l-1) \times (2l-1)$ grid when the source is at the center of the grid and directions on each edge are chosen to be directed from a node closer to the source to the one farther. In this case, $\mathcal{N}_{/p/n} = \frac{2l^2-2l+1}{(2l-1)^2}$.

From the above lemma, we note that, asymptotically, NC makes $\frac{1}{2}$ transmissions per packet per node for broadcasting in the assumed directed graph. It should be mentioned

[3] A diagonal is the set of all nodes with the same hop-length from the source and the main diagonal is the diagonal with the maximum size.

that the considered direction for the edges are not the best directions, and we can reduce the transmission by choosing another set of directions. The following theorem and the scheme defined thereafter present the most energy-efficient scheme that employs routing alone.

Theorem 7.2.1. *In an $l \times l$ grid network, the total number of transmissions per packet per node required for broadcasting k packets from any node to every node in the network by any routing scheme is at least $\frac{1}{3}$.*

Proof. Let X be a routing scheme. Let under the scheme X, v be an interior node (i.e., a node with four neighbors) in the grid that is scheduled to forward all the packets it receives. Let $w_1, \ldots, w_4$ be the neighbors of v and let $t_1, \ldots, t_4$ be the number of packets v receives from $w_1, \ldots, w_4$, respectively. Since all the nodes have to receive all the packets, totally $S = l^2 k$ packets need to be received by all the nodes. Since v must receive all the packets

$$t_1 + t_2 + t_3 + t_4 \geq k \tag{7.2}$$

Also, since v is scheduled to transmit all the packets, the contribution v makes to the sum S is given by the four individual contributions, namely, that from v to w_i, $i = 1, \ldots, 4$. Since w_i already has at least t_i packets the contribution to the sum S from the transmissions of v to w_i cannot exceed $k - t_i$. Thus, by transmitting k packets, v cannot make a contribution to S that is more than $4k - (t_1 + t_2 + t_3 + t_4) \leq 3k$. Also nodes of degree 2 cannot make a contribution to S that is greater than $3k$. Thus, any transmission in the network cannot benefit more than three nodes at a time and hence, the efficiency is at best $\frac{1}{3}$. ❑

This minimum $\mathcal{N}_{/p/n}$ of $\frac{1}{3}$ described in the above theorem can be asymptotically achieved for an $l \times l$ grid under the following scheme:

- $3|(l-1)$: Nodes in the bottommost row and in columns with index of the form $3j+1$ ($j = 0, 1, \ldots, \frac{l-1}{3}$) transmit a fraction of 1. Other nodes do not transmit. Figure 7.4 illustrates the case for a 7×7 grid. It can be shown that $\mathcal{N}_{/p/n} = \frac{1}{3} + \frac{4}{3l} - \frac{2}{3l^2}$.

- $3|l$: Nodes in the bottommost row (except the rightmost one) and those in columns with index of the form $3j+2$ ($j = 0, 1, \ldots, \frac{l}{3} - 1$) transmit the whole data. Other nodes do not transmit. It can be shown that $\mathcal{N}_{/p/n} = \frac{1}{3} + \frac{2}{3l} - \frac{1}{l^2}$.

- $3|(l-2)$: Nodes on the bottommost row except the last node and also nodes in columns with index of the form $3j+1$ ($j = 0, 1, \ldots, \frac{l-2}{3}$) transmit the whole data. Other nodes do not transmit. It can be shown that $\mathcal{N}_{/p/n} = \frac{1}{3} + \frac{1}{l} - \frac{4}{3l^2}$.

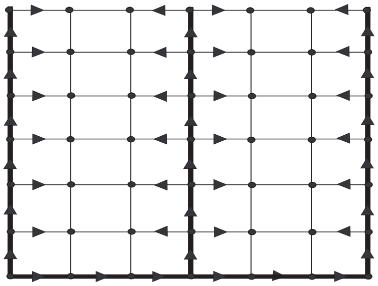

Figure 7.4: Optimal directions for a 7 × 7 grid. The nodes in the emboldened paths broadcast all the data and the other nodes remain silent.

With the same reasoning as above, for the directed grid with $(2l-1)^2$ nodes and with the source at the center, one can show that the optimal direction for this setting results in the following average energy cost.

$$\mathcal{N}_{/p/n} = \begin{cases} \frac{4l^2-7}{3(2l-1)^2} & 3|(l-2) \\ \frac{4l^2+4l-5}{3(2l-1)^2} & 3|(l-1) \\ \frac{4l^2+2l-9}{3(2l-1)^2} & 3|l \end{cases} . \tag{7.3}$$

7.2.3.2 Lossy Grid Networks

Here, we assume the more realistic case that the transmissions are subject to distance attenuation and Rayleigh fading as we described in Section 7.1. Assuming that the nodes have transmission range r and two neighboring nodes in a grid are apart by distance r, we have

$$\Pr[\text{A packet is obtained by its neighbor}] = \Pr[\lambda > \beta] = e^{-\beta}. \tag{7.4}$$

Therefore, each node v on the average needs to send fraction $\frac{\alpha'_v}{e^{-\beta}}$ instead of α'_v. We conclude that all $\mathcal{N}_{/p/n}$ and $\mathcal{C}_{/p/n}$ that we calculated for FTS in the lossless case, need to be scaled by factor $\frac{1}{e^{-\beta}}$ for this case. Since in this work, we have set $\beta = \frac{1}{2}$, we notice that for a pair of nodes, the effect of Rayleigh fading for the grid with neighboring nodes placed at a distance exactly equal to the transmission radius is a wireless erasure channel with an erasure probability of $1 - e^{-\frac{1}{2}}$.

7.2.4 Analysis of FTS on Lossless Randomly Deployed Networks

In this section, we analyze FTS presented in Section 7.2 to derive lower and upper bounds on the expected fraction of data α_v that each node v is expected to transmit at the end of the fraction exchange phase (Algorithm 4) assuming that there are no losses in the wireless medium. Both the lower and upper bounds use an estimation of κ_v for each node v to estimate the expected fraction to be transmitted by each node.

7.2.4.1 Lower Bound on the Performance of FTS

First, we notice that to derive a lower bound on the expected fractions, one must bound the number κ_v for each node $v \neq s$ from above. Each node v in its transmission radius of r distance units sees Y_v neighbors, where Y_v is a binomial random variable i.e., $Y_v \sim B(n-1, p \triangleq \frac{\pi r^2}{A^2})$. Thus, it is clear that v can listen to at most Y_v nodes. Thus $\kappa_v \leq Y_v$. Thus the fraction of data that any neighbor w of v expect to send to v is given by

$$E[\alpha_w] \geq E[\frac{1}{\kappa_v}] \geq E[\frac{1}{Y_v}] = \sum_{i\geq 1} \frac{1}{i}\binom{n-1}{i} p^i (1-p)^{n-1-i}. \tag{7.5}$$

Since direct estimation of the above expression is rather tedious, we resort to approximating the same using

$$\frac{1}{i} = \frac{1}{i+1} + \frac{1}{(i+1)^2} + \frac{1}{(i+1)^3} + \ldots \geq \sum_{j=1}^{L}\prod_{t=1}^{j}\frac{1}{i+t}, \qquad \forall L \in \mathbb{N}. \tag{7.6}$$

Using (7.6), we see that

$$\begin{aligned} E[\alpha_w] &\geq \sum_{i\geq 1}\sum_{j=1}^{L}\left(\prod_{t=1}^{j}\frac{1}{i+t}\right)\binom{n-1}{i}p^i(1-p)^{n-i-1} \\ &= \sum_{j=1}^{L}\left(\prod_{t=0}^{j-1}\frac{1}{(n+t)p}\right)\left[1 - \sum_{l=0}^{j}\binom{n+j-1}{l}p^l(1-p)^{n+j-1-l}\right]. \end{aligned}$$

It should be noted that the above lower bound for α_w is independent of the distance of w from the source s and hence averaging the bound over all the nodes in the network results in the same. Also, it can be seen that the lower bound gets better for larger values of the parameter L. However, the above bound assumes naïvely that every node receives from all its neighbors, which cannot occur in practice. The above result can be translated to the asymptotic setting in the following fashion.

Theorem 7.2.2. *The asymptotic cost of transmission per packet per node $\mathcal{N}_{/p/n}$ under FTS (without the second phase) over a random deployment network of n nodes with transmission radius sufficient for asymptotic connectivity is at least $\frac{1}{\log n}(1 + o(1))$.*

Proof. Please refer to Section B.2 of Appendix B. ❑

7.2.4.2 Upper Bound on the Performance of FTS

To estimate an upper bound on the expected fraction of transmission, one must find a lower bound on the number of parent nodes to which each node will listen. Consider the following scenario as illustrated in Figure 7.5, where a node v is situated geographically at point A at a distance D units away from the source at point S[4]. Suppose that $H(v) = i > 1$. The points in the transmission range of v that are closer to the source than v are those that lie both in the circle centered at point A and that centered at S. From simulations we see that with a very high probability $\mathbb{q}$ there is at least one node in the neighborhood of v with a hop-distance $i - 1$ from the source and is closer to the source than v. Figure 7.6

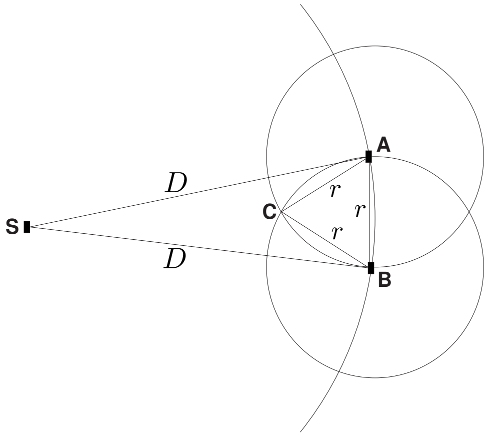

Figure 7.5: Illustration of $\Delta(r, D)$ used for deriving the lower bound.

[4] Without loss of generality, we may assume that $D > r$. The nodes in a circular region of radius r around the source are asymptotically a small fraction and hence their contribution to the transmission costs are asymptotically negligible.

presents the variation of the probability $\mathbb{q}$ that a node in an instance of random graph does not have a neighboring node closer to the source and with smaller hop-distance from the source. Simulations have been performed for $n = 500, 1000, 1500$ with varying transmission radii r starting from $r^* = 10, 8, 7.5$ units (refer Theorem 7.2.2), respectively. It is clear that this probability is monotonically decreasing with increasing n and r. We conjecture that asymptotically this event occurs for almost all nodes almost surely although we present only simulations to support the claim.

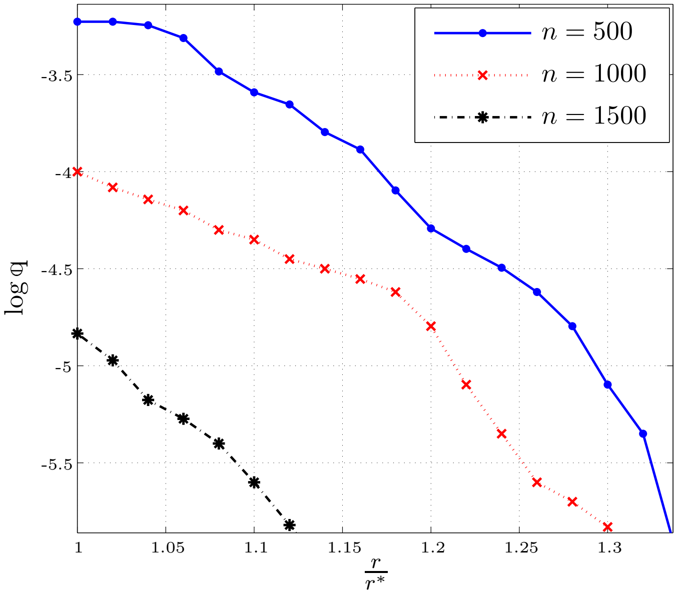

Figure 7.6: Variation of $\mathbb{q}$ with transmission radius for varying network sizes.

Therefore, assuming the above conjecture, for the aforementioned node v located at point A, we see that there exists a node $w \in N_r(v)$ that is closer to s than v and has a strictly smaller hop-distance. Then, all nodes that are located in the transmission range of both w and v and are closer to the source must have a hop distance of no more than i. Thus, the nodes in the area common to all the three circles (the circle centered at source passing through v and the circles of radius r centered at v and w, resp.) are expected to transmit to v under FTS. It can be shown that the area common to all the circles is minimum when w is located at B (*i.e.*, one of the two points of intersection between the two circles centered at A and S) as illustrated in Figure 7.5. Evaluation of this minimum area is a simple exercise in geometry and yields

$$\begin{aligned}\Delta(r,D) &\triangleq (\frac{\pi}{3}-\frac{\sqrt{3}}{4})r^2 + D^2\arcsin(\frac{r}{2D}) - \frac{r}{2}\sqrt{D^2-\frac{r^2}{4}} &(7.7)\\ &\geq (\frac{\pi}{3}-\frac{\sqrt{3}}{4})r^2 =: \underline{\Delta}(r). &(7.8)\end{aligned}$$

By the design of FTS, every node in this region with area $\underline{\Delta}(r)$ transmits to node v. Let X_v be the random variable denoting number of nodes in this region, i.e., $X_v \sim B(n-1, q = \frac{\underline{\Delta}(r)}{A^2})$. Thus, we see that $k_v \geq X_v$ and

$$E[\alpha_w] \leq E[\frac{1}{X_v}] = \sum_{i\geq 1}\frac{1}{i}\binom{n-1}{i}q^i(1-q)^{n-1-i}. \tag{7.9}$$

One can then use Chebyshev inequality [86] and (7.8) to conclude that

$$E[\alpha_w] = \frac{A^2}{nq}(1+O(\frac{1}{\sqrt{n}})) \leq \frac{A^2}{nr^2(\frac{\pi}{3}-\frac{\sqrt{3}}{4})}(1+O(\frac{1}{\sqrt{n}})). \tag{7.10}$$

Again, the lack of dependence of the bound in (7.10) on the parameter D follows from the independence of (7.8). Since (7.10) is independent of D, averaging over various nodes in different locations results in the following.

$$\mathcal{N}_{/p/n} \leq \frac{A^2}{nr^2(\frac{\pi}{3}-\frac{\sqrt{3}}{4})}(1+O(\frac{1}{\sqrt{n}})) \tag{7.11}$$

7.3 Results of Simulation

To compare the energy efficiency of FTS with present broadcasting algorithms such as Multipoint relaying (MPR) [96], Dominant Pruning (DPR) [63], Broadcast Incremental Power (BIP) [124], and Network Coding (NC) [69, 71], we simulated these algorithms on both square grids and randomly deployed networks. We used $\gamma = 1.03$ for rateless codes since it is sufficient for the transmission of $n_p \geq 2000$ packets with a decoding failure probability of less than 10^{-8} [73]. We considered the channel model and MAC scheme as explained in Section 7.1. The size of the packets are assumed to be the same in all the schemes. Also, it is worth noting that the amount of information payload per packet would be higher for FTS in comparison with NC as described in Section 7.2.2.

Schemes such as BIP and MPR are originally tailored for lossless networks, and they do not guarantee reliability in lossy networks. However, they can be extended to lossy networks by employing either multiple retransmissions or forward error correction over each link. In this way, if a channel has a corresponding loss probability of ϵ, on the average the number of transmissions will be scaled by a factor $\frac{1}{1-\epsilon}$.

7.3.1 Grid Networks

Simulations were performed for varying grid sizes to evaluate the transmission costs and latency of various schemes. Figure 7.7 depicts the transmission cost $\mathcal{C}_{/p/n}$ for different reliable broadcasting schemes for grid networks of varying sizes with source located at a corner. It is assumed that any two adjacent nodes have distance one unit and that the transmission radius $r = 1$. The transmissions are subject to distance attenuation and Rayleigh fading resulting in the probability of the successful delivery of a packet of $e^{-\frac{1}{2}}$. For FTS, both the simulation and analytical results as derived in Section 7.2.3 are depicted. It is noticed that the simulation and analytical results match very well. The cost of broadcasting using NC is also depicted. For large grid networks, the only difference in the cost of broadcasting between FTS and NC is the factor γ, the overhead of rateless coding. For NC, it is assumed that the random-sum coding is performed over a large

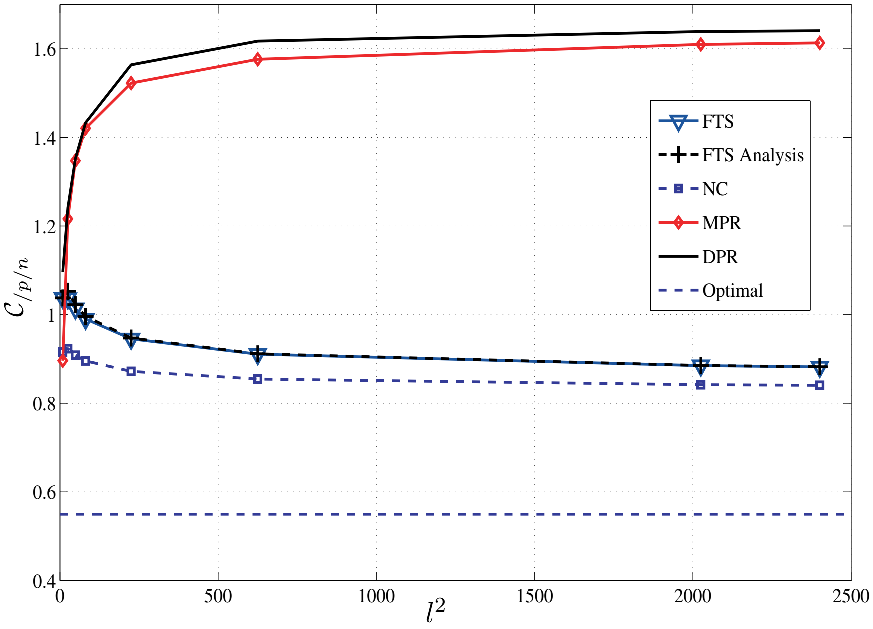

Figure 7.7: $\mathcal{C}_{/p/n}$ for various broadcasting schemes vs the size of the network for lossy $l \times l$ grids with the source at a corner.

field $GF(q)$ resulting in no coding overhead (but with higher decoding complexity). By increasing the number of packets n_p, γ approaches one and FTS and NC result in the same energy consumption per packet per node. The asymptotic optimal cost of $\frac{1}{3e^{-\frac{1}{2}}}$ obtainable from the scheme in Section 7.2.3 is also depicted in Figure 7.7. It should be noted again that the reason that NC and FTS do not attain optimality here is because the direction assignment for the edges is not optimal.

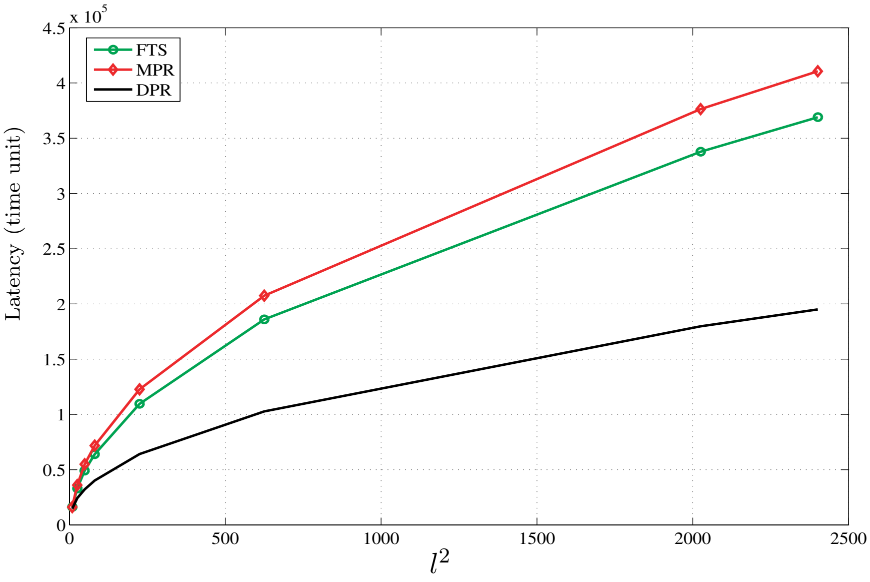

Figure 7.8: Latency for different schemes in lossy $l \times l$ grids with the source at a corner.

The latency of various schemes for a reliable delivery of 2000 packets in lossy $l \times l$ grids with the source at a corner is presented in Figure 7.8. The figure does not present the latency of NC since it is unclear as to how to integrate the MAC layer and interference with the random-sum coding scheme. Here, it must be noted that the results for energy consumption with any other MAC scheme (say, the ideal collision-free MAC with 2-hop silence) will be the same as in this slotted MAC with mini-backoff. However, the results for latency will differ depending on the MAC scheme. However, we expect NC to have a lower latency than FTS, since in FTS each node has to first wait to receive sufficient number of packets to be able to decode, before it can generate and send newly encoded packets.

7.3.2 Random Deployment Network

Here, we consider random deployment of n nodes each with transmission range r. The nodes are deployed in an area 100×100 square units with simulations performed for different values of n and r. The source is assumed to be at the center of the area. Further, the transmission range r for each network size is selected such that the resulting random graphs are connected with high probability. Also, the wireless transmissions are subject to attenuation due to distance and Rayleigh fading with parameter $\beta = \frac{1}{2}$.

First, we compare FTS and NC. For FTS, we can easily assign direction for edges based on their hop distances from the source, as explained in Section 7.2. We also use these directions in our simulations for NC. Table 7.1 compares the average $\mathcal{C}_{/p/n}$ for FTS and NC. Clearly, given the same graph and directions, NC has a lower cost, since it is optimal for the selected directions. From the table, we notice that the FTS consumes about 8% more energy than NC. However, FTS is a simple scheme that does not require any optimization. These features make FTS more flexible, practical and easy to implement.

Table 7.1: $\mathcal{C}_{/p/n}$ for random deployment networks of n nodes under FTS and NC.

n	r	Scheme	$\mathcal{C}_{/p/n}$
15	23	NC	331.10
		FTS	357.64
20	22	NC	326.04
		FTS	336.61
25	21	NC	300.29
		FTS	323.05

Next, we consider networks of up to 500 nodes. Due to the complexity of the linear programming module, evaluation of NC for such large networks was not performed. Therefore, for such large networks, the comparisons of FTS with only DPR, MPR, and BIP are presented. In the BIP scheme, different nodes can potentially have different transmission ranges to reduce the cost of broadcasting. Given the option that nodes can change their transmission range, we can extend FTS to FTS$_{adapt}$. FTS$_{adapt}$ is similar to FTS except that every node v in the network has the option of reducing its transmission range from r to r_v, where r_v equals the distance between v and the farthest neighbor of v that is going to listen to v (i.e., the farthest child of v when transmission range is r) in the data transmission phase (Algorithm 6). Thus, the probability that a child w of v receives a packet from v decreases, since we have

$$Pr\{w \text{ receives a packet from } v\} = Pr\{\lambda > \beta d_{vw}^2 / r_v^2\} = e^{-\beta d_{vw}^2/r_v^2} < e^{-\beta d_{vw}^2/r^2}. \qquad (7.12)$$

Therefore, by FTS$_{adapt}$ more number of transmissions is needed, however, each transmission has less cost. Overall, we expect that FTS$_{adapt}$ will be more energy-efficient than FTS. Figure 7.9 compares $\mathcal{C}_{/p/n}$ for different schemes FTS, FTS$_{adapt}$, MPR, DPR, and BIP. As

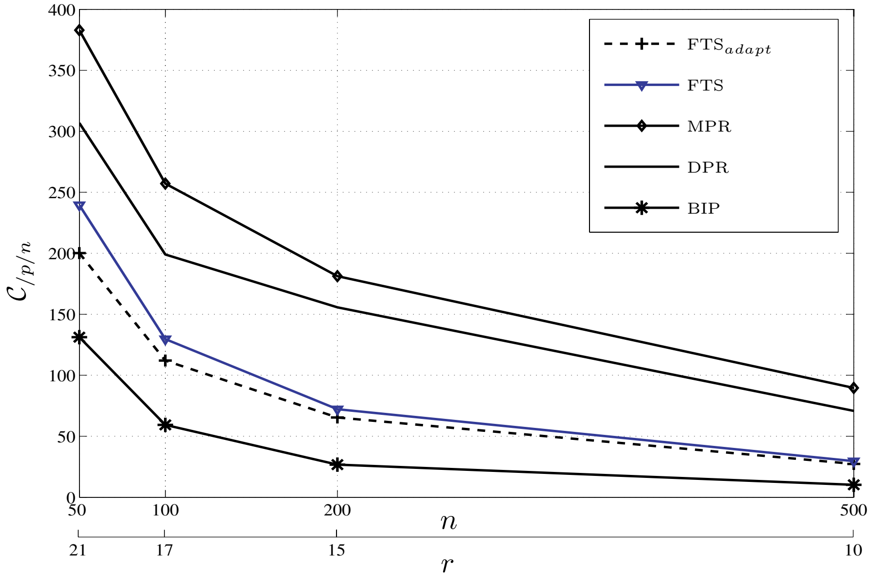

Figure 7.9: $\mathcal{C}_{/p/n}$ for different broadcasting schemes for random deployment networks.

we can see, BIP has the best possible performance. FTS$_{adapt}$ and FTS offer the next best performance. It should be noted that BIP is a centralized scheme that needs the global knowledge of the network and its complexity is very high. In contrast, FTS and FTS$_{adapt}$ are distributed schemes with low complexity. Comparing the performance of FTS$_{adapt}$ and FTS with the other two distributed schemes (DPR and MPR), we see that FTS and FTS$_{adapt}$ have much better performance. For large n, the improvement due to adaptive transmission range is small because of the fact that, on the average, there are $\Theta(\log n)$ nodes uniformly distributed in the neighborhood of each node. This makes the (Euclidean) distance d_{vw} between farthest node w in the neighborhood of a node v almost r.

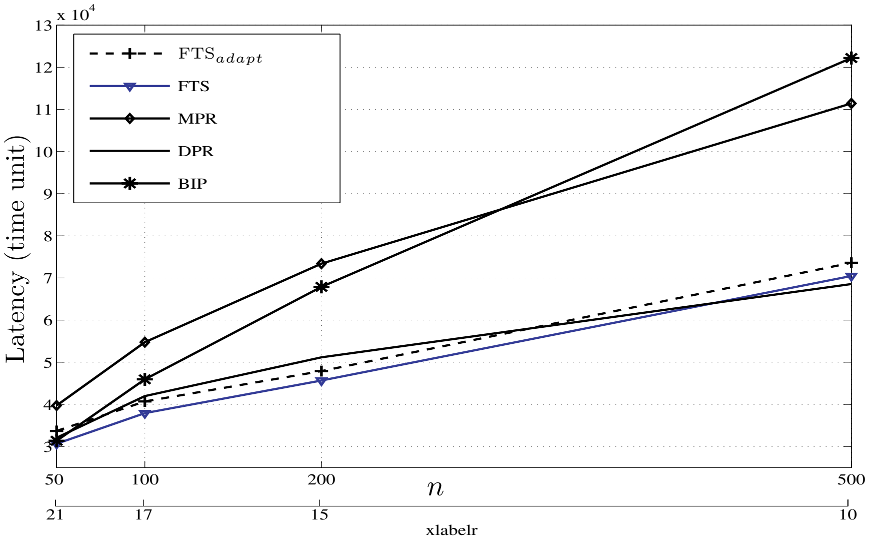

Figure 7.10: Latency for different broadcasting schemes for sending 2000 packets over random deployment networks.

Figure 7.10 depicts the latency of different broadcasting schemes. DPR seems to have the best latency whereas FTS and FTS$_{adapt}$ guarantee slightly worse latency performance. Among the two proposed schemes, FTS$_{adapt}$ has slightly larger latency than FTS. This is attributable to the fact that the number of transmission increases due to reduced transmission power when FTS$_{adapt}$ is implemented.

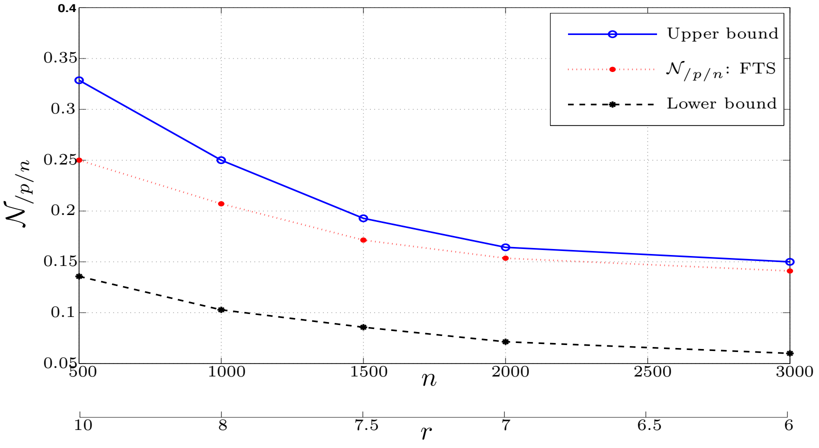

Figure 7.11: Analytical bounds for $\mathcal{N}_{/p/n}$ under FTS for random deployment networks.

In order to compare our analytical bounds derived in Section 7.2.4 with the actual performance of FTS, we simulated FTS (without Algorithm 5) for graphs of various sizes ranging from 500 to 3000 while assuming no channel losses. Figure 7.11 presents the derived upper bound of (7.11) and the lower bound of Theorem 7.2.2 in comparison to the result of simulations for FTS. Finally, it is noticed that the upper bound yields an accurate prediction of the actual cost of FTS for large graphs.

CHAPTER VIII

PROBABILISTIC BROADCASTING IN WIRELESS GRID NETWORKS

Probabilistic Broadcast (PBcast) is an energy-efficient scheme for data dissemination in Wireless sensor networks. Unlike flooding wherein all the nodes forward every received new packet, nodes forward the packets in a probabilistic fashion in PBcast. The problem of estimating the required probability of forwarding for energy-efficient and reliable broadcasting is challenging in grid networks. The analysis of PBcast in finite grid networks poses a challenge different from that posed by infinite grids. In this work, we analyze both directed and undirected grid networks using combinatorial path-enumerative techniques to derive bounds on the probability of successful receipt of packets at any particular location in the grid. We also present some iterative techniques that are useful in estimating the probability of receipt. With these results, we can effectively calculate the required probability of forwarding for PBcast at which both reliability and energy savings are effected.

This chapter is organized as follows. Section 8.1 defines the problem and presents the notations and terminologies that are used in this work. Section 8.2 presents the exact combinatorial formulation in addition to various techniques that bound the probability of successful receipt of a packet in a directed grid network. Section 8.3 extends the ideas to general undirected grid networks. We evaluate the results derived in Sections 8.2 and 8.3 in Section 8.4 via simulations.

8.1 Problem Definition and Terminologies

In PBcast, packets are forwarded in the network with a probability $p < 1$. However, since PBcast is not a reliable scheme, it is critical to study the range of values for the probability of forwarding p that guarantees sufficient reliability. The required probability for achieving reliability in randomly deployed networks in the asymptotic setting has been studied extensively using percolative techniques. However, these techniques fail to address the same issue for small networks. As an attempt to address this issue for finite networks, we study the problem of estimating the probability of forwarding that is required to ensure sufficient reliability (probability of receipt) at every node in a wireless grid network. We consider the following models to study this issue.

Throughout this work, by an $n \times n'$ directed grid, we mean a connected directed graph $G(V^d, E^d)$ where $V^d = \{0, \ldots, n-1\} \times \{0, \ldots, n'-1\}$. For two nodes $(i_1, j_1), (i_2, j_2) \in V^d$, $e^d = ((i_1, j_1), (i_2, j_2))$ is an edge iff

$$((i_2 - i_1 = 1) \wedge (j_2 - j_1 = 0)) \vee ((i_2 - i_1) = 0 \wedge (j_2 - j_1) = 1). \tag{8.1}$$

In other words, in a directed grid network, the nodes that are closer to the source can send data to the ones that are farther away. Also, due to the wireless nature of the medium, when a node transmits a packet, all nodes that are at the end of a directed link emanating from the node receive the same packet. Similarly, an undirected grid network of size $n \times n'$ is defined by an undirected graph $G(V, E)$ such that $V^d = \{-n+1, \ldots, n-1\} \times \{-n'+1, \ldots, n'-1\}$. However, the condition for edges between a pair of nodes $(i_1, j_1), (i_2, j_2) \in V$ is as follows. $e = \{(i_1, j_1), (i_2, j_2)\}$ is an edge iff

$$((|i_2 - i_1| = 1) \wedge (|j_2 - j_1| = 0)) \vee ((|i_2 - i_1|) = 0 \wedge (|j_2 - j_1|) = 1). \tag{8.2}$$

As an abuse of notation, we let (i, j) to denote both the location and the node at that location. Additionally, throughout this work the source node s that aims to disseminating data packets is assumed to be located at $(0, 0)$. However, our work can be generalized to other cases wherein the source node lies elsewhere. Lastly, we would like to add that the analysis of PBcast is carried out under the assumption that the wireless medium is lossless.

As in [115], we use the notation $\mathbb{R}[[x, y, z]]$ to denote the set of all formal power series whose coefficients are from $\mathbb{R}$. For a ring $\mathbb{E}$ and $E \in \mathbb{F}[[x_1, \ldots, x_n]], \mathcal{C}(E, \prod_{\alpha \in A} x_\alpha^{i_\alpha}) \in \mathbb{E}[[(x_\alpha : \alpha \in \{1, \ldots, n\} \setminus A)]]$ denotes the coefficient of $\prod_{\alpha \in A} x_\alpha^{i_\alpha}$ in $\mathbb{E}$. We use $\mathcal{B}(n, q, i)$ to denote the probability of receiving i heads in n Bernoulli trials of a coin whose probability of receiving heads in a single coin toss is q.

8.2 Analysis of PBcast on Directed Grid Networks

The main goal of this section is to analyze the dependence of the probability p_{ij} with which a node (i, j) will receive a packet from the source when the nodes employ PBcast with a forwarding probability p. This problem is essential in estimating the required forwarding probability while employing PBcast (or allied schemes such as *collaborative rateless broadcast* (CRBcast) [97]) so that we guarantee maximum energy savings in addition to the required reliability. Since in these algorithms, packets are forwarded in a hop-by-hop fashion, a node receives a packet only if all nodes in at least one of the paths from the source forward the packets successfully. Therefore, the problem can be identified as the estimation of the likelihood of a joint event that is the union of smaller events that are not necessarily (statistically) independent. The individual events represent the delivery of a packet via a particular path. Since this is true for both directed and undirected networks, path enumeration and the study of disjoint paths are a key ingredient in this work. A common approach to the estimation of the probability of a finite union of n events is to identify the probabilities of intersection of $k < n$ events. Such estimates include the extended Janson inequality [2], the de Caen bound [57] and the Dawson-Sankoff bound [95]. Although some generalizations of these bounds can be made to utilize information of joint events of any order $k < n$, in this chapter, we restrict our study to intersections of pairs of events. Before we proceed to the estimation of the higher order events, we present a discussion on the exact computation of the probability of receipt p_{ij} at the node (i, j).

8.2.1 Exact Computation of p_{ij}

In the directed grid, it can be seen that there are exactly $\alpha_{ij} = \binom{i+j}{i}$ paths from the source s to the node (i, j). If all the nodes on any of the paths from s to (i, j) transmit, (i, j) will receive the packet. If we enlist $W_1^{ij}, \dots, W_{\alpha_{ij}}^{ij}$ to be the distinct (but intersecting) paths from s to (i, j) and X_l^{ij} to be the events that all nodes in the path W_l^{ij} transmit the packet to its neighbors, we have $p_{ij} = P[\cup_{\kappa=1}^{\alpha_{ij}} X_\kappa^{ij}]$. Suppose that S represents the nodes that do not forward packets in a random instance of PBcast. Then, the packet will be successfully received if and only if there is a path from s to (i, j) that does not use any node in S. One can then use the combinatorial enumeration of the number of paths in [115] to conclude the following.

Theorem 8.2.1. *Consider an $n \times n'$ directed grid $G(V^d, E^d)$. Let for a subset of nodes $S = \{(i_\tau, j_\tau) : \tau = 1, \ldots, L\}$ define $M_S^{l,k}$ by the following matrix.*

1. $A_S \triangleq (l, i_1, \ldots, i_L)$, $B_S \triangleq (0, i_1, \ldots, i_L)$, $C_S \triangleq (k, j_1, \ldots, j_L)$, and $D_S \triangleq (0, j_1, \ldots, j_L)$.

2. $(M_S^{l,k})_{\iota\omega} \triangleq \binom{A_{S_\omega} - D_{S_\omega} + C_{S_\iota} - B_{S_\iota}}{A_{S_\omega} - B_{S_\iota}}$

Then, one can uniquely characterize the probability of receipt at any node (l, k) in an $n \times n'$ directed grid under PBcast with a forwarding probability of p is given by

$$p_{ij} = \sum_{S \subset V^d} p^{|S|}\, \overline{p}^{(nn' - |S|)}\, \mathbb{I}_{\mathbb{Z}_{\geq 0}}(\|M_S^{l,k}\|). \tag{8.3}$$

Proof. As outlined in [115], by comparing the direct expansion of the determinant with the expression obtained from the inclusion-exclusion principle, it can be shown that $\|M_S^{l,k}\|$ denotes the number of paths from $(0, 0)$ to (l, k) in the directed grid that does not pass through any node in S. The result follows directly from this fact. ❑

However, in Theorem 8.2.1, we notice that we perform more computational work than needed by unnecessarily calculating the determinant to identify the number of paths. We could simplify the computations by the following result.

Theorem 8.2.2. *Let for a directed grid with nodes V^d, $\varphi : 2^{V^d} \longrightarrow 2^{V^d}$ be defined such that for all $S \subset V^d$, $\varphi(S)$ is the smallest subset of V satisfying*

1. $S \subseteq \varphi(S)$, and

2. $\{(i + 1, j), (i, j + 1)\} \cap V^d \subset \varphi(S) \Rightarrow \{(i + 1, j + 1)\} \cap V^d \subset \varphi(S)$.

Then,

$$\mathbb{I}_{\mathbb{Z}_{\geq 0}}(\|M_S^{l,k}\|) + \mathbb{I}_{\varphi(S)}((l - 1, k))\mathbb{I}_{\varphi(S)}((l, k - 1)) = 1. \tag{8.4}$$

Proof. Let $S_0 = S$. Identify $S_{1\backslash 0}$, the set of all nodes (i, i') such that $\{(i - 1, i'), (i, i' - 1)\} \cap V^d \subseteq S_0$. Define $S_1 = S_0 \cup S_{1\backslash 0}$. Repeat the above for constructing a sequence of sets $S_0 \subseteq S_1 \subseteq S_2 \subseteq \ldots \subseteq V^d$. This sequence converges to $\varphi(S)$ in no more than $\max(n, n')$ iterations since the limit is the minimal set that satisfies the conditions in the definition. Assuming that S denotes the set of all nodes that do not forward, one can show by induction that every node in each S_i either does not receive the packet or does not forward it. If $(i-1,j),(i,j-1) \in \varphi(S)$, then there is no way that (i,j) will receive the packet since all paths in the directed graph pass through these nodes. The converse can be seen to be true since all nodes in $V^d \setminus \varphi(S)$ both receive the packet and forward it. Thus we see that the event that there exists a path from the source node to (l,k) and the event that $(l-1,k),(l,k-1) \in \varphi(S)$ are complementary and hence the result holds. ❑

Theorems 8.2.1 and 8.2.2 can be combined to evaluate the probability of receipt at any node by using equations (8.3) and (8.4). However, we note that even though for each subset of V^d, the work done is sub-linear in the number of nodes, the summation over a set that is exponential in the number of nodes makes the exact computation practically infeasible. The rest of this work is devoted to the estimation of the probability of receipt using combinatorial and iterative methods.

8.2.2 Combinatorial Estimation of p_{ij}

As outlined before, in this section, we will be looking at the problem of identifying the probability of receipt of a packet at the node (i,j) as the estimation of a problem of a union of events whose individual probability is known exactly. We contend ourselves with using information of the likelihood of events up to the second order, *i.e.*, the probabilities of pairwise intersection of events. The information obtained using the analysis will then be used in conjunction with various aforementioned bounds to extract meaningful inferences.

In order to identify terms up to the second order (pairwise intersection terms), one needs to evaluate 1. the number of paths, 2. the number of pairs of paths, and 3. the number of internally (node-)disjoint pairs of paths between the source and a given node (i,j). While the first two numbers are easily seen to be $\binom{i+j}{i}$ and $\binom{\binom{i+j}{i}}{2}$, the number of internally disjoint pairs of paths $l(i,j)$ between $(0,0)$ and (i,j) is not straightforward to compute. To evaluate the same, we define three generalized power series $\mathcal{L}, \mathcal{P}_1, \mathcal{P}_2 \in \mathbb{R}[[x,y]]$ in the following manner.

$$\begin{aligned}
\mathcal{L}(\epsilon, x, y) &= \sum_{(i,j)\in\mathbb{Z}^2_{\geq 0}} l(i,j)\epsilon^{2i+2j-1}x^i y^j, && (8.5)\\
\mathcal{P}_1(\epsilon, x, y) &= \sum_{(i,j)\in\{0,1,\ldots,n\}^2} \tbinom{i+j}{i}\epsilon^{i+j}x^i y^j, && (8.6)\\
\mathcal{P}_2(\epsilon, x, y) &= \sum_{(i,j)\in\{0,1,\ldots,n\}^2} \tbinom{\binom{i+j}{i}}{2}\epsilon^{i+j}x^i y^j, && (8.7)
\end{aligned}$$

where $\epsilon \in [0,1]$ is a parameter that would be used to take into consideration the probability of forwarding used in PBcast. Using the above framework, the coefficients $l(i,j)$ of $\mathcal{L}$ can be found by a simple deconvolution as described in the following theorem.

Theorem 8.2.3. *Let $\tilde{\mathcal{L}}(x,y) \triangleq \mathcal{L}(1,x,y)$. Similarly define $\tilde{\mathcal{P}}_1(x,y)$ and $\tilde{\mathcal{P}}_2(x,y)$. Then, the following holds:*

$$\tilde{\mathcal{P}}_1 = (2\tilde{\mathcal{P}}_2 + \tilde{\mathcal{P}}_1)(1 - 2\tilde{\mathcal{L}}\tilde{\mathcal{P}}_1). \tag{8.8}$$

Additionally, for $\mathcal{P}_1, \mathcal{P}_2$ given by (8.6) and (8.7), there is exactly one $\tilde{\mathcal{L}} \in \mathbb{R}[[x,y]]$ that satisfies (8.8).

Proof. A proof of this result can be found in Section B.3 of Appendix B. ❑

Having identified the number of internally disjoint pair of paths between the source and any node in the network, we now identify the pairwise joint probabilities. Consider the following sums.

$$\begin{aligned}
S^{(1)}_{ij}(p) &\triangleq \sum_{l=1}^{\alpha_{ij}} P(X^{ij}_l) = \alpha_{ij}p^{i+j-1} && (8.9)\\
S^{(2)}_{ij}(p) &\triangleq \sum_{l,l'=1}^{\alpha_{ij}} P(X^{ij}_l \cap X^{ij}_{l'}) && (8.10)
\end{aligned}$$

The above sums represent the union bound for p_{ij}, and the union bound for the event that two distinct paths deliver the packet. $S^{(2)}_{ij}(p)$ can be split into three separate sums. First, when l, l' are such that $X^{ij}_l, X^{ij}_{l'}$ are statistically independent (*i.e.*, the corresponding paths are internally node-disjoint). Next, when $l = l'$ and, third, when l, l' are distinct but $X^{ij}_l, X^{ij}_{l'}$ are not independent (*i.e.*, the corresponding paths intersect internally but are not identical). The former two are easy to identify and can be shown to be $l(i,j)p^{2i+2j-2}$ and

$\alpha_{ij}p^{i+j}$, respectively. The following theorem provides a method to identify $\delta^{ij}(p)$ – the third and the most tedious of the three terms described above.

Theorem 8.2.4. *Denote $X_\iota^{ij} \sim X_\kappa^{ij}$ when $X_\iota^{ij}, X_\kappa^{ij}$ are distinct and not statistically independent. Let $\delta^{ij}(p) = \sum_{X_\iota^{ij} \sim X_\kappa^{ij}} P(X_\iota^{ij} \cap X_\kappa^{ij})$. Define $\Delta(x,y) = \sum_{i,j \geq 0} \delta^{ij}(p)x^i y^j$. Then, $\Delta(x,y)$ is quantified uniquely by*

$$(\mathcal{P}_1(p,x,y) + 2p\Delta(x,y) + 2\mathcal{L}(p,x,y))(1 - 2\mathcal{L}(p,x,y)\mathcal{P}_1(p,x,y)) = \mathcal{P}_1(p,x,y). \qquad (8.11)$$

Proof. A proof of this result is presented in Section B.4 of Appendix B. ❑

Having evaluated the probabilities of the individual and union of pairs of events, we state our result that uses The Extended Janson Inequality [2] and The Dawson-Sankoff bound [95].

Result 1. *Using the notations defined previously, we let $S_{ij}^{(1)}(p) \triangleq \alpha_{ij}p^{i+j-1}$ and $S_{ij}^{(2)}(p) \triangleq \delta_{ij}(p) + l(i,j)p^{2i+2j-2} + S_{ij}^{(1)}(p)$, and $h \triangleq 1 + \lfloor \frac{2S_{ij}^{(2)}(p)}{S_{ij}^{(1)}(p)} \rfloor$. Then, using the aforementioned bounds, we can bound the probability of receipt p_{ij} for the node (i,j) in a directed grid as follows.*

$$\begin{aligned} P[\cup_{\kappa=1}^{\alpha_{ij}} X_\kappa^{ij}] &\geq \max\Big\{1 - e^{-\frac{(S_{ij}^{(1)}(p))^2}{4\delta^{ij}(p)}}, \frac{2}{h+1}[S_{ij}^{(1)}(p) - \frac{2}{h}S_{ij}^{(2)}(p)]\Big\}, & (8.12) \\ P[\cup_{\kappa=1}^{\alpha_{ij}} X_\kappa^{ij}] &\leq \min\Big\{1, S_{ij}^{(1)}(p) - \frac{2}{\alpha_{ij}}S_{ij}^{(2)}(p)\Big\}. & (8.13) \end{aligned}$$

The approach detailed above estimates the probability of receipt using well established techniques. In what follows, we present two simple iterative methods for tracking the probability of receipt that are based on the hop-distance from the source. The first iterative method classifies the nodes of the grid into stages based on the distance of the nodes from the source. It then bounds iteratively from above and below, the number of nodes that receive and forward packets in each stage. The second method uses the classification of nodes into stages to track the "average" distribution of the number of nodes that both receive and forward packets at each stage.

8.2.3 Iterative Bounds for p_{ij}

First, we notice that the nodes of the directed grid that participate in delivering packets to the node (i, j) are those whose x- and y-coordinates are at most i, and j, respectively. Since $p_{ij} = p_{ji}$, without loss of generality, we assume that $j \geq i$ throughout this portion of work. Consider the sub-grid $G_{ij}(V^{d,ij}, E^{d,ij})$ with all the nodes that participate in delivering packets to the node (i, j). Partition the set of nodes in the sub-grid into *stages* based on the hop-distance from the source. Note that each stage can be indexed by the hop-distance of its element nodes. Further, we can categorize the stages into three types. These types represent the increasing (I), stable (S) and the decreasing (D) portions of the sub-grid and are pictorially represented in Figure 8.1. As is clear, a stage of all nodes with hop-distance h is of the I-type if the number of nodes with hop-distance $h + 1$ is one more than that of hop-distance h. Similarly, a stage is said to be of the S- or D-type if the size of the next stage is the same or smaller than itself, respectively.

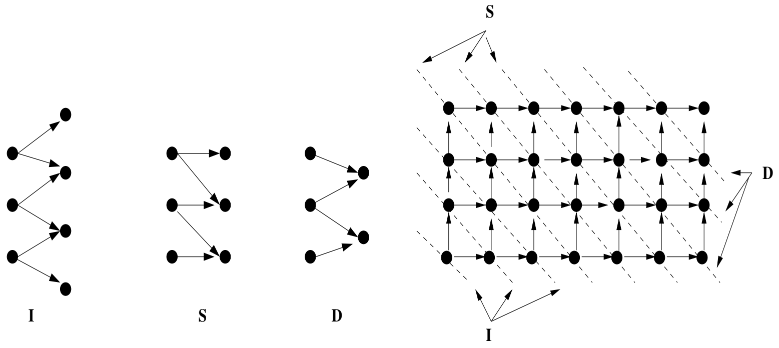

Figure 8.1: An illustration of categories of stages in a directed grid.

In this approach, we track the distribution of the number of nodes that have received the packets and are going to forward packets to its neighbors in a iterative, stage-by-stage fashion. The key element here is to define a pair of transition matrices for each hop-distance. These matrices translate a vector whose components denote bounds on the probability of receipt of a packet by a subset of nodes that have the same hop-distance h and provide an upper and lower bound on the probability of receipt for each node in the set of nodes with the hop-distance $i + 1$. The process is then repeated iteratively until the estimate for the node (i, j) is obtained. We define these transition matrices for each hop-distance between

2 and $i+j-1$ in the following manner.

For $1 \leq l < i$ (an I-type stage) and $p \in (0,1)$, define $\mathcal{T}^{*ij}_{\;l}, \mathcal{T}_{*l}^{ij}$ to be $(l+2) \times (l+3)$ matrices in the following manner.

$$
\begin{aligned}
(\mathcal{T}^{*ij}_{\;l})_{st} &= \mathcal{B}(\min(2s-2, l+2), p, t-1). && (8.14)\\
(\mathcal{T}_{*l}^{ij})_{st} &= \mathcal{B}(s,p,t-1)(1-\delta[(s-2)(s-1)]) + \mathcal{B}(1,p,t-1)\delta[s-2] + \delta[s-1]\delta[t-1]. && (8.15)
\end{aligned}
$$

For $i \leq l < j$ (an S-type stage) and $p \in (0,1)$, define $\mathcal{T}^{*ij}_{\;l}, \mathcal{T}_{*l}^{ij}$ to be $(i+2) \times (i+2)$ matrices as

$$
\begin{aligned}
(\mathcal{T}^{*ij}_{\;l})_{st} &= \mathcal{B}(\min(2s-2, l+1), p, t-1). && (8.16)\\
(\mathcal{T}_{*l}^{ij})_{st} &= \mathcal{B}(s-1, p, t-1). && (8.17)
\end{aligned}
$$

Finally, for $j \leq l < i+j$ (an D-type stage) and $p \in (0,1)$, define $\mathcal{T}^{*ij}_{\;l}, \mathcal{T}_{*l}^{ij}$ to be $(i+j+2-l) \times (i+j-l+1)$ matrices as follows.

$$
\begin{aligned}
(\mathcal{T}^{*ij}_{\;l})_{st} &= \mathcal{B}(\min(2s-2, i+j-l+1), p, t-1). && (8.18)\\
(\mathcal{T}_{*l}^{ij})_{st} &= \mathcal{B}(s-1,p,t-1)\delta[s-(i+j+2-l)] + \mathcal{B}(s-2,p,t-1)(1-\delta[s-(i+j+2-l)]). && (8.19)
\end{aligned}
$$

With the transition matrices as above, one can identify upper and lower bounds for the probability of receipt as described in the following manner. Although it is not detailed here, it can be shown that the upper bound is tight as p approaches unity.

Theorem 8.2.5. *(Iterative Bounds) Define* ${V_*}^{(2)} = V^{*(2)} = [\overline{p}^2 \quad 2p\overline{p} \quad p^2]$. *Define for* $k = 3, \ldots, i+j$, $V^{*(k)} = V^{*(k-1)}\mathcal{T}^{*ij}_{\;k-1}$ *and* ${V_*}^{(k)} = {V_*}^{(k-1)}\mathcal{T}_{*k-1}^{ij}$. *Then,*

$$
{V_*}_2^{(i+j)} \leq p_{ij} \leq {V^*}_2^{(i+j)}. \tag{8.20}
$$

Proof. First, we note that ${V^*}_i^{(2)}$ is the probability that $i-1$ nodes out of the two nodes at unit hop-distance forward the packets. The proof is complete if we show that the matrices $\mathcal{T}^{*ij}$, and ${\mathcal{T}_*}^{ij}$ overestimate and underestimate, respectively, the number of nodes active at each stage. Here, a stage is the subset of nodes in G_{ij} that have the same hop-distance. Note that in each of these matrices, the $(s,t)^{\text{th}}$ coordinate at the l^{th} stage represents the probability of transition in the event that $(s-1)$ nodes at the l^{th} stage are active (i.e, have received the packet and are going to forward it) and $(t-1)$ nodes are active in the $l+1^{\text{th}}$

stage. For hop-distances strictly smaller than i, one can notice that if $(s-1)$ nodes are active, then they have at most $\min(2(s-1), l+1)$ neighboring nodes at the next stage. Out of these $\min(2(s-1), l+1)$ nodes, a random subset of nodes will be active. The probability distribution of the number of active nodes at the $l+1^{\text{th}}$ stage is binomial with parameters $n = \min(2(s-1), l+1)$ and probability p. Therefore, one concludes that given an upper estimate on the number of active nodes at stage $l < i-1$, one can get a probability distribution that will provide an upper estimate of the number of active nodes at the next level. For stages $i \leq l < j$, one sees that $s-1$ nodes have at most $\min(2(s-1), i)$ neighbors at the next stage. Similarly, one notices that for stages $l \geq j$, $s-1$ nodes have at most $\min(2(s-1), i+j-l+1)$ neighbors at the next stage. The argument for the upper bound is then complete by a simple inductive argument that at each stage l, ${V^*}^l_j$ provides an upper bound on the probability that exactly j nodes of the l^{th} are active. The argument for lower bound follows similar reasoning. ❑

8.2.4 Average Tracking Estimate of p_{ij}

Since tracking the exact distribution of the number of active nodes at each stage is computationally intensive, another approach for estimating the probability of receipt at any given node in a directed grid is by tracking just the "average" distribution of the number of nodes that both receive and forward packets at each stage. Just as in the case of the Iterative Estimate where the transition matrices were organized into three categories, we categorize the stages here into three types. Again, these stages represent the increasing (I), stable (S) and the decreasing (D) portions of the grid network and are pictorially represented in Figure 8.1.

By a simple combinatorial enumerative technique based on power series, one can show the following result. Suppose that at a stage with l nodes, a random subset of i nodes[1] is active. Then, the probability that exactly k nodes receive the packet from these i active

[1] Here, it is assumed that every subset of size i out of the l nodes is equiprobable.

nodes is given by

$$P_{k|i} = \begin{cases} \frac{\binom{i-1}{2i-k}\binom{l-i+1}{l-k+1}}{\binom{l}{i}} & \text{Stage is of Cat. I} \\ \frac{\left[\binom{i-1}{2i-k}+\binom{i-1}{2i-1-k}\right]\binom{l-i}{l-k}}{\binom{l}{i}} & \text{Stage is of Cat. S} \\ \frac{\left[\binom{i-1}{2i-k}+2\binom{i-1}{2i-k-1}+\binom{i-1}{2i-2-k}\right]\binom{l-i-1}{l-k-1}}{\binom{l}{i}} & \text{Stage is of Cat. D} \end{cases} \tag{8.21}$$

To estimate the probability of receipt at a node (i, j), we start with the distribution $\mathcal{D}_1 = [\overline{p}^2 \quad 2p\overline{p} \quad p^2]$ of active nodes at the stage of nodes that are a single hop away from the source. Using this, we calculate the "average" distribution of the number of nodes (of the next stage) that will receive the packet from the active nodes of this stage by using (8.21). The distribution for the next stage $\mathcal{D}_2$ is obtained by multiplication of the distribution of the average number of receiving nodes with the "binomial" matrix $[\mathcal{B}(i, p, j)]_{i,j=1}^{4}$. Note that the multiplication takes care of the probabilistic forwarding aspect of PBcast. This is then repeated iteratively each time with the appropriate transition rule of (8.21) and with bigger "binomial" matrix until the distribution for the stage containing (i, j), i.e., $\mathcal{D}_{i+j}$ is obtained. For the above average tracking approach, the following quantitative results can be shown.

1. The average tracking method yields an upper bound for p_{ij} as $p \downarrow 0$.

2. As $p \uparrow 1$, the Average Tracking Estimate for a node (i, j) approaches p_{ij}. Hence, this technique provides accurate estimation of probability of receipt for $p \approx 1$.

8.3 Extension to Undirected Grid Networks

In this section, we outline the extensions of the above ideas to undirected grid networks. The analysis for such networks needs special attention because the undirected nature of the links allow many more parallel paths between the source and any particular node in the network. For instance, in an infinite undirected grid, there are infinitely many paths between the source and the node $(1, 1)$. However, the number of paths of each length is bounded.

To extend the estimates using Janson's inequality and Dawson-Sankoff bound, we need to identify the number of pairs of paths that are internally node-disjoint between the source and any node (i, j). Just as in the case of directed grids it can be shown that the number of paths (including self-intersecting ones) between $(0, 0)$ and (i, j) of length $i+j+2r$ in a finite grid is bounded above by $\sum_{l=0}^{r} \binom{i+j+2r}{i+l, j+(r-l), l, (r-l)} = \binom{i+j+2r}{r}\binom{i+j+2r}{i+r}$ and the corresponding power series is given by

$$P_1'(x, y, z) \triangleq \sum_{i,j,k \geq 0} \binom{i+j+2k}{k} \binom{i+j+2k}{i+k} x^i y^j z^{i+j+2k}. \tag{8.22}$$

The reason that this is an upper bound for the number of paths is that this calculation ignores the finiteness of the grid. In fact, this calculation is exact for an infinite grid. The computation for the number of paths that do not intersect with itself is performed as follows.

Lemma 8.3.1. *Let the power series P_1' be defined as in (8.22). Then, the number of paths of length l between s and (i, j) that are not self-intersecting can be identified from the $x^i y^j z^l$ coefficient of P_1 defined below.*

$$P_1 = (1 + CP_1'(C+1)^{-2})^{-1} P_1'(C+1)^{-2} \tag{8.23}$$

where $C(x, y, z) \triangleq \sum_{i=1}^{\infty} \binom{2i}{i}^2 z^2$.

Proof. Just as in the proof of Theorem 8.2.3, one can count paths along with their lengths and associate all paths between the source and a particular node (i, j) and of a particular length l as coefficients of $x^i y^j z^l$. In doing so, we notice that

$$\begin{aligned} P_1' &= (P_1 + P_1C + P_1CP_1 + \ldots) + (CP_1 + CP_1C + CP_1CP_1 + \ldots) && (8.24)\\ &= (C+1)(P_1 + P_1C + P_1CP_1 + \ldots) && (8.25)\\ &= (C+1)(P_1(1 + CP_1 + (CP_1)^2 + \ldots) + P_1(1 + CP_1 + (CP_1)^2 + \ldots)C) && (8.26)\\ &= (C+1)^2 P_1(1 + (P_1C) + (P_1C)^2 + \ldots). && (8.27) \end{aligned}$$

Simplifying the above expression, one obtains the result. ❑

Having identified P_1, the proper extension of the power series in this setting can be shown to be

$$P_2(x, y, z) = \sum_{i,j \geq 0} \{\frac{(P_1^{i,j}(z))^2 - P_1^{i,j}(z^2)}{2}\} x^i y^j \tag{8.28}$$

where $P_1^{i,j}(z) \in \mathbb{R}[[z]]$ is the power series that is the coefficient of $x^i y^j$ in P_1' when P_1' viewed as a power series in $(\mathbb{R}[[z]])[[x,y]]$. With an argument similar to that for (8.8), one can show that the number of pairs of paths that are each non self-intersecting and that are internally node-disjoint are given by

$$P_1 = (2P_2 + P_1)(1 - 2LP_1). \tag{8.29}$$

Let $W_1^{ij}, W_2^{ij}, \ldots$ be an enumeration of the various paths from s to (i,j) and let Y_k^{ij} denote the event that path X_k^{ij} delivers the packet. With this notation, we need to redefine $L(x,y,z)$ as $\frac{L(x,y,z)}{z}$ so that we can compute the joint probability term $\sum_{i,j} P(Y_i \cap Y_j)$. Just like Theorem 8.2.4, one can then show the following result.

Theorem 8.3.1. *Let $Y_\iota^{ij} \sim Y_\kappa^{ij}$ if $Y_\iota^{ij}, Y_\kappa^{ij}$ are distinct and not independent and let $\pi^{ij}(p) = \sum_{Y_\iota^{ij} \sim Y_\kappa^{ij}} P(Y_\iota^{ij} \cap Y_\kappa^{ij})$. Define $\Pi(x,y) = \sum_{i,j \geq 0} \pi^{ij}(p) x^i y^j$. Then, Π is quantified uniquely by*

$$(P_1(x,y,p) + 2p\Pi(x,y) + 2L(x,y,p))(1 - 2L(x,y,p)P_1(x,y,p)) = P_1(x,y,p) \tag{8.30}$$

Lastly, one can use the upper bound of (8.12) with

$$S_{ij}^{(1)} = \mathscr{C}(P_1, x^i y^j), \tag{8.31}$$

$$S_{ij}^{(2)} = \pi^{ij} + \mathscr{C}(L(x,y,p), x^i y^j) + \mathscr{C}(P_1(x,y,p), x^i y^j) \tag{8.32}$$

to derive an upper bound for the undirected grid. However, the lower bound is inapplicable since it ignores the finite boundary and hence considers paths that exist in the infinite grid but not in the finite one. However, in practice, when the node for which the probability of receipt is calculated is fairly distant from the boundary, one expect the lower bound to be valid.

The extension of the average tracking method for undirected grids can be done as follows. Since in the undirected grid, packets can be received even from farther nodes, we must allow for iteration using distribution from higher stages to update the lower stages and vice versa. Initially, we proceed just as in the case for directed grids. Once we reach the last stage, we proceed back from the stage containing the farthest nodes to the closest stage, i.e., the stage containing the source node, one stage at a time. These back and forth iterations are repeated multiple times until the distributions converge. Note that for each stage i (other than the last and the first one), the update is calculated from the estimated distribution of active nodes in the earlier stage $i-1$ and the next stage $i+1$, again using the assumption that all subsets of the same weights are equiprobable (as in the case of directed grids). Unfortunately, for undirected grids, unlike the directed case, there is quantitative result regarding the comparison of the estimate using this tracking method and the actual probability of receipt. However, the tightness of the tracking scheme still holds for large probabilities of forwarding just as in the case of directed grids.

8.4 Results of Simulation

In this section, we present the results of the derived bounds in comparison to the results of simulations. First, we present the results for directed grids and then present those for directed grids. Figures 8.2 and 8.3 present the upper and lower bounds for directed grids for nodes located at $(20, 20)$ and $(100, 100)$ for various probabilities of forwarding. The two upper bounds derived in the previous section and the minimum of the two upper bounds are presented in these figures, while only the maximum of the two lower bounds is shown.

In both these figures, it can be noted that the Average Tracking Estimate tracks the variation of the actual probability of receipt reasonably well compared to the other bounds. It can be seen that the upper bound derived from $S_{ij}^{(1)}$ and $S_{ij}^{(2)}$ shows a sharp threshold effect at $p = 0.5$, implying that for large lengths, a probability of at least 0.5 is necessary to expect any reasonable probability of receipt. Also, from the average tracking estimate curve, we notice that we require $p \approx 0.82$ and $p \approx 0.85$ to guarantee a probability of receipt of about 0.9 for locations $(20, 20)$ and $(100, 100)$, respectively. These estimates provide an accurate estimate of the required forwarding probability for PBcast to ensure high reliability. Lastly, we note that the lower bound although is inaccurate for low probabilities of forwarding, it also approximates the behavior of the network very well for high probabilities of forwarding.

Figure 8.4 presents the variation of the actual probability of receipt of packets with location in the directed grid for two probabilities of forwarding $p = 0.6$ and $p = 0.8$, respectively. The figure presents the variation of the probability of receipt over the diagonal

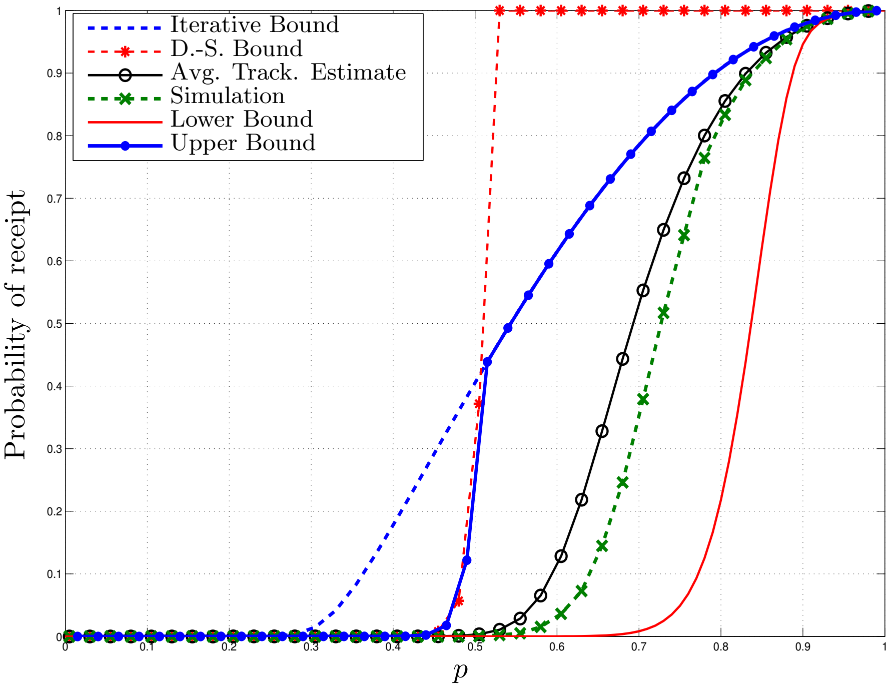

Figure 8.2: The probability of packet receipt at $(20, 20)$ vs the forwarding probability p for a directed grid.

(i, i) with the index i. It can be noted that for the lower probability, the lower bound (i.e., the maximum of the two derived bounds) is fairly close in estimating the probability of receipt whereas at higher probability, the average tracking estimate is close. Interestingly, at high probabilities of forwarding (say at $p = 0.8$), we notice that the variation of probability of receipt with the location (i, i) reaches an almost steady state-like behavior. This can also be inferred from Figure 8.5. Notice that for very high probabilities, the difference in the simulated curves for the actual probability of receipt is minor for large probabilities of forwarding. For both the simulation and the Average Tracking Estimate, we notice that the variation of probability of receipt for high ps are minute, whereas the difference is large for probabilities up to $p = 0.75$. We therefore see a flattening of the curve in Figure 8.4 for $p = 0.8$. Additionally, since the difference in the simulated curves can be noticed for $p < 0.75$, we expect the flattening to be true only over the range $p > 0.75$. However, this is not surprising since at reasonably high probabilities of forwarding, the probability of receipt

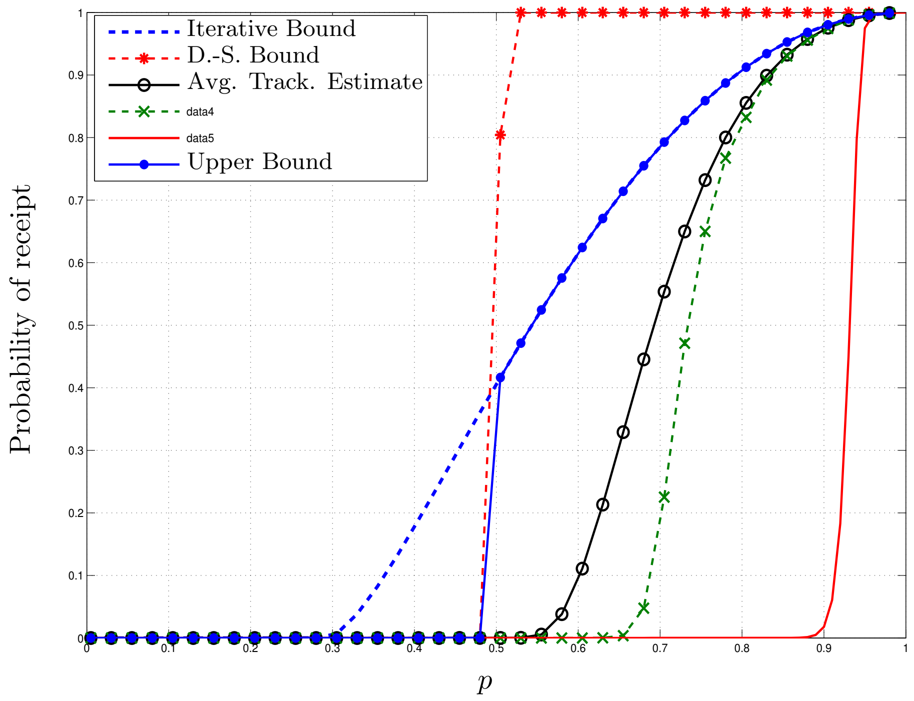

Figure 8.3: The probability of packet receipt at $(100, 100)$ vs the forwarding probability p for a directed grid.

can potentially attain a self-sustaining phenomenon due to the presence of a large number of events/paths.

Figure 8.6 presents the variation of the probability of receipt for the node at $(40, 40)$ in an undirected grid of 2601 nodes with the source at the center, i.e., $(26, 26)$. It can be seen that while the Dawson-Sankoff bound predicts a threshold-type behavior at a forwarding probability of about $p = 0.25$, the actual variation is captured well by the iterative tracking method. Also, Table 8.1 presents the probability of forwarding required to guarantee a reliability of 90% at all nodes in the undirected grid network of 2601 nodes arranged in 51

Table 8.1: The required probability of forwarding for 90% reliability at all nodes in an undirected grid of 2601 nodes.

Node	p (by simulation)	p (by avg. tracking)
$(40, 40)$	0.66	0.72
$(45, 45)$	0.67	0.74
$(50, 50)$	0.74	0.75

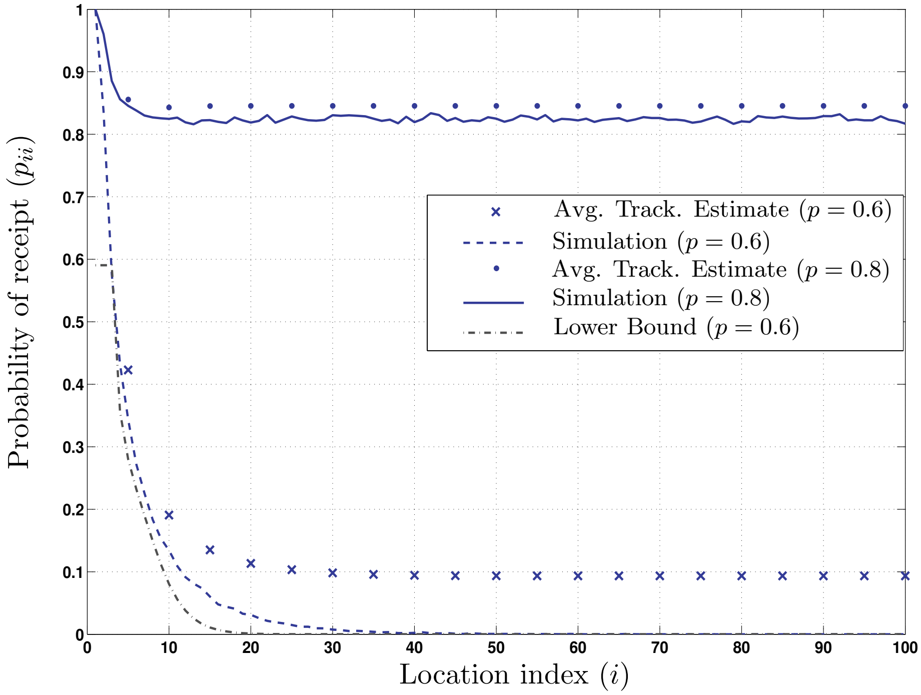

Figure 8.4: Variation of the probability of receipt at (i, i) vs i for $p = 0.6, 0.8$.

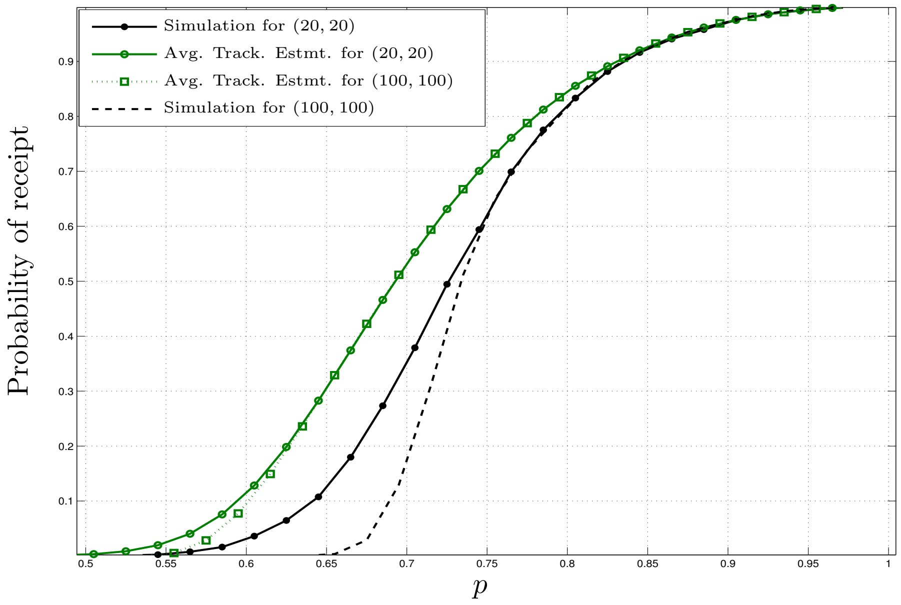

Figure 8.5: Variation of the probability of receipt at locations $(20, 20)$ and $(100, 100)$.

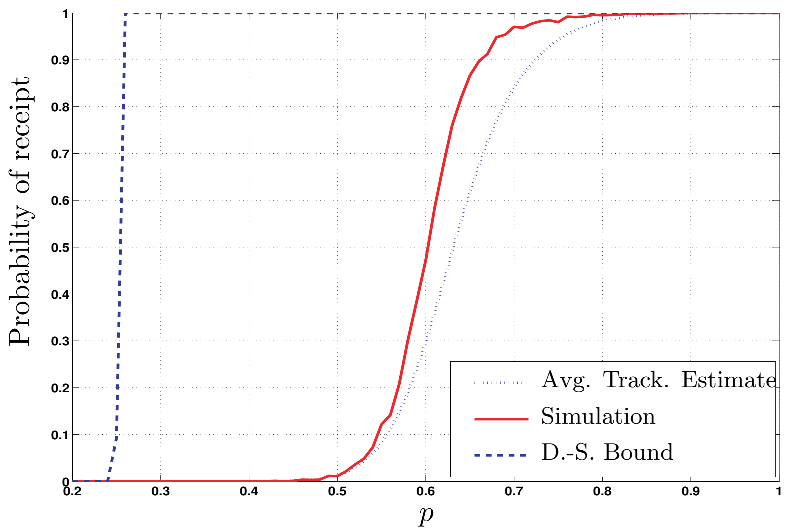

Figure 8.6: The probability of receipt vs forwarding probability for the node $(40, 40)$ in an undirected grid of 2601 nodes.

rows and columns. It can be noted that as the grid approaches the boundary, the estimate becomes more accurate. This is because the tail portions of the paths connecting the source to nodes in the vicinity of the boundary look fairly similar to that of the paths in directed networks. Hence, it can be expected that directed paths approximate well the probability of receipt for such nodes.

CHAPTER IX

RELIABLE MESSAGE DELIVERY IN DELAY-TOLERANT NETWORKS

Delay-tolerant networks (DTNs) are characterized by intermittent connectivity between nodes due to sparse node density and mobility, and, hence, long message delivery times. Unlike traditional ad-hoc networks, any snapshot of such networks is almost always disconnected. In these networks, communication and information exchange is possible between two nodes only when they are proximate. Therefore, these networks employ communication paradigms that rely on node mobility to carry data eventually to a destination in a time-bound fashion.

In this chapter, we investigate the problem of reliable and low-latency multiple single-source single-destination (unicast) message delivery in DTNs[1]. Intermittent connectivity, lack of end-to-end feedback, node mobility, and opportunistic communication in such networks make this problem very challenging. Conventional approaches to this problem involve the design of efficient routing schemes. Although considerable research is available for efficient routing of messages in DTNs, most of them employ simple replication of packets and multiple transmissions to ensure higher reliability and lower latency. Recently, hybrid schemes, wherein both replication of messages and simple erasure-coding are employed, were shown to have great potential as efficient solutions for DTNs. However, these works are either very simplistic or do not have realistic assumptions. For example, in [122], the authors consider schemes where data is first encoded with a replication factor of r and then packetized into sr chunks for some integer s. They are then relayed in a two-hop fashion. There are several drawbacks with this approach. First, although packetizing the replicated data ensures that the chunks have a smaller size than the original message, in order to ensure that all possible opportunities for dispersing data are used, these chunks must be fairly

[1] This work was done in collaboration with Ramanan Subramanian.

small, equivalently, s must be large. Second, employing a fixed-rate erasure coding scheme makes it natural to ask, “what rate is optimal?”. Clearly, the higher the rate, the better, but it comes with the cost of using more network resources such as bandwidth. Moreover, realistic assumptions such as finite packet expiry and time-varying channel losses make the scheme practically unusable since they would require the scheme to be rate-adaptable.

In this work, we show that a new class of packet-level codes called *rateless codes* can be employed for unicast communication in DTNs in order to obtain a significantly better reliability and delivery delay performance compared to existing schemes. Several aspects of rateless codes make them apt for such applications. First, their rateless nature obviates the selection of a good choice of rates even in the presence of varying channel loss conditions. Second, they are packet-level codes that have low complexity of encoding and decoding and require very low coding overhead to recover the message [109]. We show using extensive simulations under both real-world trace data such as UMassDieselNet testbed [17] and simulated trace data from the area-based random waypoint model [118] that our coding scheme offers higher reliability and lower latency than other coded and uncoded schemes. We see from simulations that our scheme suffers far less degradation in performance for the same decrease in packet expiry times or increase in message sizes when compared to others.

The rest of the chapter is organized as follows. Section 9.1 presents the details of our routing scheme. Network assumptions and models, simulation results and their implications follow in Section 9.2.

9.1 Proposed Approach for Packet Delivery in DTNs

In this work, we concern ourself with the benefits offered by rateless coding in DTNs. We do not assume any knowledge of the paths between nodes, or any statistics of the duration and inter-contact waiting times between any pair of nodes. However, during contacts between nodes, it is assumed that channel losses are absent. It is certain that coding would further enhance the performance of our proposed scheme relative to other alternatives when channel losses are present. The main aim of our approach is to increase the reliability and reduce the latency even in lossless channel conditions.

In this section, we describe in detail various aspects of our proposed scheme. First, we present its coding aspects. Next, we present the packet transfer protocol, *i.e.*, the set of rules that dictate packet transfer during contacts. Finally, we present the performance metrics that are used to evaluate our scheme.

9.1.1 Rateless-Coding-based Scheme (RCbS)

In our approach, when a message arrives at a node u of the network, the node packetizes the message into smaller data packets, each of size P. These smaller packets, usually a few thousands in number, are then encoded using the rateless scheme described above and with $\Omega(x)$ having the following form.

$$\Omega(x) = \frac{1}{\mu+1}\left(\mu x + \frac{x^2}{1\cdot 2} + \ldots + \frac{x^D}{(D-1)\cdot D} + \frac{x^{D+1}}{D}\right). \tag{9.1}$$

Here, μ and D are design parameters chosen as described in [74] such that for a set of $k = 1000$ data packets, a decoding (success) probability of $1 - 10^{-8}$ under the iterative message passing is guaranteed when the destination node receives any 1050 distinct encoded packets, or equivalently, when the coding overhead is $\Gamma_{1000} = 1.05$. Note that the probability that two encoded packets are identical is negligible.

Unlike direct contact schemes, in the RCb scheme, when a message arrives at a node u at time t and a contact with another node v is established at a later time, the source node u forwards encoded packets to v in an opportunistic manner. As in opportunistic flooding, each relay node in the network that has packets attempts to forward it to the first available node. In this work, we allow the intermediate nodes to only carry and forward packets to other nodes instead of allowing them to perform additional encoding operations although by allowing the latter, it is hoped that better results can be obtained.

9.1.2 Packet Transfer Protocol

Here, we describe our protocol for the transfer or exchange of packets when two nodes come in contact. It is assumed that the packet header stores the message ID as well as the addresses (or IDs) of the source and the intended destination node. Since any node can potentially carry packets from several messages simultaneously, it can act as the source, destination and relay for different messages at the same time. The packet transfer protocol employed is common to all the schemes considered in this work. Further, the buffers of the nodes are assumed to be very large. It is assumed that only the contact durations and expiry times limit the rate of message delivery in the network and not the sizes of the buffers. The following rules are to be executed when two nodes come in contact.

1. When the source node for a particular packet transfers the same to an acting relay node (i.e., the relay node is not the intended destination), it assigns a timestamp to that packet. Nodes use this timestamp to drop packets that have spent a duration more than T_{exp} in the network since their transmission by their respective source nodes. This duration is termed as the *packet expiry time* for the network and is used as a means to prevent congestion in the network.

2. A node does not transmit packets of a message back to its source during any contact with the latter.

3. An acting source or relay node, after transmitting packets corresponding to a particular message to the intended destination node, deletes the same from its buffer.

4. When two nodes N_1 and N_2 are in contact, the *priorities* of selection of packets to be transmitted from those stored in their buffers are determined in the following order with the first being the highest.

   (a) Packets carried by (but not originating from) a node for which the other node is the destination.

   (b) Packets originating from one of the nodes for which the other node is the destination, *i.e.*, the former is the source of the packet(s).

   (c) Packets originating from one of the nodes for which the other node is not the destination.

   (d) Packets that have neither originated from the node carrying it nor is destined for the other node (i.e., both N_1 and N_2 are acting relay nodes for the packets).

5. The packet with the highest priority is always transmitted first. If two or more packets have the same priority, the protocol acts such that they are equally likely to be chosen to be transmitted first. If the contact duration is long enough to allow additional packet transmissions, the protocol repeats the same choice process for the remaining packets.

The above priority rules can be justified as follows. Packet transmission to the destination is of high priority; relay to destination transmissions are carried out first in order to avoid dropping of the packets from the relay node's buffer due to potential expiry. Thus, source to destination transmissions are of lower priority, and come second. The next priority is to disperse packets from the source, hence source to relay transmissions will be prioritized next. Finally, relay-to-relay transmissions are of lowest priority since such transmissions may not always lead to quicker delivery of packets to the destination.

9.1.3 Performance Metrics

We use the following performance metrics to evaluate our coding scheme and to compare it with certain relevant unicast delivery schemes.

1. *Latency* $\mathcal{L}(\mathcal{M})$ of a message $\mathcal{M}$ is the time between the instant $\tau(M)$ that the message is generated at its source node and the time it is available at the destination node v. *Average Latency* of the system $\mathcal{L}_{\mathcal{S}}$ is the average latency that a random message takes to be delivered at its destination node, *i.e.*, $\mathcal{L}_{\mathcal{S}} = \langle \mathcal{L}(\mathcal{M}) \rangle_{\mathcal{M}}$.

2. *Reliability* $\eta(t)$ is the probability that a message random $\mathcal{M}$ has a latency smaller than t.

3. *Probability of success* P_{Success} is the probability that a random message $\mathcal{M}$ generated at a (random) time $\tau(\mathcal{M})$ is available at its destination before the simulation concludes at time $T_{\max}$ i.e., $P_{\text{Success}} = \langle P[\mathcal{L}(\mathcal{M}) \leq T_{\max} - t \,|\, \tau(\mathcal{M}) = t] \rangle_{\tau(M)}$. It can be noted that messages arriving later have a lower probability of reaching the destination since they have lesser time to spend in the network due to finite simulation time. A large $T_{\max}$ must be chosen to ignore this issue.

In the above definitions, we use $\langle Y \rangle_X$ to refer to the result obtained by computing the statistical (ensemble) average of the random variable Y with respect to the parameter X.

9.2 Results of Simulation and Discussion

We organize this section in the following format. First, we present the benchmark schemes with which we compare our proposed scheme (RCbS). Second, we present the message arrival model assumed. Next, we present three different network models that were used to performance evaluation. Finally, we present the results of simulations.

9.2.1 Relevant Schemes

1. Uncoded Unpacketized Scheme (UCUPS(r)): In this scheme, when a message arrives at a node, the latter does not break the message into smaller chunks. It waits for contacts whose duration enables the message to be transferred as a whole. This step is repeated multiple times until the source transmits r copies into the network. Relay nodes having messages forward to nodes in an opportunistic fashion. This is a multi-hop extension of the SRep scheme given in [50].

2. Uncoded Packetized Scheme (UCPS(r)): In this scheme just as in RCbS, when a message arrives at a node, it is broken down to smaller packets each with a size of P bits. Nodes where messages are generated then transmit these packets in a round-robin fashion. The source transmits each packet a maximum of r times. Just as in UCUPS, relay nodes having packets forward them to other nodes in an opportunistic fashion.

3. Erasure-coding based Scheme (ec(r, s)): This scheme proposed by Wang *et al.* [122] encodes the message using an erasure code with a replication factor r and then divides the encoded data into sr chunks for some integer s. The packets are then transferred to sr relays that in turn wait to deliver the packets to the destination directly. When exactly s relays deliver their packets to the destination, the destination decodes the data perfectly.

4. Dijkstra-based Greedy Scheme (DbGS): In this scheme, we assume that the contacts and the duration of all pairs of buses for the duration of simulation $[0, T_{\max}]$ are known. First, the list of all paths between a given pair of source and destination node is constructed. Then, the path with the shortest delay is selected and the maximum number of packets admissible is transmitted. Then, the next shortest-delay path is selected and transmission is performed. This process is repeated greedily until the destination has sufficient number of packets. This scheme is similar to that in [51, 50].

In all the schemes, the packet transfer protocol described in Section 9.1.2 is implemented. The differences in the schemes lie in the way data is processed at the source and at the relays and not the way it is relayed to other nodes. In order to make fair comparisons, all simulated schemes were set to the same replication factor as that of the erasure coding-based scheme in [122]. *I.e.*, when comparing results with those of ec(r, s), the source of a message (of size M) in every other scheme is allowed to transmit a maximum of $\frac{rM}{P}$ packets for that message. We denote by RCbS(r) the RCb scheme with the additional constraint that the source of a message transmits at most $\frac{rM}{P}$ packets of that message.

9.2.2 Message Arrival Model

The arrival model considered in our simulations is similar to the one used in [118]. We model message arrivals in the network with a Poisson process where the total traffic has a mean inter-arrival time $\frac{1}{\lambda}$ s (i.e., arrival rate of λ) occurring at any node in the network with equal probability. Effectively, the mean inter-arrival time for a message at a particular node is given by $\frac{n}{\lambda}$. Further, we assume that each arrival event results in a message of M bits. Finally, the destination for a message arriving at a node is equally likely to be any of the other $n - 1$ nodes.

9.2.3 Network Models

The network models used in the performance evaluation of the various aforementioned schemes are described below.

9.2.3.1 A Simple Campus Bus Model

In this model, we consider two simple variations of a campus bus model of $n = 10$ buses. In the first, called as the *fixed route campus bus* (FRCB) model, we assume that the nodes are buses moving on well-defined routes on a campus. Each bus corresponds to one particular route around the campus. Figure 9.1 illustrates the map of the campus and two of the bus routes chosen. Different stops and termini in the campus are marked with dark dots on the map. Operating buses are assumed to take a constant duration of $T = 1$ minute between any two adjacent stops on its route (termed as *inter-stop duration*). Once at a stop, the buses halt there for a time that is uniformly distributed between $[T^w_{\min}, T^w_{\max}] = [10, 20]$ s. Two buses communicate with each other if they are present at the same bus stop during an overlapping interval of time. The amount of data interchange that can occur is directly proportional to the duration of overlap at that stop. Our routes were so chosen that the probability of three buses being at a stop is negligible. In the rare event that three buses are present at a location, communication happens only between the first two buses that arrive at that stop. However, once one of the buses leaves the stop, the remaining two can communicate even if they were not doing so previously.

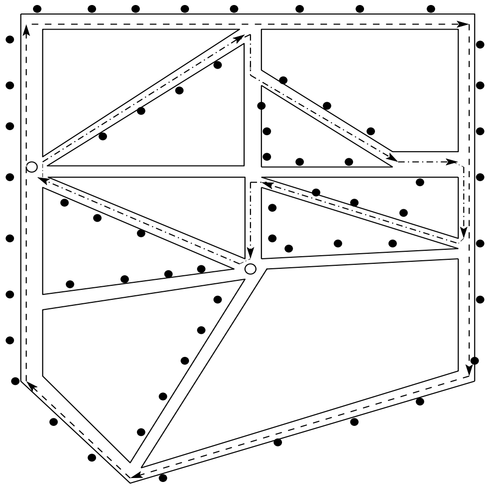

Figure 9.1: Map of the campus and some bus routes in the campus.

The second model is a random walk variation of on the campus map model. In this model, called as the *random route campus bus* (RRCB) model, buses do not have fixed routes. Whenever buses come to an intersection, they select at random, one of the roads from the set of roads intersecting at that junction and proceed on it. The inter-stop duration and waiting times at stops are assumed to be the same as in the FRCB model.

9.2.3.2 *Area-Based Random Waypoint Model (ABRWP)*

In this network model, $n = 10$ mobile nodes are assumed to be deployed in a 4 km × 4 km region. As shown in Figure 9.2, each node is restricted to move within a square cell of width 1 km inside the deployment region according to the Random Waypoint mobility model [13]. The parameters of the Random Waypoint motion are $v_{min} = 9$ m/s, $v_{max} = 11$ m/s and is the same for all the nodes. Further, the pause time at any waypoint follows the exponential distribution with a mean of 1 s. The radio range of any node is assumed to be 250 m. A similar network model was also studied by the authors in [118].

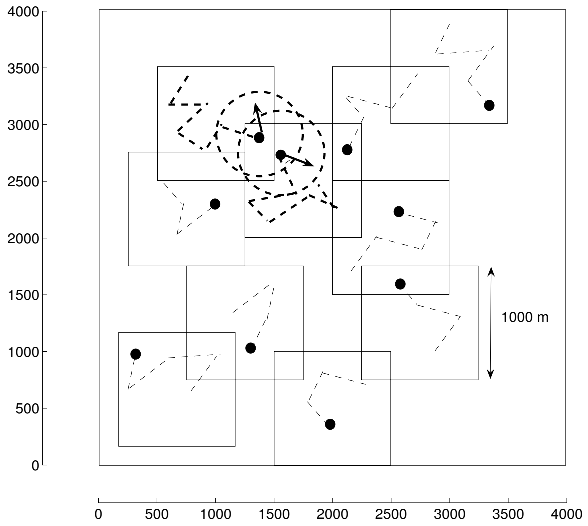

Figure 9.2: Area-Based Random Way Point Mobility Model.

9.2.3.3 *UMassDieselNet Testbed Data*

We also simulated the proposed RCb scheme with an actual DTN testbed data available in [17]. This dataset consists of communication traces for a network of 30 buses serving the University of Massachusetts at Amherst. The dataset of traces, taken over a period of 60 weekdays, contains information about contacts between the buses. Each contact information consists of the time of contact, number of bytes transmitted during the contact and the IDs of the buses in contact.

We now present the results of our simulations on the aforementioned network models. We classify our results into two categories based on the rate of message arrival. First, we describe the results when the message traffic is low and when there is little or no congestion in the network. Next, we present the scenario when there are multiple messages and when the demand for network resources such as bandwidth is high.

9.2.4 Low Arrival Rate Results

When the arrival rate λ is small, the network is not congested and the buffers of the nodes are near-empty. Hence, there is no severe competition for network resources such as bandwidth. In this section, by very low arrival rate, we mean rates such that $\frac{1}{\lambda} \gg \mathcal{L}_{\mathcal{S}}$. Under such arrival conditions, most of the time, only one message is present in the network. In this section, we present the results of simulation for our scheme and compare it with other schemes of relevance for the two campus bus models (RRCB and FRCB models) under such low traffic conditions.

Both models are simulated with the parameters given in Section 9.2.3. During contacts, it is assumed that the buses can transfer information at the rate of 1 Mb/s. Messages were set to a size of 1 Mb with packet size to which the message is broken down being set to $P = 1$ kb. As was previously mentioned, we do not restrict the size of the buffer sizes. The network with these parameters was simulated to evaluate the variations of reliability vs time, latency vs expiry time, and latency vs message size, respectively. Finally, the simulations were performed several times for a duration of $T_{\max} = 18$ hours.

Figures 9.3 and 9.4 present the variation of reliability $\eta(t)$ for varying t when expiry time was set to 100 minutes. It can be seen that the greedy scheme (DbGS) has the highest reliability in both the models since it uses information of the paths that are present in the network. In 99% of the simulation runs for RRCB model, our coding scheme succeeds in delivering sufficient encoded packets (and hence the message) to the destination within 2.673 hours. However, for the same time, other schemes such as ec(3,16), ec(2,16) and UCPS are able to recover the message only about 40.5%, 18.5% and 17.5% of the simulation runs,

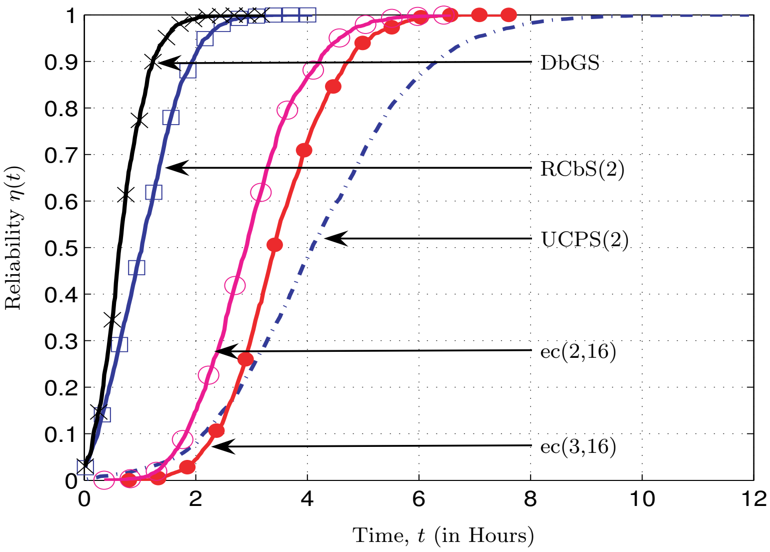

Figure 9.3: Dependence of reliability $\eta(t)$ on time t for the RRCB model.

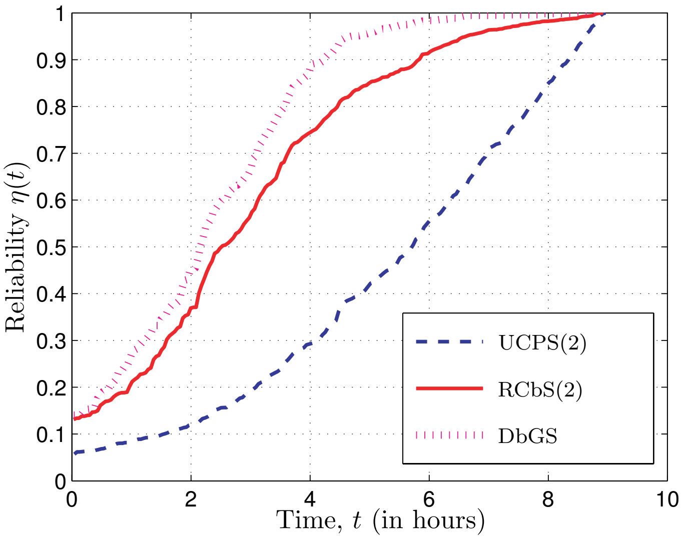

Figure 9.4: Dependence of reliability $\eta(t)$ on time t for the FRCB model.

respectively. It can also be noticed from the slope of the graphs that DbGS and RCbS(2) have smaller spread (variance) than the other three schemes making them more preferable.

In the FRCB model, although the same trend was noticed, only about 30.15 % improvement is noticed when the time taken to achieve a reliability of 90% in RCbS and UCPS are compared. From Figure 9.4, it can also be seen that distribution that while UCPS has a near-uniform density for times for successful recovery of the message, the same density for

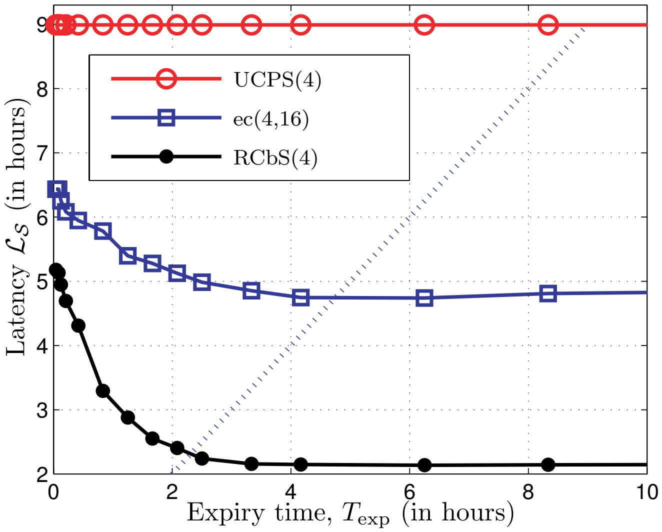

Figure 9.5: Latency $\mathcal{L}_{\mathcal{S}}$ vs expiry time for the FRCB model.

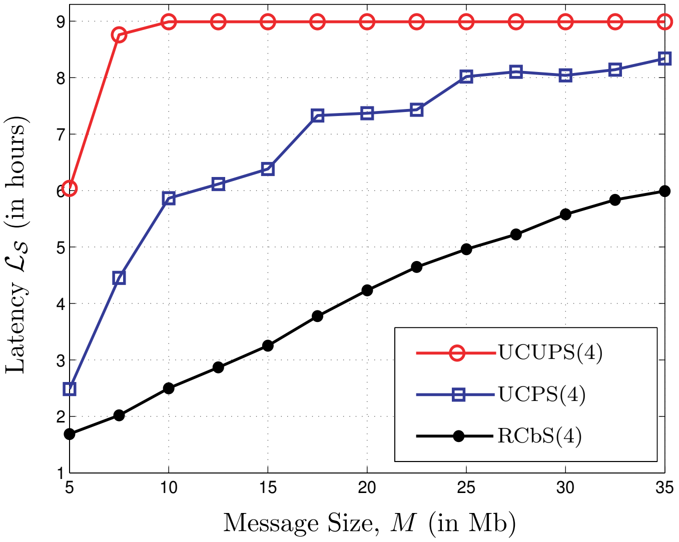

Figure 9.6: Latency $\mathcal{L}_{\mathcal{S}}$ vs Message Size M for the FRCB model.

our coding scheme has a slowly decaying tail.

In order to understand the behavior of various schemes under differing expiry times, we simulated the models for a duration of $T_{\max} = 9$ hours with a replication factor of $r = 4$ and expiry time T_{exp} varying from a mere 2.5 minutes up to 8.334 hours. We present our results only for FRCB model, since the inferences for the RRCB model show a strong similarity with that of the FRCB model. Figure 9.5 presents the variation between the two parameters of interest. Here, it must be mentioned that in any simulation run where a scheme failed to deliver the message, the latency for the failing scheme was set to $T_{\max}$. In the figure, it can be seen that the line $\mathcal{L}_{\mathcal{S}} = T_{\text{exp}}$, shown as a dotted line in the figure, intersects the RCbS(4) and UCPS(4) curves at $T_{\text{exp}}^{\text{RCbS}} = 2.31$ hours and $T_{\text{exp}}^{\text{UCPS}} = 4.74$ hours, respectively. Any choice of expiry times beyond these values for the respective schemes will provide little benefit in terms of latency, since in most of the runs, the required number of packets for the particular message will be delivered within the time of expiry of the first encoded packet transmitted from that message. For the same model, a comparison of the three schemes for the probability of successful message delivery is shown in Table 9.1. While all the packetized schemes report an improvement for the whole range of expiry times, RCbS(4) becomes extremely reliable for $T_{\text{exp}} > T_{\text{exp}}^{\text{RCbS}}$. To investigate the relationship between latency and message size, we simulated the performance of RCbS, UCPS and UCUPS with $r = 4$ for message sizes M ranging from 5 Mb to 35 Mb with the expiry time fixed at $T_{\max} = T_{\text{exp}} = 9$ hours. The results are presented in Figure 9.6.

Table 9.1: Dependence of P_{Success} on T_{exp} for FRCBM.

T_{exp}	UCUPS(4)	UCPS(4)	RCbS(4)
0.20833	0	0.5400	0.7495
0.41667	0	0.6055	0.8260
1.25000	0	0.6465	0.9680
2.08333	0	0.7160	0.9965
2.50000	0	0.7265	1.0000
3.33333	0	0.7555	1.0000
4.16667	0	0.7620	1.0000
8.33333	0	0.7695	1.0000

Just as in the previous simulations, we set the latency to be $T_{\max}$ for algorithms that do not deliver. As a result, we see that the latency of schemes tend to saturate. A linear growth with message size is clearly observed for RCbS. However, the other two uncoded schemes show saturation, implying that they fail to deliver the message within the simulation period.

Table 9.2 shows the superior performance of RCb scheme in terms of probability of successful message reception P_{Success}. RCbS is able to offer a success rate of over 87 % even for messages up to 35 Mb long. For the same message size, it can be seen that the UCPS performs far worse, while UCUPS fails even for much lower message sizes. The latter is expected since UCUPS relies on longer contacts to deliver long messages.

Table 9.2: Dependence of P_{Success} on M for FRCBM.

Message Size in Mb	UCUPS(4)	UCPS(4)	RCbS(4)
5.0	0.5872	0.93333	1.0000
7.5	0.0426	0.75035	1.0000
10.0	0	0.61277	1.0000
15.0	0	0.48794	1.0000
20.0	0	0.32908	0.9887
25.0	0	0.20851	0.9617
30.0	0	0.20000	0.9163
35.0	0	0.16596	0.8766

9.2.5 Multiple Message Arrivals

In this model, more than one message can potentially be present in the network at any time. Several unicast transmissions may take place simultaneously as opposed to the simple very low arrival case in Section 9.2.4. In the following, we discuss the simulated performance of the proposed RCb scheme for DTNs for the various network models.

Table 9.3: P_{Success} vs $\frac{1}{\lambda}$ for ABRWP.

λ^{-1}(in hours)	UCPS(4)	ec(4,16)	RCbS(4)
1.1111	0.49596	0.6707	0.95819
1.3889	0.49596	0.6764	0.96633
1.6667	0.50824	0.6735	0.96211
2.2222	0.5142	0.6740	0.96288
2.7778	0.51455	0.6788	0.96975
5.5556	0.52756	0.6841	0.97731

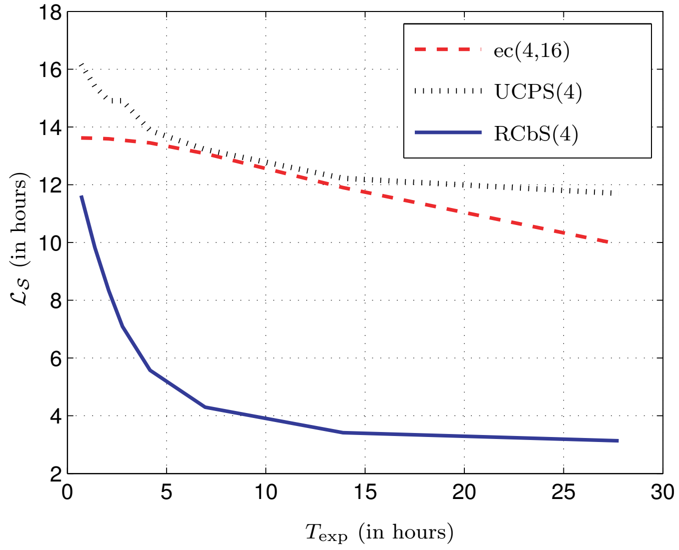

Figure 9.7: Latency $\mathcal{L}_{\mathcal{S}}$ vs expiry time for the ABRWP Model.

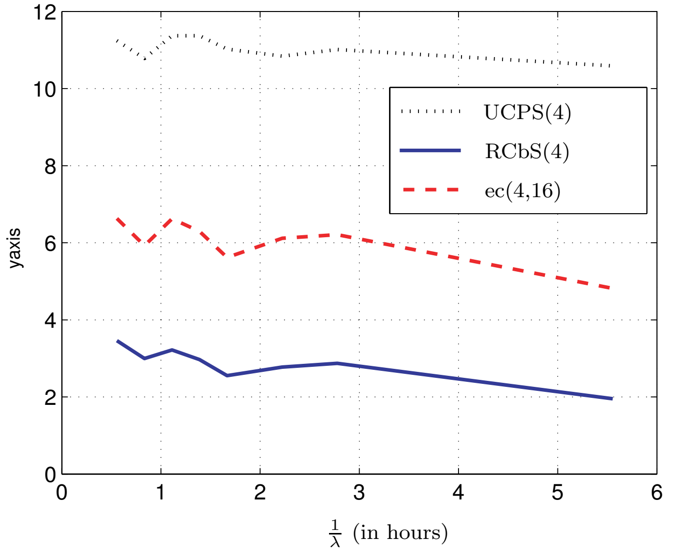

Figure 9.8: Latency $\mathcal{L}_{\mathcal{S}}$ vs inter-arrival time for the ABRWP Model.

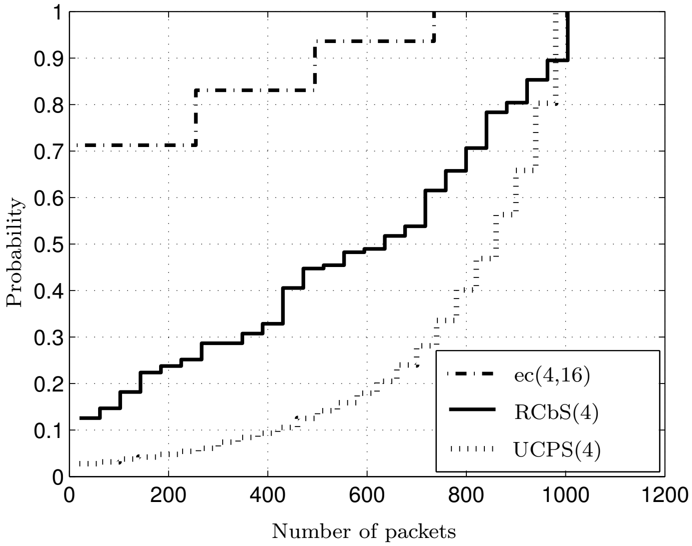

Figure 9.9: Cumulative distribution of the number of packets of partially received messages for the ABRWP Model.

9.2.5.1 Results for the ABRWP model

For this network model, the message size M and the replication factor r is set to be 50 Mb and 4, respectively. The packet size and bandwidth were chosen to be 1 kb and 1 Mb/s respectively. The simulation duration $T_{\max}$ was set to be 27.78 hours. The results of our simulation show consistent improvements in both the average network latency $\mathcal{L}_{\mathcal{S}}$ and the success probability P_{Success} over existing schemes. Table 9.4 shows the variation of P_{Success} with the expiry time T_{exp} for a mean inter-arrival time of 2.7778 hours. For very large expiry times, one observes that the ec(4,16) scheme gives about 51% success rate and UCPS gives

Table 9.4: P_{Success} vs T_{exp} for ABRWP

T_{exp}(in hours)	UCPS(4)	ec(4,16)	RCbS(4)
0.69444	0.16259	0.30297	0.40411
1.38889	0.20414	0.30356	0.50321
2.08333	0.23189	0.30463	0.58615
2.77778	0.23189	0.30728	0.65671
4.16667	0.29138	0.31273	0.74493
6.94445	0.33152	0.33385	0.82251
13.8889	0.3995	0.40206	0.88214
27.7778	0.44778	0.51471	0.90420

about 45%. However, the proposed RCb scheme offers more than 90% success. Further, the maximum expiry time for a moderate success rate of about 80% can be as high as 6 hours. For such an expiry time, the other schemes offer a success rate of only about 30%. Further, the proposed scheme also achieves the success rates with far less average latency, as illustrated by Figure 9.7, wherein a threefold improvement over the ec(4,16) scheme is observed. Similarly, for unlimited expiry times (*i.e.*, $T_{\text{exp}} = T_{\max}$), the variations of the success probability and the average network latency with the message inter-arrival time $\frac{1}{\lambda}$ ranging from 33 to 333 minutes are given in Table 9.3 and Figure 9.8, respectively.

An interesting trend can be observed in the cumulative distribution function of the destination buffer occupancy of incomplete messages (*i.e.*, those messages for which the destination does not have sufficient packets to reconstruct the messages) in Figure 9.9. It can be seen that the distribution for the UCPS is concentrated at higher values implying that the scheme suffers from failed message transmission primarily due to the non-availability of the last few packets to complete the message. However, for the RCb scheme, the distribution is more uniform. For the ec(4,16) scheme, the cumulative distribution function is piecewise constant since the scheme always transmits blocks of 250 packets for the chosen values of parameters.

9.2.5.2 Results for the UMassDieselNet Testbed Data

For this testbed dataset, the message size was set to be $M = 512$ kb and a repetition factor of $r = 4$ was chosen for all schemes. The packet size and bandwidth were chosen to be 512 bits and 1 Mb/s respectively. The simulation duration $T_{\max}$ was set at 24 hours. Figures 9.10 and 9.11 shows P_{Success} and $\mathcal{L}_{\mathcal{S}}$ as a function of T_{exp} for the UMassDieselNet dataset available at the online CRAWDAD database [17]. Here, the inter-arrival time $\frac{1}{\lambda}$ is fixed at 1.667 hours, while the expiry time T_{exp} is varied from 0.6944 hours to 27.7778 hours. Again, the RCb scheme outperforms UCPS as well as the ec(4,16) scheme, even though the benefits over the latter are not as pronounced as in the ABRWP mobility case. Figure 9.12 shows the variation of the number of messages that the source has exhausted transmitting and the number of messages received by the destinations, for the two schemes, for identical

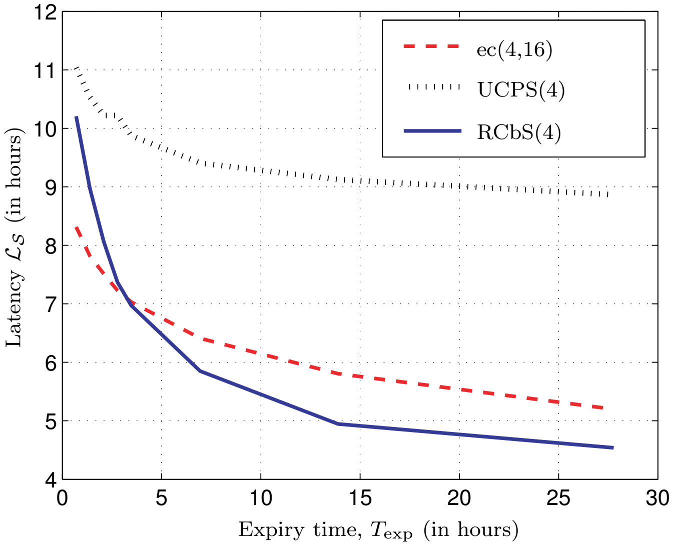

Figure 9.10: Latency vs expiry time for the UMassDieselNet dataset.

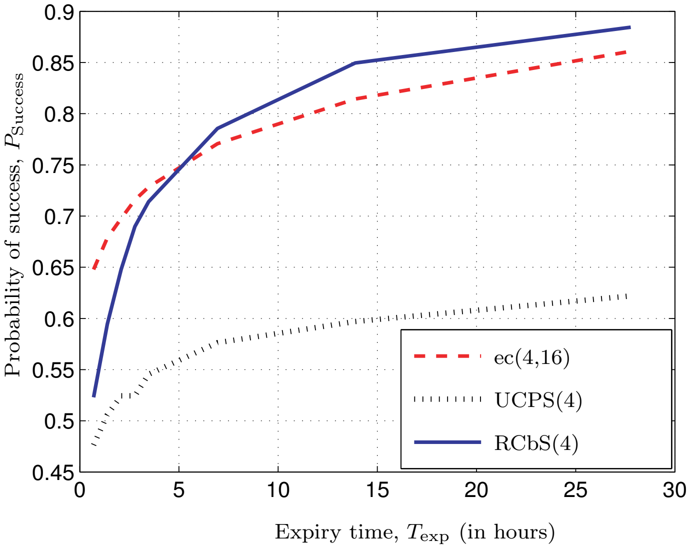

Figure 9.11: Success probability vs expiry time for the UMassDieselNet dataset.

message arrivals in the network and a fixed expiry time of 27.7778 hours. Here, a source is said to have *exhausted* transmitting a message if it has forwarded kr packets belonging to that message to the network, where k and r refer to the number of packets in a message and the repetition factor, respectively. The plots clearly show an improvement by 20% to 40% over the ec(4,16) scheme. One can clearly see that the latter scheme has lesser number

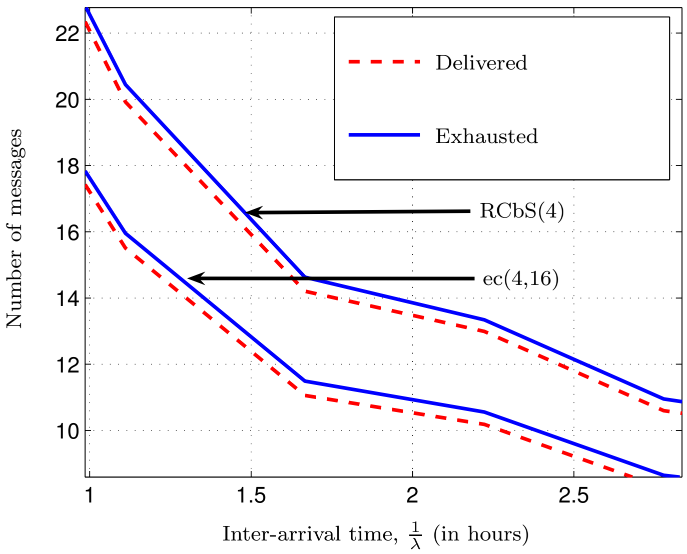

Figure 9.12: Transmission efficiency vs inter-arrival time for the UMassDieselNet dataset.

of exhausted messages, since it ignores contact durations shorter than the time required to transmit one block of packets. However, in our scheme, all contacts will be used to transmit packets, including ones with short durations that are ignored by ec(4,16).

PART III

Markov Chain Approach to Capacity of Finite-buffer Networks

CHAPTER X

UNICAST THROUGHPUT OF NETWORKS WITH MEMORYLESS MOBILITY ON GRIDS

This work analyzes the unicast throughput in a ad-hoc wireless network of nodes performing random walks on a grid[1]. Due to the random mobility aspect, these networks are characterized by frequent lack of end-to-end communication paths and employ the *store, carry, and forward* paradigm for communication. A model of the network based on simplistic Poisson-process approximations are inaccurate in predicting the throughput when the nodes move using a memoryless random-walk based mobility model. Here, a novel analysis approach using Markov chains is developed for such mobility models, and throughput is derived for the single unicast scenario. The model is comprehensive enough to allow both the incorporation of contention and the limitation due to finite buffers.

The chapter is organized as follows. Section 10.1 explains the shortcomings of some of the present approaches and motivates our work. Section 10.2 presents the models and definitions that are assumed throughout this work. Section 10.3 presents our general Markov chain-based approach to the problem of throughput computation. Sections 10.4 and 10.5 detail our contributions to the computation of throughput in the two-hop mode of communication. Finally, Section 10.6 presents a discussion on the results of our simulations.

10.1 Motivation

In this section, we illustrate the ineffectiveness of modeling of networks with memoryless mobility based on the Poisson-process contacts. This approach is commonly employed in similar contexts [33, 1, 45, 128]. A critical issue in performance analysis of mobile networks with finite buffers is that the models and assumptions employed need to be able to track buffer variations and the queuing effects in a fairly accurate manner. The authors in [33]

[1] This work was done in collaboration with Ramanan Subramanian.

argue that an accurate analysis of *sparse mobile ad-hoc networks* (SPMANETs) can be obtained by such a modeling. The authors of [33] validate this claim for the case wherein multiple copies of the same message are disseminated across the network to decrease message delay. However, in their model, no restrictions on the sizes of node buffers are posed. However, as we shall see, in the case of limited buffers, the Poisson-process model results in a gross simplification of buffer statistics and an overestimation of the throughput of the network.

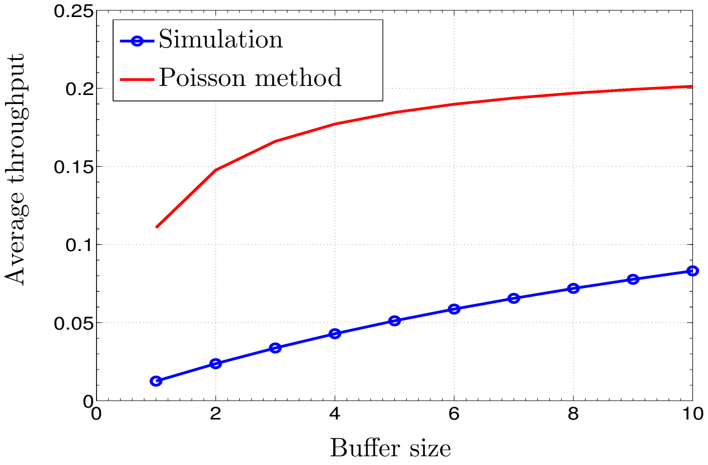

Figure 10.1: Average throughput vs buffer size.

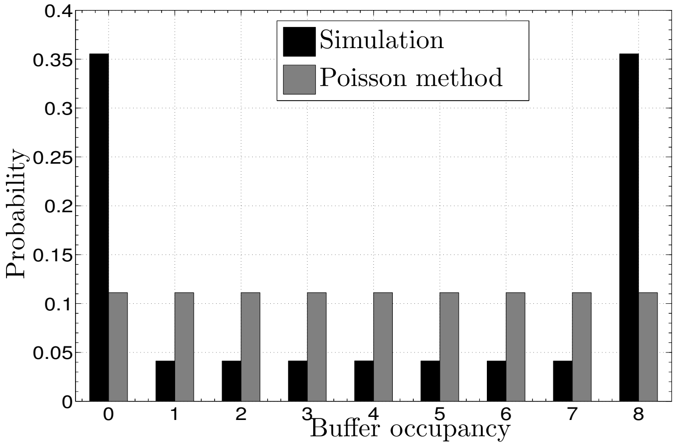

Figure 10.2: Steady-state buffer occupancy probabilities for a buffer size of 8.

An illustration of the inadequacy of Poisson-process mobility models is shown in Figures 10.1 and 10.2. The figures show the throughput and buffer occupancy distribution for a network of 102 mobile nodes on a 20 × 20 square grid. All the nodes were assumed to perform a discrete-time random walk on the square grid. The throughput was calculated for a two-hop single-copy routing for this network for a selected pair of nodes as the source and the destination and the remaining hundred nodes acting as relays. At every contact with a relay or the destination, the source node transmits a single block of packets. Note that no two packets transmitted by the source are identical. Furthermore, the size of the buffer of each relay node was set to 8 blocks. The variation of the throughput achieved (over a long duration of time and under the two-hop scheme) with the buffer size is depicted in Figure 10.1 for both the exact random walk simulation as well as for the equivalent Poisson model of [33]. We see that the Poisson approximation deviates considerably from the exact simulation, by as much as 100%. The reason for this is clearly observed from Figure 10.2 depicting the plot of the steady-state buffer occupancy probabilities. One can see that the Poisson approximation method fails to capture a critical feature that the simulation indicates – the network spends considerably longer periods in the empty-buffer state when it is in contact with the destination. As a result, the Poisson-based approximation overestimates the throughput in finite-buffer conditions.

Therefore, we notice that in such a network the Poisson model is inaccurate, and as a consequence, the inter-contact times of packet arrival and departure from nodes do not follow a geometric distribution. Accurate models for the distribution of contact times in such mobility scenarios are not available. In this work, we model the finite-buffer network using Markov chains that reflect the memoryless mobility in the transition between the states of the chain. We show that by using Markov chains, accurate prediction of the throughput of such networks is feasible.

10.2 Models and Definitions

In this section, we first describe the network and the mobility model assumed. We then describe the packet transfer protocol for various nodes in the network. Finally, a discussion contention resolution is presented.

10.2.1 Network and Node Mobility Model

We model the network as a deployment of a set of nodes that perform the *natural random walk* on a uniform two-dimensional grid in discrete time. We note that this is a space-time discretized version of the Brownian motion model defined in a finite region. The mobility characteristics of nodes in the network are assumed to be independent and identical to each other. By the term *natural random walk*, we mean that a mobile node at any time can choose to remain in the same location in the grid, or choose to move to an adjacent location in the next time step. Figure 10.4 presents the transition probability for the natural random walk at the corner, edge and interior vertices of the grid.

Throughout the paper, we deal with the *reversible uniform grid-mobility model* as a special case of mobility. Here, the deployment region consists of an $\sqrt{N} \times \sqrt{N}$ square grid on a 2-D plane. The random walk is in discrete time, with a constant step size of T. There is an assignment of transition probabilities such that the resulting Markov chain is both *reversible* and *uniform*, as shown in Figure 10.4. Though not indicated, the transition probabilities for i to j is the same as from j to i for any two points i and j in the grid. We use this assignment of transition probabilities (weights) for convenience in analysis. Clearly, the resulting Markov chain for the natural random walk model is irreducible and ergodic. Equivalently, we can also consider the deployment region to be a unit-square region divided into equal N square cells, as shown in Figure 10.3, wherein nodes walk between adjacent cells, rather than vertices. We shall adopt this interpretation for the rest of this section.

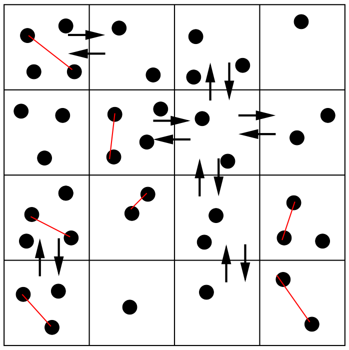

Figure 10.3: The network model.

Two nodes communicate with each other only if they are in the same cell. Further,

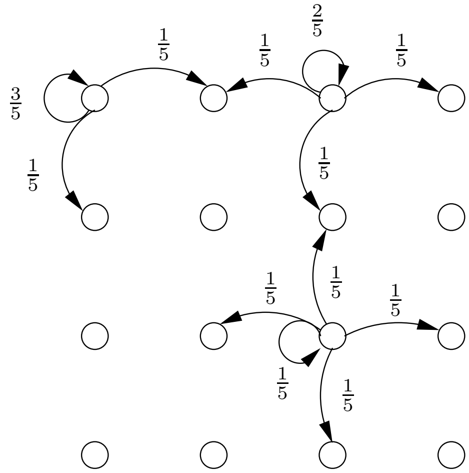

Figure 10.4: Assumptions for the parameters of node mobility.

in order to ensure collision-free communication, we assume that when two nodes in a cell communicate, the MAC protocol that is in place, makes sure that all other nodes in the same location are silent. A communication link lasts for a duration equal to the step size T in the discrete random walk. Each communication link leads to successful transmission of a block of k packets between nodes, *i.e.*, in our model, the wireless channel is assumed to be lossless and time-invariant. We further assume that the nodes possess buffers of B blocks. Without loss of generality, we assume $k = 1$ throughout this paper.

10.2.2 Packet Transfer Protocols

Next, we describe the packet routing protocols used in our analytic modeling. We assume that messages at the source are packetized, and that the source tries to transmit an unlimited number of them continuously to the destination. Throughout this work, we assume that the source node never transmits multiple copies of the same packet. Hence, at any given time, there is at most one copy of the packet in the network. A contact is said to be "successful" if a pair of nodes that are in contact, win the contention phase and get to communicate with each other. The following models are used for the two-hop protocol and contention resolution.

- **Two-hop scheme:** A mobile (relay) node, after accepting packets from the source, retains the same until a successful contact with a destination node occurs. No transmission happens when contacts occur between two mobile (relay) nodes.

- **Contention Resolution:**
  - At any given location in the network, at most one pair of nodes is allowed to communicate. All other nodes in the same location remain silent.
  - Whenever the source and destination are in the same location, the source successfully transfers a block of k packets to the destination node.
  - If several relay nodes meet a source node and the destination node is not in the same location, one of the relays is selected at random and the source node attempts to transfer k packets to the relay.
  - If several relay nodes meet a destination node and the source node is not in the same location, one of the relays is chosen at random and it transfers k packets from its buffer to the destination, if possible.

10.3 Description of the General Approach

In this work, we are interested primarily in the *throughput* of the network that is defined as follows. Once the network is initialized and the communication paradigm is in effect, packets are transmitted across the network. After a sufficiently long time (after which we assume that the network is in its steady state), during which the communication happens continuously, we observe the network for a duration of τ. For a given source-destination pair, say s and d, the expected rate at which the network transfers packets from s to d is defined as the throughput capacity. In other words if $\mathcal{N}_{s,d}(\tau)$ packets are transmitted from s to d in the duration τ, the throughput capacity is:

$$\mathcal{C}_{s,d} = \frac{\mathcal{N}_{s,d}(\tau)}{\tau} \tag{10.1}$$

Also, The *average latency* is defined as the expected amount of time a packet has to spend in the relays nodes' buffers before reaching the destination.

To model the throughput in a general random walk over an undirected graph setting (*i.e.,* not necessarily the natural random walk over the grid), we construct the Markov chain and the *state* of the network as follows. Assume that the network has n nodes performing a random walk defined on a weighted graph G independently. Let $X(t) = (X_1(t), X_2(t), X_3(t), \ldots, X_n(t))$ be the locations of the nodes at times $t > 0$ and let $\mathcal{M}(t) = (\mathcal{M}_1(t), \mathcal{M}_2(t), \mathcal{M}_3(t), \ldots, \mathcal{M}_n(t))$ be the buffer occupancies of the nodes at the end of all communication in that time slot. Then the pair $(X(t), \mathcal{M}(t))$ uniquely identifies the state of the network at time t. Clearly, given $(X(t), \mathcal{M}(t))$, we know that the probability of transition to $(X(t+1), \mathcal{M}(t+1))$ is defined from the mobility model and the protocol model. This results in a Markov chain which uses full knowledge of the network state. The above network-level Markov chain will be ergodic if the random walk itself is ergodic, which is the case in our work. Let $\mathbf{\Phi}$ be the state transition matrix for the chain tracking $(X(t), \mathcal{M}(t))$ and let $\boldsymbol{\psi}$ be its steady-state distribution. A point to note here is that the position-marginal distributions $\boldsymbol{\psi}_{X_j}$ pertaining to the positions of nodes j will be independent and identical to the random walk steady-state statistics. However, in general, the buffer-marginal distributions $\boldsymbol{\psi}_{\mathcal{M}_j}$, though identical, will neither be independent of each other, nor be independent of the node locations.

The computation of throughput capacity is complete if determine the frequency of visits to certain states called as the *desirable states* of the chain. To derive the throughput capacity of this network, the following result ensures the sufficiency of the knowledge of the steady-state probability distribution $\boldsymbol{\psi}$ of the states of the chain.

Theorem 10.3.1. *Let q_j be a desirable state for a given source destination pair (s, d), and let $\nu(q_j)$ be the number of packets delivered to d that originated from s in each of the states. Then, the throughput of the network is given by:*

$$\mathcal{C}_{s,d} = \sum_j \nu(q_j)\psi_{q_j}. \tag{10.2}$$

Proof. This follows directly from the irreducibility of the constructed chain. ❑

Since the state space for the network is generally huge, it is very difficult in general to compute the steady-state distribution through direct analysis. We tackle this situation by using reduction and estimation techniques over this chain for various specific communication scenarios that are detailed below.

10.4 Two-hop Single-unicast with Immobile Source and Destination

Our first case of analysis assumes that the source and destination nodes are immobile and are located at two diagonal opposite corners (*i.e.*, $(0,0)$ and $(\sqrt{N}-1, \sqrt{N}-1)$, resp.,) in a $\sqrt{N} \times \sqrt{N}$ grid network. We label this case as ISD_2. One can also repeat the same analysis for other source/destination locations with virtually no difference in the methodology. Let us suppose that n mobile relay nodes, each with a buffer limit of B blocks, are deployed in the network. We are interested in deriving the throughput capacity of such a network. For our initial discussion, we ignore the effect of contention and shall introduce it later. Effectively, this is an idealized situation wherein, within a time slot, the source/destination is capable of transmitting/receiving one block to/from each of the node it is in contact with, respectively.

Relay nodes receive packets on meeting the source and deliver them on meeting the destination. By the two-hop restriction, a relay is not allowed to transmit its packet to another relay. As stated in the previous section, we construct the network-level Markov chain to enable us in the analysis of throughput.

Since the network is homogeneous, the contributions of each node to the throughput is identical. An important consequence of this result in the two-hop protocol is that it enables us to derive the throughput on a per-node basis independently of the other nodes. This results in a three-dimensional Markov chain that has $N(B+1)$ states as depicted in Figure 10.5. Each layer represents a different occupancy state b for the buffer. Further, the node locations (except the source and destination locations) in the grid are to be replicated at each layer as shown in the figure. The transition probabilities are not indicated, it simply suffices to say that they are exactly the same as the corresponding transition probabilities for the random walk over the grid.

A little clarification on the nomenclature used for states follows here. In Figure 10.5, states indicated by a solid bubble have self-loops, and those indicated by hollow bubbles do not. The hollow bubble states are designated as *active states*, wherein the node picks up/delivers packets from the source/destination. These states are designated as

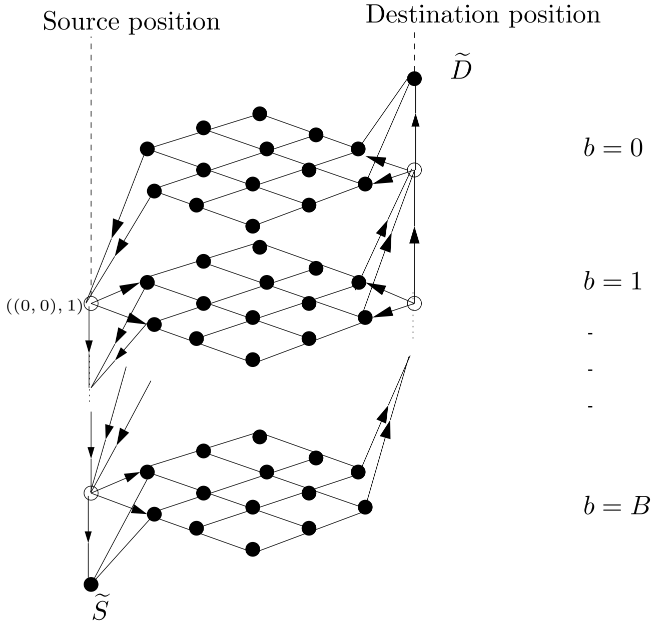

Figure 10.5: Markov chain for the two-hop grid network (ISD_2).

$S_1, S_2, \ldots, S_B$ and $D_0, D_1, \ldots, D_{B-1}$. There is no state designated as S_0 or D_B since a relay node cannot have zero/full buffer after successful contact with the source/destination node. States $\widetilde{S}$ and $\widetilde{D}$ are *non-contributing states*, corresponding to saturated and empty buffer state while the node meets the source and destination, respectively. All other states in the ISD_2 chain are *passive states*, and are designated by the two-tuple (x, b) corresponding to the location of the node in the grid and the buffer occupancy. Detailed analysis and results for ISD_2 are presented below.

Theorem 10.4.1. *The throughput of ISD_2 network with node buffer sizes B, grid size N, and n mobile nodes is given by*

$$\mathcal{C}_{s,d} = \frac{n}{N}\left(\frac{B}{B + \left[\frac{5\gamma(N)+8}{2}\right]}\right) \approx \frac{n}{N}\left(\frac{B}{B + 3\log N}\right), \tag{10.3}$$

where $\gamma(N) = 4\sum_{k=0}^{N-2}\varphi(k,0)$ and $\varphi(k,l)$ is the solution to the two-dimensional difference equation system

$$4\varphi(k,l) = \varphi(k-1,l) + \varphi(k+1,l) + \varphi(k,l-1) + \varphi(k,l+1), \; 0 \le k,l \le N-3 \tag{10.4}$$

with the boundary conditions

$$\varphi(N\!-3,k) = 0, \qquad 0 \le k \le N-1 \tag{10.5}$$

$$\varphi(N-4, N-3) = \frac{1}{2} \tag{10.6}$$

$$\varphi(0,0) = = \frac{1}{2}. \tag{10.7}$$

Proof. Refer Section C.1 of Appendix C for a proof of the result. ❑

Next, we incorporate the effect of contention into our model. In the resulting network, every time a given node meets the source or the destination node, it encounters some contenders for the finite bandwidth resource. Using the ASTA theorem [77], one can show that the ISD_2 Markov chain can again be reduced to that from a single node's perspective. We will then have additional states in the new chain that are designated by $\widetilde{S}_1, \widetilde{S}_2, \ldots, \widetilde{S}_B$, and $\widetilde{D}_0, \widetilde{D}_1, \ldots, \widetilde{D}_{B-1}$. A sample state addition for a part of the original chain is shown in Figure 10.6. These additional states correspond to the event that no change of the buffer state occurs when meeting the source or the destination. Such an event can occur a relay node loses contention in the presence of competing nodes. The Markov chain for this network largely remains the same, with the exception of these additional states and the corresponding modifications to the transition probabilities, as shown in the figure. Here, p_c is the probability that a node wins contention on meeting the source or destination and can be easily computed from the steady-state random walk parameters. The analysis of the resulting chain is similar to the previous contention-free case and is omitted to avoid redundancy. We then have the following result.

Theorem 10.4.2. *For ISD_2 network restricted by contention, the total throughput is given by*

$$\mathcal{C}_{s,d} = \frac{np_c}{N}\left(\frac{B}{B + (1-p_c) + p_c\left[\frac{5\gamma(N)+8}{2}\right]}\right), \tag{10.8}$$

where $p_c = \frac{N}{n}\left[1 - (1-\frac{1}{N})^n\right]$ and $\gamma(N)$ is as defined in Theorem 10.4.1.

Using the above result in conjunction with the *Little's law* [20], we can conclude that the average packet delay in our network is given by

$$\mathcal{L} = \frac{N}{2p_c}\left[B + (1-p_c) + p_c\left\{\frac{5\gamma(N)+8}{2}\right\}\right].$$

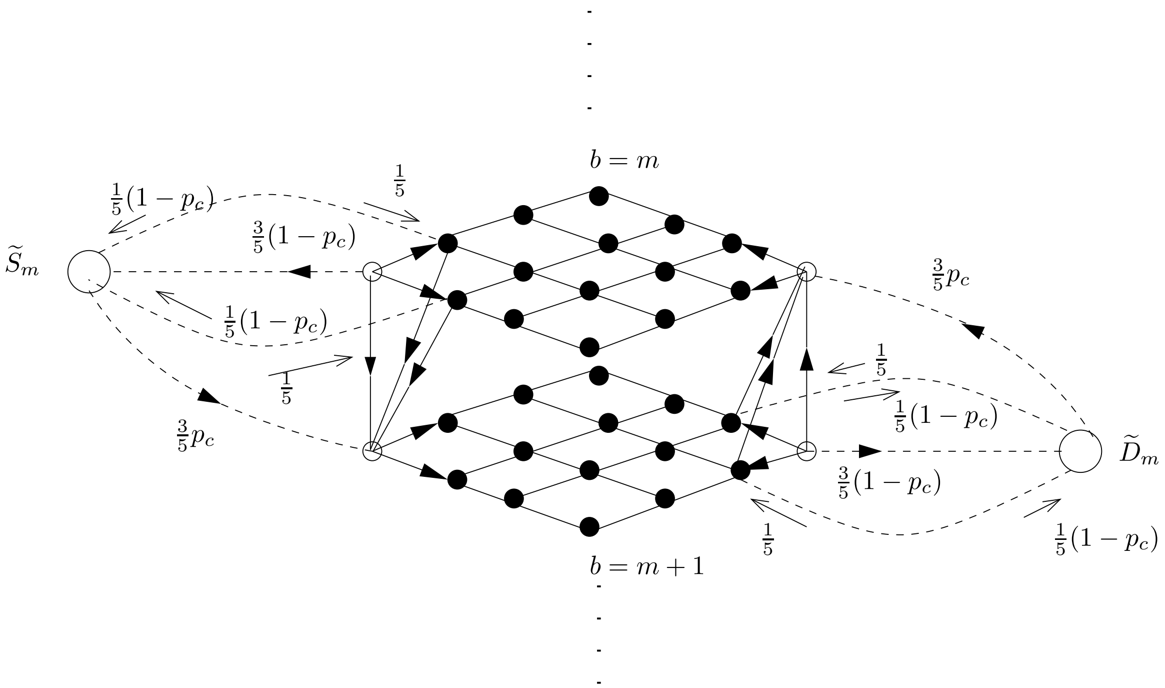

Figure 10.6: State-space modification to include contention.

An observation from the above analytic result is that the finite-buffer and sparseness together have a compounded throughput reduction effect. Further, as in the case of general ad-hoc networks, the per-node throughput decreases with the increase in n even though the total throughput of the network increases. However, interestingly, this decrease is accompanied by a slight mitigation of the limiting effect due to finite buffers.

10.5 Two-hop Single-unicast with Mobile Source and Destination

Since exact modeling of mobile-source mobile-destination unicast under the two-hop scheme involves construction of a Markov chain tracking both the location of the node, source, destination and the occupancy of the buffer, the number of states in one such Markov chain will be exactly N^3B states. Since such a model is computationally intensive, as a simplification, consider an approximate Markov model MSD_2 for this network setting from the perspective of an relay node wherein each relay node keeps track of its buffer $\mathcal{M}(t)$ and an additional partial location information that is a quaternary-valued random variable $\mathcal{R}(t)$

defined as follows.

- $\mathcal{R}(t) = 1$ if at time t, the node shares the same location as the source.
- $\mathcal{R}(t) = 2$ if at time t, the node shares the same location as the destination.
- $\mathcal{R}(t) = 3$ if at time t, the node shares the same location as both the source and the destination.
- $\mathcal{R}(t) = 4$ if at time t, the node shares the same location with neither the source nor the destination.

The fundamental idea behind such a model is the fact that under the chosen random walk pattern, the buffer occupancy does not depend on the exact location but only the instance of the random variable $\mathcal{R}$ at that time. In MSD_2, we have to keep track of only $4(B+1)$ states, a drastic reduction in the complexity of the original problem. However, we shall note in Section 10.6 that this reduction in complexity does not lead to an oversimplification of the tradeoffs pertaining to the problem. Since we track neither the exact positions of the relay node nor that of the source or destination, we model the transition of the partial location information using an "average" model based on the steady-state probabilities of the nodes. This average model for tracking the variation of $\mathcal{R}(t)$ shall assume the nodes are distributed in the network locations with their respective steady-state probability. For instance, in order to compute $\Pr[(\mathcal{R}(t+1) = 1)|(\mathcal{R}(t) = 1)]$, we assume that the locations of both relay node and that of the source is given by their steady-state distribution. Hence, the probability of this event can be calculated by averaging the probability that both the relay and the source nodes move from the same node to another or remain together at the location they were at time t. Equivalently,

$$\Pr[(\mathcal{R}(t{+}1){=}1) \wedge (\mathcal{R}(t){=}1)] = \sum_{l,l'=1}^{N} \frac{1}{N} P_{ll'}^2 \tag{10.9}$$

$$= \frac{5N + 8\sqrt{N} + 8}{25N}. \tag{10.10}$$

The computation for conditional probabilities other values can be performed similarly. Further, note that the computation of these transition probabilities in independent of the buffer state and transitions. Using these computed average marginal transition probabilities, the packet transmission protocol for two-hop and the contention resolution model as described in Section 10.2, we can model the various transition probabilities $\mathbf{\Phi}((r',b')|(r,b))$ for various values r, r', b, b' in the Markov chain MSD_2. The dynamics of this model that are straightforward are enlisted below.

$$\begin{aligned} 1.\ \mathbf{\Phi}((1,b')|(r,b)) &= 0 \text{ if } b' < b. & (10.11)\\ 2.\ \mathbf{\Phi}((2,b')|(r,b)) &= 0 \text{ if } b' > b. & (10.12)\\ 3.\ \mathbf{\Phi}((r',b')|(r,b)) &= 0 \text{ if } r', r \in \{3,4\} \text{ and } b \neq b'. & (10.13) \end{aligned}$$

Once the exact Markov chain is constructed using the parameters, one can directly compute the steady-state probabilities for the states of the chain. Using the steady-state probabilities, the throughput of the two-hop scheme for a network of mobile source and destination nodes computed using the MSD_2 model is as follows.

Theorem 10.5.1. *Consider the MSD_2 for a network of n relay nodes each with a buffer size of B in a grid with $\sqrt{N} \times \sqrt{N}$ vertices. Then, the throughput between the source node s and the destination node d is given by*

$$\mathcal{C}_{s,d} = \frac{np_c}{N} \frac{B}{B + 1 - p_c + \frac{1-\beta(N)+\alpha(N)(1-p_c)}{\beta(N)}}, \tag{10.14}$$

where $\alpha(N), \beta(N)$ are parameters depending only on the random walk transition matrix P.

Proof. Refer to Section C.2 of Appendix C for a proof of this result. ❑

10.6 Results of Simulation

In this section, the simulation results for the variation of the throughput with the buffer sizes and the grid sizes in both scenarios (ISD_2 and MSD_2) are presented. To study the variation with the size of buffers, we vary the buffer size of each node from 0 to 10, in a 20 × 20 network with 100 nodes. To study the variation with the size of the grid, the grid size is varied from 8 × 8 to 30 × 30, keeping the density $\frac{n}{N}$ of the nodes fixed at 0.5. For this simulation, the buffer size for each node is fixed at 8 packets. Finally, the variation of throughput with the number of nodes(n) is studied for a fixed grid size of 20×20 and a fixed buffer size of 8. We discuss the results obtained for each of the schemes in detail below.

Plots of the average throughput from analysis and simulations for the ISD_2 (two-hop immobile source and destination) case are shown in Figures 10.7, 10.8,and 10.9. It is seen that the simulation results closely matches the analytic plots, confirming with the claim of exactness of our analysis. Further, the parameters in Figure 10.7 are the same as those use for plotting Figure 10.1, and both depict the variation of throughput with buffer size. While the Poisson model failed, our model is very close to accuracy[2]. Figure 10.8 shows the variation of throughput with grid-size for a fixed density of nodes. The proposed Markov model clearly captures the effect of diminishing throughput. Therefore, for achieving the same throughput, the node density needs to be considerably increased to compensate for sparseness. Figure 10.9 shows the variation of the throughput with the node population n. The simulation confirms the fact that even though the total network throughput is increased, the per-node throughput actually decreases with increase in n due to contention effects.

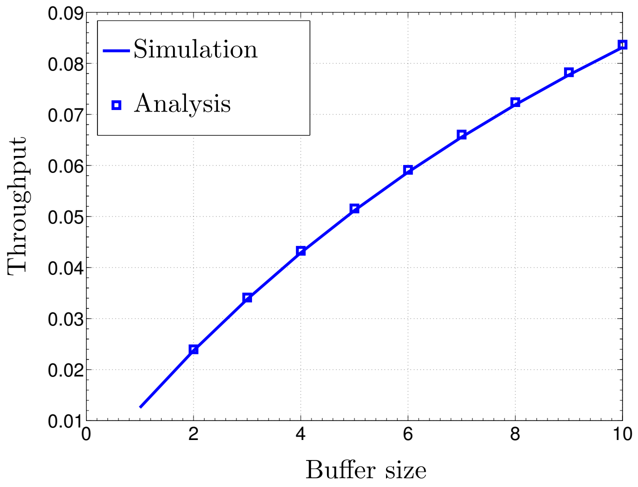

Figure 10.7: Throughput vs buffer size for the immobile source-destination case.

For the MSD_2 case (two-hop with mobile source and destination), the corresponding

[2] Note that the inaccuracy is only due to the approximate formula employed for $\gamma(N)$

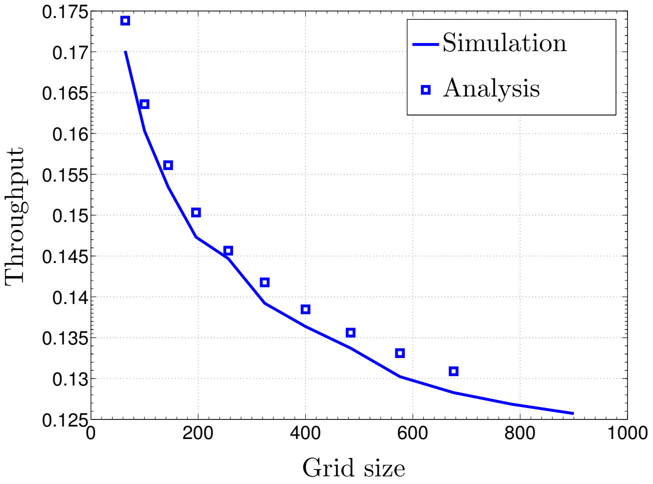

Figure 10.8: Throughput vs grid size for the immobile source-destination case.

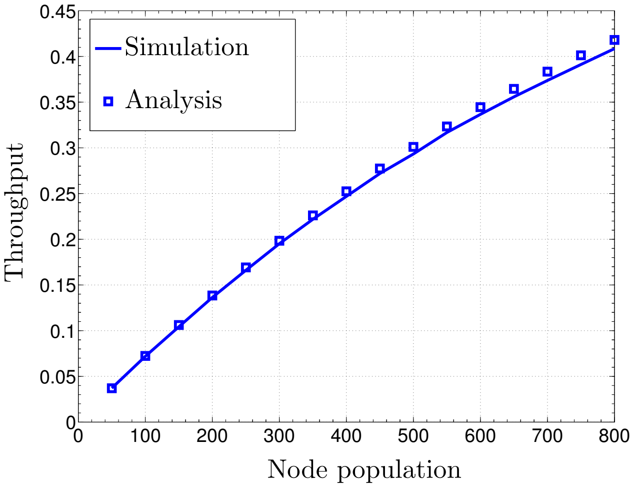

Figure 10.9: Throughput vs node population for the immobile source-destination case.

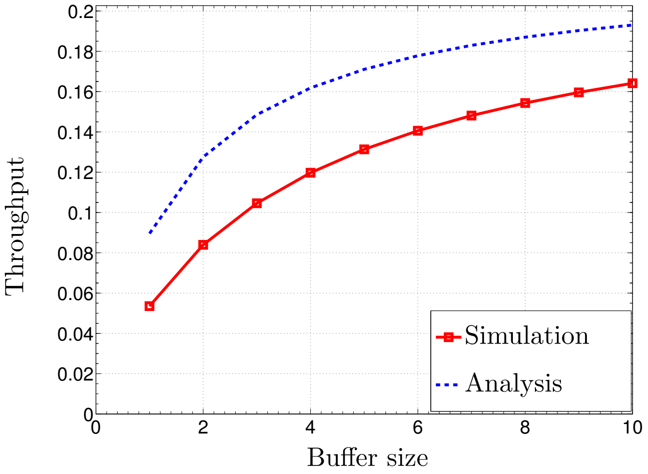

Figure 10.10: Throughput vs buffer size for the mobile source-destination case.

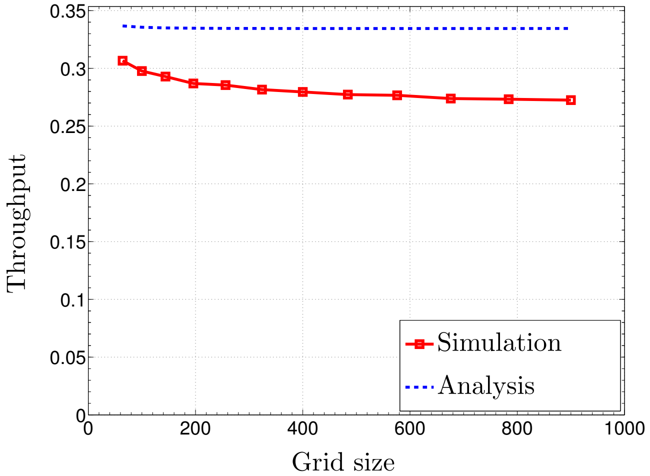

Figure 10.11: Throughput vs grid size for the mobile source-destination case.

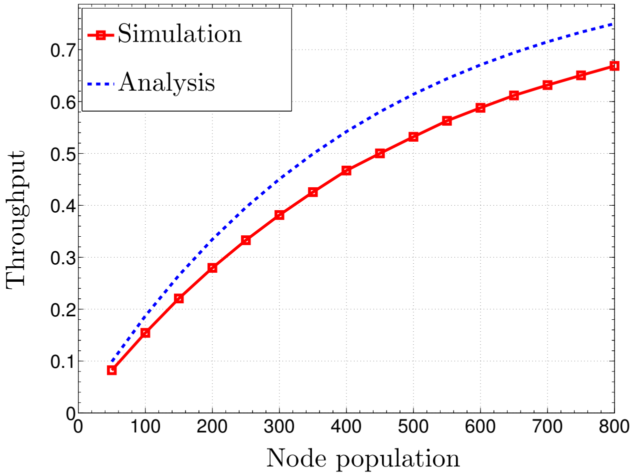

Figure 10.12: Throughput vs node population for the mobile source-destination case.

results for the same parameters as was used in the ISD_2 case are shown in Figures 10.10, 10.11, and 10.12. Here, the trends in general are similar to the ISD_2 case. However, the prediction error from analysis is slightly higher. This is a result of the partial independence assumption introduced in the analysis to keep the state space tractable. However, the partial independence assumption, unlike the Poisson contact assumption, tracks the throughput with a reasonable error margin (about $10 - 15\%$ depending on the buffer and node population). However, the prediction shows virtually no variation with the grid size in Figure 10.11. It must be noted that in this case, the node population density $\frac{n}{N}$ is kept constant. The apparent lack of variation can be explained as follows. In the analysis of throughput for the MSD_2 case, the dependence of throughput on N for fixed density is solely captured by the ratio $\frac{\alpha(N)}{\beta(N)}$, which can be shown to approach 1.25 asymptotically in N. However, from the actual simulations, this dependence decays as $\Theta(\frac{1}{\log N})$, as in the fixed $s - d$ case. Due to the convergence of this ratio, we fail to capture the actual trend of the system.

CHAPTER XI

CAPACITY OF NETWORKS WITH FINITE BUFFER

In networks, packets that have to be routed from one node to the other may have to be relayed through a series of intermediate nodes. Also, each node in the network may receive packets via many data streams that are being routed simultaneously from their source nodes to their respective destinations. In such conditions, the packets may have to be stored at intermediate nodes for transmission at a later time. If an unlimited buffer is available, the intermediate nodes need not have to reject or drop packets that arrive. However, often times, buffers are limited in size. While the analysis is interesting as a theoretical exercise in itself, under realistic models and assumptions, this problem is also of interest for efficient network design and congestion control.

An allied problem is that of buffer sizing and congestion control that is of paramount interest to router design engineers. Today, we have arrived at a juncture wherein the Internet handles a large volume of data. Typical routers today route several tens of gigabits of data each second [6, 125]. Realistic studies have shown that, at times, Internet routers handle about ten thousand independent streams/flows of data packets. With a reasonable buffer size of few Gigabytes of data, each stream can only be allocated a few tens of data packets. Therefore, at times when long parallel flows congest a router, the effects of such a small buffer space per flow come to play. Though our work is motivated partly by such concerns, our work is far from modeling realistic scenarios. This work modestly aims at providing a theoretical framework to understand the fundamental limits of single information flow in finite-buffer networks.

This chapter is organized thus. First, we present the formal definition of the problem and the network model in Section 11.1. Next, we investigate the capacity of line networks in Section 11.2. We present our proposed upper and lower bounds inspired from results in queueing theory followed by an estimate of the capacity that is based on certain simplifying assumptions. We then present our extensions to general wired networks in Section 11.3. Finally, Section 11.4 presents the results of simulations evaluating our analytic results in various network settings.

11.1 Problem Statement and Network Model

Throughout this chapter, we model the network by a directed graph $\overrightarrow{G}(V, \overrightarrow{E})$, where each node $n \in V$ has a buffer size of m_v packets and packets can be transmitted over a link $\overrightarrow{e} = (n, n')$ only from the node n to n'. The system is analyzed using a discrete-time model, where each node can transmit at most a single packet over a link in an epoch. Moreover, the network is assumed to be wired. Therefore, a node may potentially choose to send different packets over different outgoing links. The loss process on each link is assumed to be memoryless and that the loss process on different links are assumed to be independent. Therefore, by this model, a node will receive a packet on an incoming link when the neighboring node transmits a packet and when the packet is not erased over the link.

A class of simple networks that we will be considering in detail is the family of line networks, where nodes are arranged in a line. As illustrated in Figure 11.1, a line network is a directed graph of h hops with $V = \{v_0, v_1, \ldots, v_h\}$ and $\overrightarrow{E} = \{(v_i, v_{i+1}) : i = 0, \ldots, h-1\}$ for some integer $h \geq 2$. Moreover, for each $i = 0, \ldots, h-1$, the erasure probability on the link (v_i, v_{i+1}) is denoted by ε_i. The network illustrated in the figure has h hops and $h-1$ intermediate nodes. In the figure, the intermediate nodes are shown by black ones. Note that the buffer size for each node v_i is simply denoted by m_i (as opposed to m_{v_i}) and that the buffer size for different nodes can be different. Also, we let $\{X_i(l)\}_{\mathbb{Z}_{\geq 0}}$ to be the random process denoting the loss process on the link (v_{i-1}, v_i).

$$X_i(l) = \begin{cases} 1 & \text{if packet is succesfully transmitted over } (i-1, i) \text{ at epoch } l \\ 0 & \text{otherwise} \end{cases} . \qquad (11.1)$$

Additionally, we denote a general directed wired network by a directed network graph $\overrightarrow{G}(V, \overrightarrow{E})$ that has no directed cycles. A node $u \in V$ can communicate with $v \in V$ if and

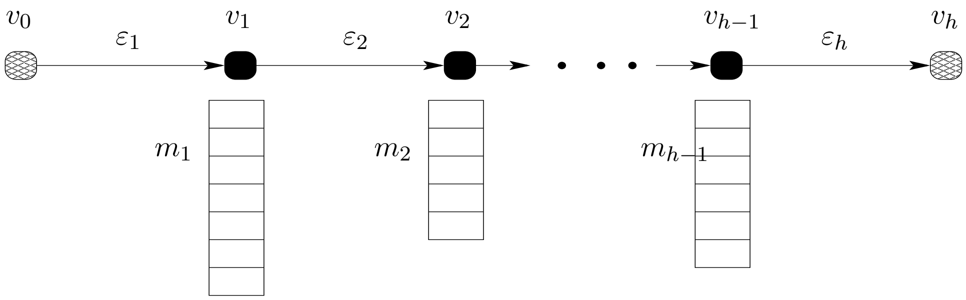

Figure 11.1: An illustration of the line network.

only if $(u, v) \in \overrightarrow{E}$. Just as in (11.1), for each $\overrightarrow{e} = (x, y) \in \overrightarrow{E}$, we have an associated erasure process $X_{xy}(l)$ denote the event that packet transmitted at the l^{th} epoch is either delivered or erased. The probability of erasure of a packet over the link $\overrightarrow{e} = (x, y) \in \overrightarrow{E}$ is denoted by $\varepsilon_{\overrightarrow{e}} = \varepsilon_{xy}$.

Finally, the problem that we interest ourselves is the maximum attainable information rate[1] that can be transmitted between a given pair of nodes in a network with the aforementioned model. Note that it is assumed the destination node has no buffer constraints and that the source node can generate innovative packets during each epoch. For instance, in the particular case of the line network illustrated in Figure 11.1, we would like to identify the maximum rate of information that the node v_0 can transmit to node v_h.

11.2 Capacity of Line Networks

In this section, we investigate the effect of finite buffers on the capacity of line networks. Since in a line network, no node has more than a single outgoing edge, the results of this section are equally applicable for both wireless and wired line networks. First, we present a framework for the exact computation of the capacity of line networks. We then present bounds on the capacity using techniques from queueing and graph theory. We conclude this section by presenting an effective means of approximating the capacity of a line network.

[1] To be precise, we are interested in finding the capacity denoted by *supremum* of all the rates achievable by all possible coding strategies.

11.2.1 Exact Computation of Capacity

The problem of identifying capacity is directly related to the problem of finding schemes that are *rate-optimal*, *i.e.*, schemes whose rate cannot be surpassed by any other scheme. For a line network, an optimal scheme can be seen to satisfy the following property.

- Each node uses every opportunity to convey a packet that the incident outgoing edge allows to transmit a packet.

Equivalently, if an Oracle were to announce all the future epochs during which the channel would not erase the packet provided the node were to transmit at those instances, then the channel must transmit a packet at these times provided it has a packet of information to transmit then. In the presence of lossless feedback from the next hop, the optimal scheme must perform these in the corresponding order.

1. If the buffer of a node is not empty at a particular epoch, then it must transmit at least one of the packets at that time.
2. A node deletes the packet transmitted at an epoch if it receives an acknowledgement from the next hop at that epoch.
3. A node accepts an arriving packet if it has space in its buffer and sends an acknowledgment to the previous node.

In the absence of feedback, rate-optimality can be achieved by employing network coding based on random linear combinations in a finite field $\mathbb{F}_q$ of large size q as is described in [47]. The optimal scheme can be described as follows.

1. At each epoch, a node having a buffer size of m selects a vector $\mathbf{a}$ uniformly at random from $\mathbb{F}_q^m$ and uses its components to generate a random linear combination in the following manner. For each buffer slot $i \in \{1, \ldots, m\}$, the packet P_i stored in that slot is represented as a vector over $\mathbb{F}_q$ and the output packet is generated by computing $\sum_{i=1}^{m} a_i P_i$. This generated packet is then transmitted during the epoch.

2. If a packet P is received by a node at an epoch, it first generates the output packet at that instant and then updates its buffer in the following manner. It selects a vector $\mathbf{b} \in \mathbb{F}_q^m$ uniformly at random and for each $i \in \{1, \ldots, m\}$, adds the packet $b_i P$ to the packet stored in the i^{th} buffer slot.

Note that in the network coding scheme described above, after sufficient time after the commencement of packet transfer from the source, the buffer slots of all the nodes will always have packets unlike the scheme with feedback. However, it is not true that all of these packets are *innovative*, *i.e.*, packets may contain common information. Such a condition may occur when the packets are linearly dependent in the algebraic sense. Another point that must be highlighted is the fact that for a given set of erasures $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_h)$ and buffer sizes $\mathcal{M} = (m_1, \ldots, m_{h-1})$, the information rate $\mathcal{C}(\mathcal{E}, \mathcal{M}; \mathbb{F}_q)$ achieved by the network coding scheme over the field $\mathbb{F}_q$ and the information rate $\mathcal{C}(\mathcal{E}, \mathcal{M}; \infty)$ in the presence of lossless feedback are related in the following manner. For each sequence of finite fields $\{\mathbb{F}_{q_l}\}_{l \in \mathbb{N}}$ such that $|\mathbb{F}_{q_l}| \to \infty$, we have

$$\lim_{l \to \infty} \mathcal{C}(\mathcal{E}, \mathcal{M}; \mathbb{F}_{q_l}) = \mathcal{C}(\mathcal{E}, \mathcal{M}; \infty). \tag{11.2}$$

From (11.2) we observe that in order to identify the capacity of line networks without feedback, we can analyze the network with lossless feedback. Therefore, we shall henceforth be solely interested in identifying $\mathcal{C}(\mathcal{E}, \mathcal{M}; \infty)$ given the set of erasures and buffer sizes.

In order to model the network with lossless feedback, we need to track the number of packets that each node possesses at every instant of time. We do so by using the rules of buffer update under the optimal scheme. Let $\mathbf{n}(l) = (n_1(l), \ldots, n_{h-1}(l))$ be the vector whose i^{th} component denotes the number of packets the i^{th} intermediate node possesses at time l. The variation of state at the l^{th} can be tracked using auxiliary random variables $Y_i(l)$ defined by

$$Y_i(l) = \begin{cases} \sigma(n_{i-1}(l)) X_i(l) & i = h \\ \sigma(n_{i-1}(l)) X_i(l) \sigma(m_i - n_i(l) + Y_{i+1}(l)) & i = 2, \ldots, h-1 \\ X_i(l) \sigma(m_i - n_i(l) + Y_{i+1}(l)) & i = 1 \end{cases}. \tag{11.3}$$

From the definition of the auxiliary binary random variables in (11.3), we see that $Y_i(l) = 1$ only if all the following three conditions are met.

1. Node v_{i-1} has a packet to transmit to v_i.
2. The link $(i-1, i)$ does not erase the packet at the l^{th} epoch, *i.e.*, $X_i(l) = 1$.
3. Node v_i has space after it has updated its buffer for any changes due to its transmission at that epoch.

The changes in the buffer states can then by seen to be given by the following.

$$n_i(l+1) = n_i(l) + Y_i(l) - Y_{i+1}(l), \qquad i = 1, \ldots, h-1. \tag{11.4}$$

Note that since $\mathbf{Y}(l) = (Y_1(l), \ldots, Y_h(l))$ is a function of $\mathbf{n}(l)$ and $\mathbf{X}(l) = (X_1(l), \ldots, X_h(l))$, $\mathbf{n}(l+1)$ depends only on its previous state $\mathbf{n}(l)$ and the channel conditions $\mathbf{X}(l)$ at the l^{th} epoch. Hence, we see that $\{\mathbf{n}(l)\}_{l \in \mathbb{Z}_{\geq 0}}$ forms a Markov chain with $\prod_{i=1}^{h-1}(m_i + 1)$ states. However, at each time instant, the number of packets that can be transmitted over a channel is bounded by unity, we see that for every $i = 1, \ldots, h-1$ and $l \in \mathbb{Z}_{\geq 0}$,

$$0 \leq Y_i(l) \leq 1 \quad \text{and} \quad |n_i(l+1) - n_i(l)| \leq 1. \tag{11.5}$$

Therefore, the number of non-zero entries in each row of the probability transition matrix[2] $P(\mathcal{E}, \mathcal{M})$ is bounded above by $\min(3^{h-1}, \prod_{i=1}^{h-1}(m_i + 1))$. A more detailed structural categorization can be performed thus. Let $\preceq$ be a total ordering on the set of states $\mathcal{S}$ of the Markov chain such that states $\mathbf{s} \preceq \mathbf{s}'$ if and only if $\mathbf{s}'$ can be reached from $\mathbf{s}'$ without use of any transition that involves a packet to successfully leave from the last node v_{h-1}. Without loss of generality, we may assume that this is the ordering of states used to define the transition matrix[3] $P(\mathcal{E}, \mathcal{M})$. Denote T_ι to be the set of states that have $n_{h-1}(\cdot) = \iota$ for $\iota = 0, \ldots, m_{h-1}$. Let $\Gamma_\iota^-, \Lambda_\iota, \Gamma_\iota^+$ represent the transition between states in T_ι and those in $T_{\iota-1}$, T_ι, $T_{\iota+1}$, respectively. Then, it can be shown that $\Gamma_i^+ = \Gamma^+$, $\Lambda_i = \Lambda$, and $\Gamma_i^- = \Gamma^-$ for

[2] The ij^{th} term of the matrix $P(\mathcal{E}, \mathcal{M})$ represents the probability that the next state is j given that state is presently i.

[3] In such an ordering, the row in the transition matrix P corresponding to a state $\mathbf{s} \in \mathcal{S}$ is given by $1 + s_1 + \sum_{i=2}^{h-1} s_i \prod_{j=1}^{i-1}(m_j + 1)$.

$\iota = 1, \ldots, m_{h-1} - 1$ and that the dynamics of the network can be represented as follows.

$$P(\mathcal{E}, \mathcal{M}) = \begin{pmatrix} \Lambda_0 & \Gamma_0^+ & \mathbf{0} & & & \cdots & & \mathbf{0} \\ \Gamma^- & \Lambda & \Gamma^+ & \mathbf{0} & & \cdots & & \mathbf{0} \\ \mathbf{0} & \Gamma^- & \Lambda & \Gamma^+ & \mathbf{0} & \cdots & & \mathbf{0} \\ & & & & \vdots & & & \\ \mathbf{0} & & \cdots & & \mathbf{0} & \Gamma^- & \Lambda & \Gamma^+ \\ \mathbf{0} & & \cdots & & & \mathbf{0} & \Gamma^-_{m_{h-1}} & \Lambda_{m_{h-1}} \end{pmatrix}. \tag{11.6}$$

The dynamics of the network given in (11.6) is represented pictorially by the chain in Figure 11.2. Note that due to the finite buffer condition, the submatrix of P corresponding to transitions from T_{m_h-1} are different from the submatrix corresponding to transition from states in T_i for $i = 1, \ldots, m_{h-1} - 1$. Similar is the case with the transition sub-matrix corresponding to transitions from T_0. However, the latter is also true when the buffer size is not limited. Additionally, it must be mentioned that for $i = 0, \ldots, m_{h-1}$, the transition of states between any two of states within T_i can be organized in a way similar to the one depicted in the figure. In addition to the above structural properties, the matrices themselves have the following algebraic properties.

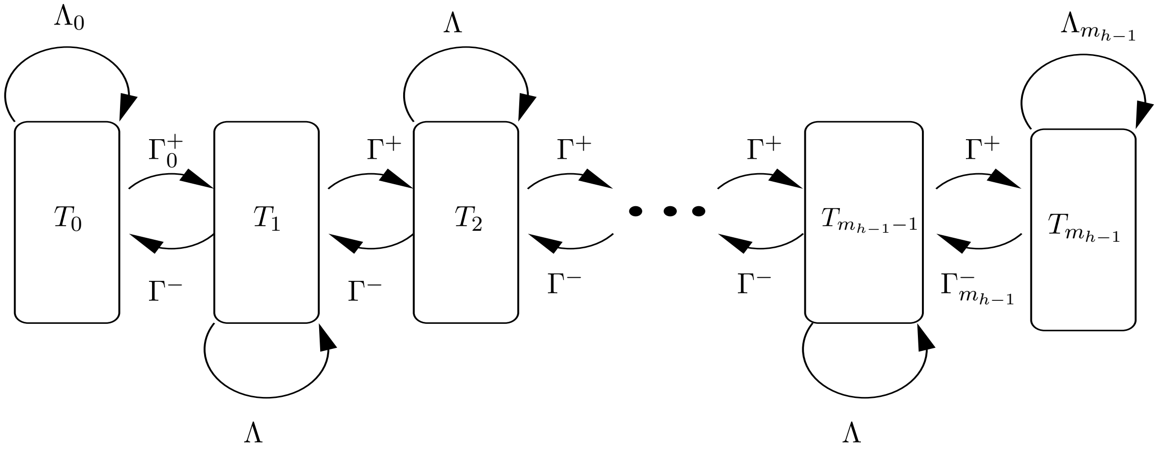

Figure 11.2: An illustration of the Markov chain modeling the dynamics of the line network when all the buffer sizes are equal.

Lemma 11.2.1. *In a generic line network, the following hold*[4]*.*

[4] *We indicate the indices for* Λ, Γ^+ *and* Γ^- *matrices just for the sake of clarity.*

a. For $h > 2$, Γ_i^-, $i = 1, \ldots, m_{h-1}$ are non-singular and upper triangular

b. For $h > 2$, Γ_i^+, $i = 0, \ldots, m_{h-1} - 1$ are singular and lower triangular.

c. $I - \Lambda_i$, $i = 0, \ldots, m_{h-1}$ are non-singular.

The proof of these properties can be found in Section D.1 of Appendix D.

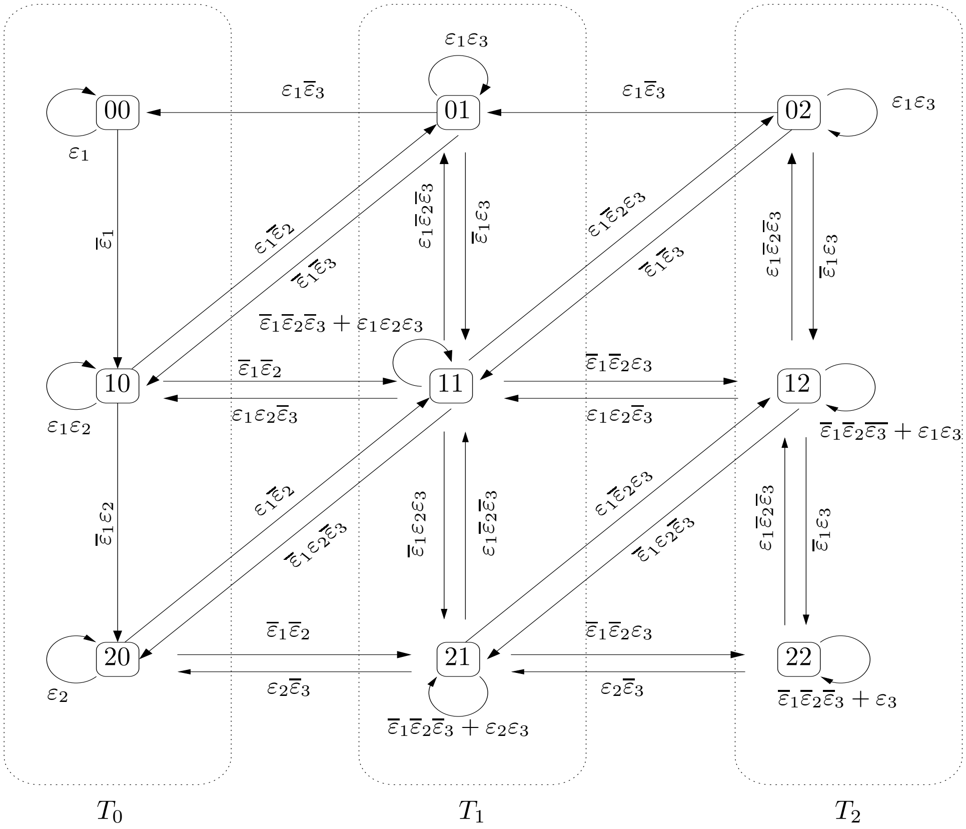
$\varepsilon_1\varepsilon_3$
$\varepsilon_1\overline{\varepsilon}_3$
$\varepsilon_1\overline{\varepsilon}_3$
$\varepsilon_1\varepsilon_3$
00
01
02
ε_1
$\overline{\varepsilon}_1$
$\varepsilon_1\overline{\varepsilon}_2$
$\overline{\varepsilon}_1\overline{\varepsilon}_3$
$\varepsilon_1\overline{\varepsilon}_2\overline{\varepsilon}_3$
$\overline{\varepsilon}_1\varepsilon_3$
$\varepsilon_1\overline{\varepsilon}_2\varepsilon_3$
$\overline{\varepsilon}_1\overline{\varepsilon}_3$
$\varepsilon_1\overline{\varepsilon}_2\overline{\varepsilon}_3$
$\overline{\varepsilon}_1\varepsilon_3$
$\overline{\varepsilon}_1\overline{\varepsilon}_2\overline{\varepsilon}_3 + \varepsilon_1\varepsilon_2\varepsilon_3$
$\overline{\varepsilon}_1\overline{\varepsilon}_2$
$\overline{\varepsilon}_1\overline{\varepsilon}_2\varepsilon_3$
10
11
12
$\varepsilon_1\varepsilon_2\overline{\varepsilon}_3$
$\varepsilon_1\varepsilon_2\overline{\varepsilon}_3$
$\varepsilon_1\varepsilon_2$
$\overline{\varepsilon}_1\overline{\varepsilon}_2\overline{\varepsilon_3} + \varepsilon_1\varepsilon_3$
$\overline{\varepsilon}_1\varepsilon_2$
$\varepsilon_1\overline{\varepsilon}_2$
$\overline{\varepsilon}_1\varepsilon_2\overline{\varepsilon}_3$
$\overline{\varepsilon}_1\varepsilon_2\varepsilon_3$
$\varepsilon_1\overline{\varepsilon}_2\overline{\varepsilon}_3$
$\varepsilon_1\overline{\varepsilon}_2\varepsilon_3$
$\overline{\varepsilon}_1\varepsilon_2\overline{\varepsilon}_3$
$\varepsilon_1\overline{\varepsilon}_2\overline{\varepsilon}_3$
$\overline{\varepsilon}_1\varepsilon_3$
$\overline{\varepsilon}_1\overline{\varepsilon}_2$
$\overline{\varepsilon}_1\overline{\varepsilon}_2\varepsilon_3$
20
21
22
ε_2
$\varepsilon_2\overline{\varepsilon}_3$
$\varepsilon_2\overline{\varepsilon}_3$
$\overline{\varepsilon}_1\overline{\varepsilon}_2\overline{\varepsilon}_3 + \varepsilon_3$
$\overline{\varepsilon}_1\overline{\varepsilon}_2\overline{\varepsilon}_3 + \varepsilon_2\varepsilon_3$
T_0
T_1
T_2

Figure 11.3: Markov chain for a line network of three hops with erasure probabilities $\varepsilon_1, \varepsilon_2, \varepsilon_3$ and intermediate nodes having a buffer size of two packets each.

To illustrate the concept of the above lemma, consider the Markov chain for a line network with three hops, with erasure probabilities $\mathcal{E} = (\varepsilon_0, \varepsilon_1, \varepsilon_2)$, and with buffer sizes $\mathcal{M} = (2, 2)$ presented in Figure 11.3. The algebraic properties detailed in Lemma 11.2.1 can be understood from Figure 11.3 in the following way.

1. Any transition involving a decrease in the second component involves a non-negative change in the magnitude of the first component.

2. Any horizontal transition involving a decrease in the second component is always feasible.

3. Any transition involving an increase in the second component involves a non-positive change in the magnitude of the first component.

4. Not all horizontal transition involving an increase in the second component is feasible. For example, the transitions from the state $(0, 0)$ to $(0, 1)$ and from the state $(0, 1)$ to $(0, 2)$ are infeasible, and hence $(\Gamma_0^+)_{11} = (\Gamma_1^+)_{11} = 0$.

While the first two facts relate to the upper triangular structure and non-singularity of Γ^-, the latter two relate to the lower triangular and singularity properties of Γ^+. The Markov chain for the dynamics of the state of the line network with lossless feedback is *irreducible, aperiodic, positive-recurrent*, and *ergodic* [80, 29]. By ergodicity, we can obtain temporal averages by statistical averages and therefore have the following. We then see that the computation of capacity is equivalent to the computation of the likelihood of the event that at a particular instant, the system is in a state where the buffer of the last node is non-empty and the last-hop link does not erase the packet at that instant. This quantity is given by

$$\begin{aligned} \mathcal{C}(\mathcal{E}, \mathcal{M}; \infty) &= (1 - \varepsilon_h) \Pr[\{\mathbf{s} \in \mathcal{S} : s_{h-1} > 0\}] \qquad (11.7) \\ &= (1 - \varepsilon_i) \Pr[\{\mathbf{s} \in \mathcal{S} : (s_i > 0) \wedge (s_{i+1} < m_{i+1})\}], \quad i = 1, \ldots, h-2. \qquad (11.8) \end{aligned}$$

The problem of identifying the capacity of line networks is reduced to the problem of identifying the steady-state probabilities of the aforementioned Markov chain. However, the size of the Markov chain and the multiple reflections due to the finiteness of buffers at each intermediate node make this problem mathematically intractable for even networks of reasonable hop-lengths and buffer size. We therefore content ourselves by estimating the capacity of line networks.

11.2.2 Bounds on the Capacity of Line Networks

The problem of identifying the steady-state probability of similar networks have been approximated in queueing theory. The analogue of these line networks in queueing theory is an open network of queues connected in tandem where the number of packets in the system can vary with time. The event of packet dropping due to full-buffer condition at nodes is modeled as blocking after service[5]. Most approaches to this problem has been to approximate the dynamics of the network by selecting a node and the incident edges incident on it, one at a time. In this work, we take a discrete epoch-based approach that is similar to those in [4, 3, 16, 112, 22, 104].

The key idea in this section is to investigate the Markov chain introduced in the previous section to identify an approximation for the dynamics of the chain that allows a node-by-node analysis for deriving useful bounds. The main reason for the intractability of the exact system is the strong dependence of $Y_i(l)$ on not only $Y_{i-1}(l)$, but also $Y_{i+1}(l)$. This dependence translates to a strong dependence of $n_i(l)$ on both $n_{i-1}(l)$ and $n_{i+1}(l)$. However, we notice that this dependence of the i^{th} state on the $i+1^{\text{th}}$ state would be ignored if we assume that the next hop sends an acknowledgement whenever it receives a packet as opposed to the original setup when it sends an acknowledgment when a packet is received and stored. Making this assumption would result in a simplified Markov chain[6] wherein the state updates are given by the following rule for all $l \in \mathbb{Z}_{\geq 0}$.

$$\tilde{n}_i(l+1) = \begin{cases} \tilde{n}_i(l) + \sigma(\tilde{n}_{i-1}(l))X_i(l)\sigma(m_i - \tilde{n}_i(l)) - \sigma(\tilde{n}_i(l))X_{i+1}(l) & 1 < i < h \\ \tilde{n}_i(l) + X_i(l)\sigma(m_i - \tilde{n}_i(l)) - \sigma(\tilde{n}_i(l))X_{i+1}(l) & i = 1 \end{cases}. \tag{11.9}$$

To avoid confusion, we appellate the chain that is obtained by the dynamics defined by (11.3) and (11.4) as the Exact Markov Chain (EMC) and the one defined by (11.9) as the Approximate Markov Chain (AMC). Also, we allow $\mathbf{n}(l)$ and $\tilde{\mathbf{n}}(l)$ to always denote the state of an instance of the process generated by the EMC and the AMC, respectively. The following properties hold for these two chains.

Theorem 11.2.1. (*Temporal Boundedness Property of AMC*) *Consider a line network with h hops and an instance of the channel realizations for all time $\{X_i(l) : i = 1, \dots, h\}_{l \in \mathbb{Z}_{\geq 0}}$. Suppose we track the variation of the states of the EMC and the AMC using this instance*

[5] This analogue is exact since the random duration taken for a packet to be successfully transmitted over the channel can be viewed as the *service time* of the server.

[6] This approximation is equivalent to assuming that blocking results in packet loss unlike the exact setup where it gets re-serviced.

of channel realizations and with the same initial state $\mathbf{n}(0) = \tilde{\mathbf{n}}(0)$*. Then, for any* $l \in \mathbb{Z}_{\geq 0}$ *and* $i = 1, \ldots, h-1$*, the following holds.*

$$n_i(l) \geq \tilde{n}_i(l). \tag{11.10}$$

The proof of the above Temporal Boundedness Property of the AMC can be found in Section D.2 of Appendix D. The Temporal Boundedness property guarantees that statistically, the probability that a node has an empty buffer is overestimated by the AMC. In fact, if we can identify the steady-state distribution of the states of AMC, we can provide a lower bound for the steady-state probability of any subset of states $\mathcal{A} \subseteq \mathcal{S}$ that have the form

$$\mathcal{A} = \{\mathbf{s} \in \mathcal{S} : (s_j \geq a_j)\},\ j \in \{1, \ldots h-1\}. \tag{11.11}$$

Using the Temporal Boundedness property in conjunction with (11.7), we can provide a lower bound $\underline{\mathcal{C}}(\mathcal{E}, \mathcal{M}; \infty)$ for the capacity of the line network by underestimating the probability in (11.7) by using the steady-state distribution of the AMC. Equivalently, the capacity of the line network is at least that of the throughput achievable from the EMC.

Thus, the problem of identifying capacity is reduced to identifying the steady-state probability of AMC. Since the update equation for the AMC allows for a node-by-node analysis, we can identify the steady-state distribution in a hop-by-hop fashion. To identify the capacity of approximated by the AMC, we need to track the statistics of the random process of packet departure at each node in the network. The following result enables us to exactly tract the inter-departure times after each hop.

Theorem 11.2.2. *Consider a discrete-time queuing system where a server with* m *customer slots is fed by an arrival process whose distribution of inter-arrival times is given by* $f = \sum_{i=1}^{N-1} p_i \mathbb{G}(\mathbb{t}_i)$*. Suppose that the distribution*[7] *of service time is* $\mathbb{G}(\mathbb{t}_N)$ *with* $\mathbb{t}_i \neq \mathbb{t}_j$ *for* $1 \leq i < j \leq N$*. Then, the distribution of inter-departure times* $\Upsilon(f, m, \mathbb{t}_N)$ *is given also a weighted-sum of geometric distributions,* i.e.*,* $\exists\, p'_1, \ldots, p'_N \in \mathbb{R}$*, such that*

$$\Upsilon(f, m, \mathbb{t}_N) = \sum_{i=1}^{N} p'_i \mathbb{G}(\mathbb{t}_i) \tag{11.12}$$

[7] *Note that we do not require that all* $p_i s$ *or all* $p'_i s$ *be positive. We only need that their sums be unity and that they generate a valid probability distributions, respectively.*

A detailed analysis including the means for exact computation for the weights $\{p'_i : i = 1, \ldots, N\}$ is presented in Appendix E. Note that the above result and the discussion in Appendix E are motivated for networks with distinct channel erasure probabilities. This additional assumption is made simply for the sake of convenience. Since the capacity of line networks is a continuous function of the channel loss parameters, the capacity of a line network with non-distinct loss parameters can be approximated to any order of precision by that of a line network with distinct channel loss probabilities.

We can identify the distribution of inter-departure times of the departure process at the last node by iteratively feeding the departure process[8] from each node as the arrival process to the next node in the network. The distribution of inter-departure of packets f_h from the last intermediate node v_{h-1}, which is also the distribution of the inter-arrival times as seen by the node v_h is then given by

$$f_i = \begin{cases} \Upsilon(f_{i-1}, m_{i-1}, \varepsilon_i) & i = 2, \ldots, h \\ \mathbb{G}(\varepsilon_1) & i = 1 \end{cases} . \tag{11.13}$$

Finally, we notice that the lower estimate $\underline{\mathcal{C}}(\mathcal{E}, \mathcal{M}; \infty)$ for the capacity $\mathcal{C}(\mathcal{E}, \mathcal{M}; \infty)$ of the line network with erasures $\mathcal{E}$ and buffer sizes $\mathcal{M}$ using the AMC can be derived from the distribution of inter-arrival times of packets $f_h = \sum_{l=1}^{h} c_l \mathbb{G}(\varepsilon_l)$ by

$$\underline{\mathcal{C}}(\mathcal{E}, \mathcal{M}; \infty) = \frac{1}{\langle f_h \rangle} = \frac{1}{\sum_{l=0}^{h} \frac{c_l}{1-\varepsilon_l}}. \tag{11.14}$$

Note that to compute the coefficients c_l for a network, we need to perform the computations described in Appendix E. This above idea of lower bound can be extended easily to an upper bound using the following result. The fundamental idea behind the following bound is to intelligently manipulate the memory sizes of each buffer so that the packet drop in the modified network is provably smaller than that in the actual network.

Theorem 11.2.3. *Consider a line network with distinct erasure probabilities $\mathcal{E}$ and buffer*

[8] Note that this is accurate only if the departure process from the queue defined in Theorem 11.2.2 is a renewal process. However, extensive simulations over a wide range of channel erasure probabilities and buffer constraints strongly agree with the theoretical computations assuming that the process is renewal.

sizes $\mathcal{M}$. Define a sequence of distributions $\{g_i\}_{i=1}^h$ by

$$g_i = \begin{cases} \Upsilon(g_{i-1}, \sum_{l=1}^{i-1} m_l, \varepsilon_i) & i = 2, \ldots, h \\ \mathbb{G}(\varepsilon_1) & i = 1 \end{cases} . \tag{11.15}$$

Then, from Theorem 11.2.2, we see that $\exists\, d_1, \ldots, d_h \in \mathbb{R}$ such that $f_h = \sum_{l=1}^h d_l \mathbb{G}(\varepsilon_l)$. Therefore,

$$\overline{\mathcal{C}}(\mathcal{E}, \mathcal{M}; \infty) = \frac{1}{\langle g_h \rangle} = \frac{1}{\sum_{l=0}^h \frac{d_l}{1-\varepsilon_l}}. \tag{11.16}$$

is an upper bound on the capacity of the line network.

An outline of the proof of this result can be found in Appendix D.4.

As an illustration of the above bounds, consider a simple three hop network with erasures $\mathcal{E} = (0.5, 0.7, 0.8)$ and buffer sizes $\mathcal{M} = (5, 5, 5)$. Then, from Appendix E, we see that the lower bound is computed by constructing f_i for $i = 1, 2, 3$. These are given by

$$f_1 = \mathbb{G}(0.5) \tag{11.17}$$

$$f_2 = -1.4670 \times 10^{-2}\mathbb{G}(0.5) + 1.01467\mathbb{G}(0.7) \tag{11.18}$$

$$f_3 = 1.2065 \times 10^{-4}\mathbb{G}(0.5) + -7.4451 \times 10^{-2}\mathbb{G}(0.7) + 1.07433\mathbb{G}(0.8). \tag{11.19}$$

From the above, we can compute the lower bound of capacity as

$$\underline{\mathcal{C}}(\mathcal{E}, \mathcal{M}; \infty) = \frac{1.2065 \times 10^{-4}}{1 - 0.5} + \frac{-7.4451 \times 10^{-2}}{1 - 0.7} + \frac{1.07433}{1 - 0.8} = 0.19517. \tag{11.20}$$

By a similar argument, one can show that

$$\overline{\mathcal{C}}(\mathcal{E}, \mathcal{M}; \infty) = \frac{7.8412 \times 10^{-6}}{1 - 0.5} + \frac{-4.8557 \times 10^{-3}}{1 - 0.7} + \frac{1.004847}{1 - 0.8} = 0.19967. \tag{11.21}$$

11.2.3 Iterative Estimation of the Capacity of Line Networks

In this section, we present an iterative estimate for the capacity of line networks that is based on some simplifying assumptions regarding the EMC[9]. We notice that the difficulty of exactly identifying the steady-state probabilities of the EMC stems from the finite buffer condition that is imposed. The finite buffer condition introduces a strong dependency of

[9] The conception of this idea was done in collaboration with Nima Torabkhani.

state update at a node on the state of the node that is downstream. This effect is caused by blocking when the state of a node is forced to remain as it is because the packet that it transmitted was successfully delivered to the next node, but the latter could not store the packed due to full buffer occupancy. Additionally, the non-tractability of the EMC is compounded by a non-memoryless output process at each node. In this section, we ignore some of these issues to develop an iterative estimation method that considers the effect of blocking albeit with some simplifying assumptions.

To develop an iterative technique, we assume the following.

1. The packets are ejected from nodes in a memoryless fashion. Equivalently, we assume that $\Pr[(n_{i-1}(t) > 0) \wedge (X_i(t) = 1)|n_i(t) = k]$ does not vary with the occupancy k of the i^{th} node. This assumption allows us to keep track of only the information rate and not the exact statistics.

2. The blocking event occurs independent of the state of a node, *i.e.*, $\Pr[(Y_{i+1}(l) = 0) \wedge (X_{i+1}(l) = 1)|(n_i(t) = k)]$ is the same for $k = 1, \ldots, m_i$. This allows us to track just the blocking probability and not the joint statistics.

3. At any epoch, given the occupancy of a particular node, the arrival rate and the event blocking are independent of each other.

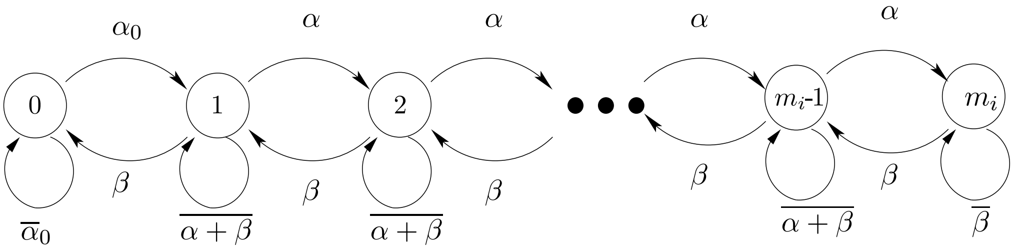

Figure 11.4: The Markov chain for the node v_i obtained by the simplifying assumptions.

These assumptions spread the effect of blocking equally over all non-zero states of occupancy at each node. Given that the arrival rate of packets at the node v_i is r_i packets/epoch, and that the probability of the next node is $p_{b_{i+1}}$, we can show that dynamics of the state change for the node v_i is given by the Markov chain depicted in Figure 11.4 with the parameters set to the following.

$$\begin{array}{rcl} \alpha & = & r_i(\varepsilon_{i+1} + \overline{\varepsilon}_{i+1} p_{bi+1}) \\ \beta & = & (1 - r_i)\overline{p_b}_{i+1}\overline{\varepsilon}_{i+1} \qquad . \\ \alpha_0 & = & r_i \end{array} \tag{11.22}$$

Also, the steady-state distribution can be computed to be

$$\Pr[n_i = k] = \varphi(k|r_i, \varepsilon_{i+1}, p_{bi+1}) \triangleq \begin{cases} \frac{1}{1+\frac{\alpha_0}{\beta}\left(\sum_{l=0}^{m_i-1} \frac{\alpha^l}{\beta^l}\right)} & k = 0 \\ \frac{\frac{\alpha_0 \alpha^{k-1}}{\beta^k}}{1+\frac{\alpha_0}{\beta}\left(\sum_{l=0}^{m_i-1} \frac{\alpha^l}{\beta^l}\right)} & 0 < k \leq m_i \end{cases} . \tag{11.23}$$

The blocking probability that the node v_{i-1} perceives from the node v_i assuming that v_i sees an arrival rate of r_i form v_{i-1} and a blocking probability of p_{bi+1} caused by v_{i+1} [10] can then be calculated from (11.23) as follows.

$$p_{bi} = \begin{cases} (\varepsilon_{i+1} + \overline{\varepsilon}_{i+1} p_{bi+1})\varphi(m_i|r_i, \varepsilon_{i+1}, p_{bi+1}) & i < h-1 \\ \varepsilon_{i+1}\varphi(m_i|r_i, \varepsilon_{i+1}, 0) & i = h-1 \end{cases} . \tag{11.24}$$

Similarly, (11.23) can be used to compute the arrival rate at the next node using the following.

$$r_{i+1} = \begin{cases} \overline{\varepsilon}_i(1 - \varphi(0|r_i, \varepsilon_{i+1}, p_{bi+1})) & 1 < i < h-1 \\ \overline{\varepsilon}_i(1 - \varphi(0|r_i, \varepsilon_{i+1}, 0)) & i = h-1 \end{cases} . \tag{11.25}$$

Given two vectors $\mathbf{r} = (r_1, \ldots, r_h) \in [0,1]^h$ and $\mathbf{p_b} = (p_{b1}, \ldots, p_{bh}) \in [0,1]^h$, we term $(\mathbf{r}, \mathbf{p_b})$ as an approximate solution to EMC, if they satisfy the equations (11.23), (11.24), and (11.25) in addition to having $r_1 = \overline{\varepsilon}_1$ and $p_{bh} = 0$. Fortunately, the following result guarantees both the uniqueness and the method of identifying the approximate solution to the EMC.

Theorem 11.2.4. *Given a line network with link erasures $\mathcal{E} = (\varepsilon_1, \ldots, \varepsilon_h)$ and intermediate node buffer sizes $\mathcal{M} = (m_1, \ldots, m_{h-1})$, there is exactly one approximate solution $(\mathbf{r}^*(\mathcal{E}, \mathcal{M}), \mathbf{p_b}^*(\mathcal{E}, \mathcal{M}))$ to the EMC. Moreover, this solution can be found iteratively.*

Details of the proof of uniqueness and the means to find the solution is presented in Section D.3 of Appendix D. Finally, the estimate of the capacity can be obtained from the

[10] Note that the arrival rate from the node v_1 is $r_1 = \overline{\varepsilon}_1$ and that the blocking probability of v_h is zero.

approximate solution using the following definition[11].

$$\mathcal{C}^*(\mathcal{E}, \mathcal{M}) = r_h^*(1 - p_{b_h^*}) = r_h^*. \tag{11.26}$$

11.3 Capacity of General Wired Acyclic Directed Networks

The capacity of wired acyclic directed networks is significantly more complicated than that of line networks. The presence of multiple incoming edges and multiple outgoing edges allows for packets to be replicated over outgoing links. Similarly, it is possible for a node to receive multiple copies of the same packet during different epochs. In such networks, it seems that simple replication can effect faster delivery of a packet. However, multiple copies of the same packet use up more buffer space and hence consume more network resource. If a copy of a packet is delivered, it would be ideal if all the other copies are erased instantaneously. However, this is not realistic even in networks that have lossless hop-by-hop feedback. Therefore, a careful addition of redundancy is necessary to maintain a good balance of reliability and resource usage. The following discussion motivates network coding [47] as an ideal way of achieving capacity by discussing some of the properties that an optimal scheme must possess.

11.3.1 Rate-Optimal Scheme for Wired Acyclic Directed Networks

Any optimal strategy for such networks must transmit packets *opportunistically.* In other words, all nodes must transmit packets at every epoch to each of its neighbors. Moreover, if possible, at each epoch, it must try to convey as much "information" as possible *i.e.,* statistically, the joint entropy of the packets sent on each of the links must be maximized. If the optimal strategy involves replication, then it must favor the transmission of packets that are not yet delivered over copies of delivered packets.

Consider the *random linear network coding* (RLNC) described in Section 11.2.1 with the following changes. For each epoch l and for each link $(u, v) \in \overrightarrow{E}$, a different random linear combination is computed and the resultant packet is transmitted from u to v at the l^{th} epoch. Also, once the generation for every outgoing link is completed at an epoch,

[11] See (D.20) in Appendix D.3 for alternate definitions for computing the capacity.

different random linear combinations of the arriving packets are added to each of its buffer locations. Such a scheme is easily seen to satisfy the above criteria for optimality and hence is a suitable candidate[12]. We conjecture that RLNC does in fact attain the maximum achievable rate of information flow in such networks.

In a sense, the RLNC strategy may alternatively viewed as performing the following. Each node attempts to transmit packets on its outgoing links that are linearly independent of each other. At each instant, every node deletes those packets it possesses that are (algebraically) linearly dependent on the set of packets received by the destination up to that instant.

Several complications arise in the analysis of the RLNC scheme in general networks. First, the possibility of packet replication makes a Markov modeling of the dynamics of the system intractable. This is mainly due to the fact that a node having multiple incoming edges could receive a single packet of information even if it receives multiple distinct packets during an epoch. Such an event can occur when the packets are linearly dependent on one another. Therefore, even if the individual statistics of the arrival process on each of the incoming edges incident on a node are exactly known, it is insufficient to model exactly, since the packets arriving from different arrival processes may be have common information. Therefore, the exact system dynamics must track the joint statistics of the packet processes on various links in the network. Another issue that comes to light is the concept of occupancy of a node. In a line network, the occupancy can be denoted as the number of packets of information that has not yet been conveyed to the unique next-hop neighbor. However, for a node with more than one outgoing edge, it is unclear as to how to define the concept of occupancy. Clearly, the number of packets of information stored by a node that has not been conveyed to a neighbor via an outgoing link is different from that for another neighboring node. Therefore, the number of states for a node increases with the number of outgoing links. In fact, it can be argued that a node with k outgoing edges requires $2^k - 1$ states to accurately represent its local dynamics. Therefore, the jump from the analysis of capacity of line networks to that of general networks is not straightforward.

[12] This however assumes a large Galois field used for computing linear combinations.

In order to keep the system tractable, we make a simplifying assumption regarding the packet arrival processes as seen by different nodes in the network. We assume that the packet arrival processes at all nodes can be modeled by geometric arrivals and that the packet arrival process originating from different nodes are statistically independent. This assumption, part of a set of assumptions referred to as the *Memoryless Packet Process (MPP) assumptions*, allows us to develop an approximate Markov setup that tracks just the mean inter-arrival time, or equivalently, the rate of packet arrival on different links. Such a setup enables a flow-like perspective to analyzing packet traversal through the network. As we shall see in Section 11.4 via simulations, this assumption is not stringent and allows a fairly accurate prediction of the capacity of the network. In what follows, we exploit this assumption to derive an upper and lower estimate on the capacity of a network. Finally, we present an extension of the iterative estimation technique for line network that models the phenomenon of blocking experienced by packets. First, we introduce a Markov modeling for a specific model of nodes in a general network that tracks packets as they traverse it.

11.3.2 A Markov Model for a Node in a General Network

Consider a node $u \in V$ in a network $\overrightarrow{G}(V, \overrightarrow{E})$ with d_i incoming and d_o outgoing edges and a buffer size of m_u as depicted in Figure 11.5. Let the nodes that can send packets to u be denoted by $\mathcal{N}^+(u) \triangleq \{v_i, \ldots, v_{d_i}\}$. Similarly, let the nodes to which u can send packets be denoted by $\mathcal{N}^-(u) \triangleq \{w_i, \ldots, w_{d_o}\}$. Assume that the following MPP assumptions hold in the network regarding the arrival and departure processes.

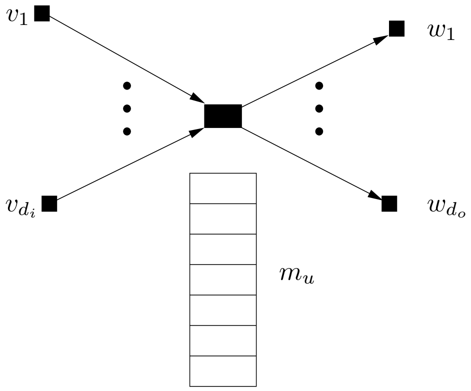

Figure 11.5: A Node in a general wired network.

1. For each $k = 1, \ldots, d_i$, suppose that the packets arrive on (v_k, u) in a memoryless fashion with a rate of λ_k packets/epoch *i.e.,* with the inter-arrival distribution being $\mathbb{G}(\frac{1}{\lambda_k})$. Also, any two packets are distinct and are assumed to possess no common information. Also, the processes on different incoming links are assumed to be statistically independent.

2. At any instant, for every $k = 1, \ldots, d_o$, a packet is sent on (u, w_k) it is successfully received and stored at w_k with a probability ω_k independent of the past and future events on the edge.

Note that this is hypothetical since in any realistic model of a network, the probability that a packet is successfully transmitted and stored at the next hop depends not only on the channel conditions, but also state of the next-hop node. Since the state of the next-hop node has dependence on its past, the probability of successful receipt can also be expected to have a dependence on its past. Since exact modeling of RLNC is complicated, we assume a degraded model for node operation that can simulates RLNC[13]. In fact this mode of node operation can be replaced by any other scheme that fits into the Markovian set-up of the MPP assumptions.

Suppose that the node operates using the following rules, one after another.

1. At each epoch, every node u selects a random ordering of the outgoing edges and transmits the packets it houses one by one. If the packet is successfully received and stored at a neighbor, u deletes the packet from its buffer and transmits the next packet (if any) on the next edge in the selected order. Else, it tries to transmit the same packet on the next outgoing edge. This process is continued until all packets are transmitted or a transmission is attempted on each link.

2. After the transmission attempts are made, the node attempts to accept the arriving packets. If more packets are received than it can store, it selects a random subset of the set of arriving packets whose size equals the amount of space available and stores

[13] Note that the outlined mode assumes the presence of lossless feedback on links. Even under this assumption, it can be proved that the throughput of RLNC is strictly higher than such a scheme.

the selected packets. Appropriate acknowledgement messages are then sent.

Note that under such a mode of operation, at any epoch when two nodes receiving two packets from a particular node, the packets received have no common information content. Equivalently, no replication is performed at any node and it is possible to define a concept of state or occupancy for each node. It can be seen that this concept of occupancy follows a Markov chain behavior and can be studied thus.

At any instant, the number of packets arriving can range from 0 up to d_i and the number of packets departing can range from 0 to d_o. Hence, at each epoch, the state s can change to any other in the set $\{s - d_o, \ldots, s + d_i\} \cap \{0, \ldots, m_u\}$. At any epoch, the probability a_k with which k packets arrive is calculated by

$$A(x) = \sum_{i=0}^{d_i} a_i x^i = \prod_{k=1}^{d_i} (\overline{\lambda}_k + \lambda_k x). \tag{11.27}$$

Similarly, the probability e_k with which k packets depart e_k at any epoch (provided there are sufficient packets in the buffer) is found by

$$E(x) = \sum_{i=0}^{d_o} e_i x^i = \prod_{k=1}^{d_o} (\overline{\omega}_k + \omega_k x). \tag{11.28}$$

The dynamics of the number of packets $n_u(l)$ stored at u at the l^{th} epoch is a Markov chain that is similar to the one depicted in Figure 11.6. Note that since at most two packets can arrive or leave in an epoch, the state of the node can potentially have a change of at most two packets. For all input parameters, the Markov chain can be shown to be aperiodic, irreducible and ergodic. Therefore, it possesses a unique steady-state distribution. Letting $\Lambda = (\lambda_1, \ldots, \lambda_{d_i})$ to denote the vector of arrival rates and $\Omega = (\omega_1, \ldots, \omega_{d_i})$ to denote

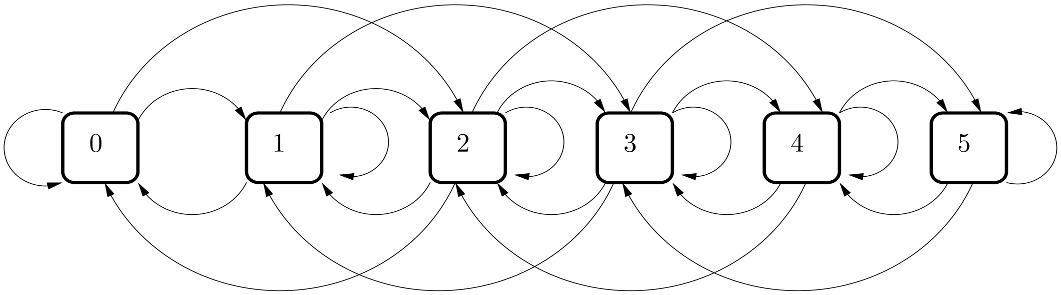

Figure 11.6: The dynamics of a node u with $m_u = 5$ and $d_i = d_o = 2$.

the vector of departure rates, the unique steady-state distribution $\vartheta(\cdot, \Lambda, \Omega, m_u)$ for the chain can be computed using a pair of probability transition matrices T_E and T_A[14] that correspond to the transitions between states that are effected by the departure and arrival of packets, respectively. Note that ϑ is the steady-state distribution after the arriving packets are processed. These transition matrices are defined as follows.

$$T_E = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \sum_{k=1}^{d_i} e_k & e_0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \sum_{k=2}^{d_i} e_k & e_1 & e_0 & 0 & \cdots & 0 & 0 & 0 \\ \sum_{k=3}^{d_i} e_k & e_2 & e_1 & e_0 & \cdots & 0 & 0 & 0 \\ & & & \vdots & & & & \\ \sum_{k=m_u}^{d_i} e_k & e_{m_u-1} & e_{m_u-2} & e_{m_u-3} & \cdots & e_2 & e_1 & e_0 \end{bmatrix}. \tag{11.29}$$

$$T_A = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots & a_{m_u-2} & a_{m_u-1} & \sum_{k=m_u}^{d_i} a_k \\ 0 & a_0 & a_1 & a_2 & \cdots & a_{m_u-3} & a_{m_u-2} & \sum_{k=m_u-1}^{d_i} a_k \\ 0 & 0 & a_0 & a_1 & \cdots & a_{m_u-4} & a_{m_u-3} & \sum_{k=m_u-2}^{d_i} a_k \\ & & & \vdots & & & & \\ 0 & 0 & 0 & 0 & \cdots & 0 & a_0 & \sum_{k=1}^{d_i} a_k \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix}. \tag{11.30}$$

Note that the i, j^{th} entry in T_E corresponds to the transition of the occupancy from $i-1$ to $j-1$ with the departure of $i-j$ packets. Similarly, the i, j^{th} entry in T_A corresponds to the transition from $i-1$ to $j-1$ with the arrival of $i-j$ packets. The actual transition matrix for the Markov chain is then seen to be $T_E T_A$. The steady state ϑ of the occupancy at each epoch just after the arriving packets are accepted can then be found by the eigenvector relation.

$$\vartheta T_E T_A = \vartheta. \tag{11.31}$$

[14] For notational consistency, we can extend $e_k = 0$ for $k > d_o$ and $a_k = 0$ for $k > d_i$. Also, for notational convenience, we use $\vartheta(\cdot)$ as a short-hand $\vartheta(\cdot, \Lambda, \Omega, m_u)$.

Similarly the steady-state distribution $\vartheta^\dagger$ just after the packets have been sent but before arriving packets are accommodated is given by

$$\vartheta^\dagger T_A T_E = \vartheta^\dagger. \tag{11.32}$$

However, these two steady-state distributions are related by $\vartheta^\dagger = \vartheta T_E$ and $\vartheta = \vartheta^\dagger T_A$. The maximum rate of information that the node can pump through all of the d_o outgoing edges $J(\{u\} \times \mathcal{N}^-(u), \Lambda, \Omega, m_u)$ can then be seen to be

$$J(\{u\} \times \mathcal{N}^-(u), \Lambda, \Omega, m_u) = \sum_{k=0}^{m_u} k\vartheta(k) - \sum_{k=0}^{m_u} k\vartheta^\dagger(k) = \sum_{k=0}^{m_u} k[\vartheta(k) - \vartheta^\dagger(k)]. \tag{11.33}$$

To evaluate the rate of information is sent over the link (u, w_i), one must investigate the rule for packet departure. If at an epoch, more packets are stored than the number of links that allow successful transmission, then each link conveys a packet of information to its neighbors. However, if the occupancy n_u at an epoch l is smaller than the number s of outgoing links that allow for transmission, statistically, each link can be statistically assumed to equally receive $\frac{n_u}{s}$ packets – a consequence of the randomly selection of ordering for outgoing links. Mathematically, the time average of the information rate on the edge (u, w_i) can be seen to be

$$I(\{(u, w_i)\}, \Lambda, \Omega, m_u) = \sum_{\substack{S \subset \{0,\ldots,d_o\} \\ i \in S}} \Big(\prod_{k \in S} \omega_k\Big)\Big(\prod_{k' \in S^c} \overline{\omega}_{k'}\Big)\Big(\sum_{j \geq |S|} \vartheta(j) + \sum_{j < |S|} \frac{j}{|S|}\vartheta(j)\Big). \tag{11.34}$$

In a similar argument, we notice that some of the arriving packets get randomly blocked if all the arriving packets cannot be stored. We can evaluate the probability with which a packet arriving on the edge (v_i, u) is blocked from

$$p_b(\{(v_i, u)\}; \Lambda, \Omega, m_u) = \sum_{\substack{S \subset \{0,\ldots,d_i\} \\ i \in S}} \Big(\prod_{k \in S \setminus \{i\}} \lambda_k\Big)\Big(\prod_{k' \in S^c} \overline{\lambda}_{k'}\Big)\Big(\sum_{m_u - j < |S|} \frac{|S| - m_u + j}{|S|}\vartheta^\dagger(j)\Big). \tag{11.35}$$

Note that the above two equations will have to be modified for any other Markovian mode of operation that is used to model RLNC.

11.3.3 Bounds on the Capacity of General Networks

In this section, we develop a simple bounds on the unicast capacity of a directed acyclic network. The upper bound studies the network in a node-by-node fashion to develop the maximum amount of information that can be relayed on any subset of outgoing edge of a node. We then combine this information in conjunction with the max-flow min-cut theorem [15] to develop our result. The lower bound uses a natural extension of the lower bound for line networks with the MPP assumptions.

In order to exploit the max-flow min-cut theorem, we have to suitably define the capacity of a cut in such networks. To associate the traditional meaning for the capacity of the cut to be based on the channel erasures alone would ignore the effect of finite buffers. However, we modify (11.33) to derive an upper bound on the actual capacity of a cut. The fundamental idea is to notice that a node u can receive a packet via a link only if the link does not erase this. Therefore, assuming that on each incoming edge $\overrightarrow{e} = (v, u)$, the arrival process is memoryless with a rate of $\overline{\varepsilon}_{vu}$, maximizes the inforamtion rate supported by the link. Additionally, assuming that the joint entropy rate on the packet processes on all the incoming links is the sum of the individual entropy rates of the packet process on each link, we maximize the joint rate at which information is pumped onto the node u. Similarly, assuming that all the neighbors on the outgoing links of u have infinite buffers, decreases the time packets spend in the buffer of u and hence maximizes the rate at which information is conveyed via its outgoing edges. This assumption is equivalent to assuming that the packets depart in a memoryless fashion at the rate of $\overline{\varepsilon}_{uw_i}$ on the link (u, w_i). Thus, the rate of information delivered via all the outgoing edges computed with these three assumptions on the arrival and departure process will be greater than the joint rate at which any realistic scheme conveys information. Note also that these assumptions effect an idealized model of the network around the node u, and the results of Section 11.3.2 are applicable.

Thus, an upper bound on the joint rate of information flow on any subset of outgoing edges $S \subset \mathcal{N}^{-}(u)$ under any scheme employed is bounded from above by that computed from (11.34) using the Markov chain of Section 11.3.2 for the node u with buffer size m_u, $|\mathcal{N}^{+}(u)|$ incoming edges pumping packets with the arrival rate vector $\Lambda(u, S) = (\overline{\varepsilon}_{vu})_{v \in \mathcal{N}^{+}(u)}$, and $|S|$

outgoing edges with the departure rate vector $\Omega(u,S) = (\overline{\varepsilon}_{uw})_{w\in S}$. This bound computed using the chain is seen to be

$$\mathcal{L}(u,S) = \sum_{k=1}^{m_u} k\Big(\vartheta(k,\Lambda(u,S),\Omega(u,S),m_u) - \vartheta^{\dagger}(k,\Lambda(u,S),\Omega(u,S),m_u)\Big). \tag{11.36}$$

Once this maximum rate is computed for all nodes u and all subsets of outgoing edges, we can suitably define the capacity of any subset of edges and use the following result to obtain an upper bound for the capacity $\mathcal{C}(s,d,\overrightarrow{G})$ between a pair of nodes s and d in $\overrightarrow{G}$.

Theorem 11.3.1. *Consider a pair of nodes s,d in a wired acyclic directed network $\overrightarrow{G}$. Let the capacity of any subset of edges be given by the function $\kappa : 2^{\overrightarrow{E}} \longrightarrow \mathbb{R}^{+}$ with the definition. $S \mapsto \sum_{u\in V} \mathcal{L}(u,(u\times\mathcal{N}^{-}(u))\cap S)$. Then,*

$$\mathcal{C}(s,d,\overrightarrow{G}) \leq \min_{F:s-d\ cut} \kappa(F). \tag{11.37}$$

Proof. The proof follows from the max-flow min-cut theorem and from the fact that the the maximum information that traverses through an $s-d$ cut is bounded above by $\kappa(S)$. The latter can be seen by partitioning all the edges of the cut based on the starting node and applying (11.36) for each node with the appropriate set of outgoing edges. ❑

We would like to add here that it is possible to improve the upper bound on the information flow-rate of (11.36) by investigating the assumption that the packets on different link have no common information, which is not necessarily the case, in general. We leave it as a future exercise to investigate this direction. Lastly, we conclude our discussion on the upper bound by noting that the above theorem can be used in conjunction with any improvement of (11.36).

To develop a lower bound on the capacity of general networks, we can use any realistic mode of operation of nodes in conjunction with the MPP assumptions. In specific, we can extend the mode of operation that we use in line networks as follows.

1. At each epoch, every node selects a random ordering of the outgoing edges and transmits the packets it houses one by one. If it receives an acknowledgement of successful receipt, then it deletes the packet from its buffers and transmits the next packet (if

any) on the next edge. Else, it tries to transmit the same packet on the next outgoing edge. This process is continued until all packets are transmitted or a transmission is attempted on each link.

2. Once the transmissions are attempted, the node processes the arriving packets. Successful receipt is sent to the nodes from which an arriving packet is received. However, if the number of packets arriving at any epoch is greater than the space available, a random subset of the set of arriving packets whose size equals to the available space is selected and stored.

Note that under such a mode of operation, a successful receipt acknowledgement is sent by the node that received the packet even if the packet is dropped. The main advantage under this mode of node operation is that the departure process from any node u on any outgoing link (u, v) is memoryless and dependent only on the link alone and not the state of the neighboring node v. Moreover, the departure can occur on each link at any epoch with a probability of ε_{uv}. Additionally, by the MPP assumptions, both the departure and arrival rates in the vicinity of each node is memoryless and independent and we can use the results of Section 11.3.2. However, in order to identify the lower bound on actual departure rate of packets on each outgoing link of a node, we must know the arrival rates on each of its incoming edge. This is achieved by the assumption of absence of any directed cycles in the network.

The absence of directed cycles allows us to define a natural partial ordering '$\preceq$' on the set of nodes of the network in the following manner. We let $u \preceq v$ if there is a directed path from u to v in $\overrightarrow{G}$. Without loss of generality, we can assume that for all nodes u in the network lie on some path from the source s to the destination d, *i.e.*, $s \preceq u$ and $u \preceq d$. We can define a total ordering of nodes '$\rhd$' such that $u \rhd v$ if either '$u \preceq v$' or 'both $u \npreceq v$ and $u \npreceq v$'. In a line network, $\rhd$ is the ordering based on the hop-distance of the node from the source s. Therefore, identifying the departure rates from various nodes in the order determined by $\rhd$ provides a natural extension of the lower bound of the line network. The rate on all the links can be identified in the following manner.

1. Let $s \rhd v_1 \rhd \cdots \rhd v_n \rhd d$ be any total ordering of the nodes as described above.

2. Set $i = 1$. Define for each edge $\overrightarrow{e} = (s, v) \in \overrightarrow{E}$, $\rho_{uv} = \overline{\varepsilon}_{sv}$.

3. If $i < n + 1$, compute the steady-state distribution $\vartheta_i(\cdot)$ for the i^{th} node v_i by setting its the arrival rate vector as $\Lambda_i = (\rho_{v_i w})_{w \in \mathcal{N}^+(v_i)}$, the departure rate vector as $\Omega_i = (\overline{\varepsilon}_{v_i w})_{w \in \mathcal{N}^-(v_i)}$, and buffer of m_{v_i} packets. Else, stop.

4. For every $w \in \mathcal{N}^-(v_i)$, compute $\rho_{v_i w}$ from (11.34) using ϑ_i and increment i by 1.

Using the computed rates, one can construct a lower bound the capacity of the network as follows.

Theorem 11.3.2. *Assuming that the MPP assumptions hold, we have*

$$\mathcal{C}(s, d, \overrightarrow{G}) \geq \sum_{\overrightarrow{e} \in \mathcal{N}^+(d) \times \{d\}} \rho_{\overrightarrow{e}}. \tag{11.38}$$

Proof. The proof follows from a simple extension of the argument for line networks using a total ordering of the nodes described above. ❑

11.3.4 Capacity Estimation in General Networks

In this section, we discuss an iterative capacity estimation technique that exploits the MPP assumptions in addition to considering the blocking phenomenon that packets experience in the network. Notice that to map the realistic network scenario to memoryless arrival and departure process on various links, the upper and lower bound used the MPP assumptions in conjunction with either the assumption that neighboring nodes have infinite buffer space or the assumption that nodes send confirmation even if they drop the packet. In effect, neither of the two bounds take into consideration the realistic phenomenon of blocking of packets. However, as previously discussed, considering blocking will introduce dependence of the packet process over each edge on its past. Therefore, to use the results of Section 11.3.2, we have to make certain simplifying assumptions on the blocking phenomenon. In this work, we model the blocking on every edge $\overrightarrow{e} = (u, v)$ of the network as follows.

- Every packet that arrives at v successfully (without getting erased) is blocked in a memoryless fashion with probability q_{uv}. Also, at any epoch, the blocking of packets on any subset of incoming edges of v is assumed to be independent of one another.

Under the above assumption, the blocking process and hence the departure process on every link of the network is modeled as a memoryless process. Since each packet arriving on an edge $\overrightarrow{e} = (u, v)$ is blocked with a probability of q_{uv}, a packet arriving on $\overrightarrow{e}$ is accepted only if both the channel allows the packet and the node accepts it. Therefore, the effective departure rate on the edge (u, v) is seen to be $\overline{\varepsilon}_{uv}\overline{q}_{uv}$. Assuming that the node performs operates in the mode described in Section 11.3.2, we can use (11.34) and (11.35) to identify both the rate of information flow and the blocking probabilities on every edge of the network. Thus, the problem reduces to finding a solution $(\varrho_{uv}, q_{uv})_{(u,v)\in\overrightarrow{E}}$ that satisfy following system of non-linear equations for each $(u, v) \in \overrightarrow{E}$.

$$\varrho_{uv} = \begin{cases} \overline{\varepsilon}_{uv} & u = s \\ \frac{I(\{(u,v)\},(\varrho_{wu})_{w\in\mathcal{N}^+(u)},(\overline{\varepsilon}_{uu'}\overline{q}_{uu'})_{u'\in\mathcal{N}^-(u)},m_u)}{\overline{q}_{uv}} & u \neq s \end{cases}, \tag{11.39}$$

$$q_{uv} = \begin{cases} p_b(\{(u,v)\};(\varrho_{wv})_{w\in\mathcal{N}^+(v)},(\overline{\varepsilon}_{vv'}\overline{q}_{vv'})_{v'\in\mathcal{N}^-(v)},m_v) & v \neq d \\ 0 & v = d \end{cases}. \tag{11.40}$$

Note that in the above equations ϱ_{uv} represents the fraction of time at which packets will be delivered to v. However, the actual rate of information flow on the edge is computed by $\rho_{uv} = \overline{q}_{uv}\varrho_{uv}$. Since the above set of equations are an approximation to the actual dynamics, it is not clear as to whether there even exists a solution to the above system. However, the argument for the proof of Theorem 11.2.4 can be modified with the use of any of the aforementioned total ordering of the nodes to show the existence and uniqueness of the solution. Finally, the solution to the system of equations can be found by identifying the limit of the sequence defined by the following iterative procedure. In practice, a small number of iterations L suffice to converge to the solution within reasonable accuracy.

1. Set $i = 1$ and for each edge $(u, v) \in \overrightarrow{E}$, set $q_{uv} = 0$ and $\varrho_{uv}^{(1)} = \begin{cases} 0 & u \neq s \\ \overline{\varepsilon}_{uv} & u = s \end{cases}$.

2. Compute $\varrho_{uv}^{(i+1)}, q_{uv}^{(i+1)}$ by using $\varrho_{uv}^{(i)}, q_{uv}^{(i)}$ on the right-hand side of (11.39) and (11.40) and increment i by 1.

3. If $i < L + 1$, perform step 2.

Since the mode of operation is such that information is not replicated at any node, the estimate of the total information that arrives at the destination is the sum total of the information rate arriving on each of its incoming edges.

$$\hat{\mathcal{C}}(s, d, \overrightarrow{G}) = \sum_{v \in \mathcal{N}^+(d)} \varrho_{vd}^*(1 - q_{vd}^*) = \sum_{v \in \mathcal{N}^+(d)} \varrho_{vd}^*, \tag{11.41}$$

where we let $(\varrho_{uv}^*, q_{uv}^*)$ to be either the component-wise limit of the sequence $\{\varrho_{uv}^{(i)}, q_{uv}^{(i)}\}_{i \in \mathbb{N}}$ when $L = \infty$, or $(\varrho_{uv}^{(L)}, q_{uv}^{(L)})$ when $L < \infty$. Additionally, by the conservation of information flow, the above estimate can be obtained by computing the rate of flow of information through any cut H using the following.

$$\hat{\mathcal{C}}(s, d, \overrightarrow{G}) = \sum_{uv \in H} \varrho_{uv}^*(1 - q_{uv}^*). \tag{11.42}$$

11.4 Results of Simulation

In this section we present the results of simulation, where we compare our analytic results to actual simulations of line and general wired acyclic networks. We first present our results of line networks and then go on to present that of general networks later. Finally, we conclude this section with a brief discussion on the possibility of future improvements to our analysis.

11.4.1 Results on Line Networks

A line network is uniquely defined if one provides the number of hops, the set of erasures on each link and the buffer sizes at each of the intermediate node. To understand the variation of our bounds (given by (11.14) and (11.16)) and estimate (given by (11.26)) in Section 11.2, we vary one of these three parameters keeping the remaining two fixed. In each of the figures, the simulation of the actual capacity is presented in addition to our results. Figure 11.7 presents the variation of the capacity with the hop length for a network with each intermediate node having a buffer size of five packets. Moreover, two choices of

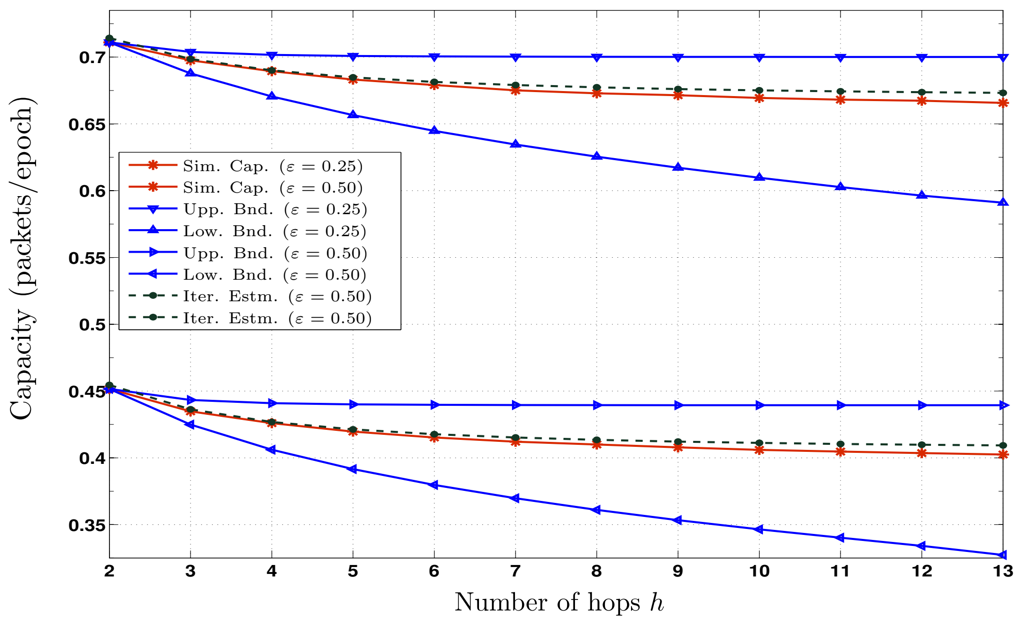

Figure 11.7: Capacity of a line network with $m = 5$ as a function of the number of hops h.

erasure for the links are depicted. The simulations are performed when the probability of erasure on every link is set to either 0.25 or 0.5. While it is noticed that the bounds and the estimate capture the variation of the actual capacity of the network, the estimate is the most accurate and it predicts the variation with about 1.5% of error. It was noticed that the estimate is always an over-estimate of the actual capacity of the network.

In order to study the effect of buffer size, we simulated a line network of eight hops and intermediate nodes with each link having the same erasures as in the previous setting. Figure 11.8 presents the variation of our results and the actual capacity as the buffer size of the intermediate node is varied. It can be seen that as the buffer size is increased, all curves approach the ideal min-cut capacity of $1 - \varepsilon$. Also, the accuracy of the bounds improve with the buffer size.

Finally, the effect of the channel conditions on the capacity of a line network is presented in Figure 11.9. Here, a network of eight hops and intermediate nodes with a buffer size of five packets was simulated. It is noticed that as the probability of erasure increases, the drop in the capacity due to finite buffer becomes more pronounced. For example, for the

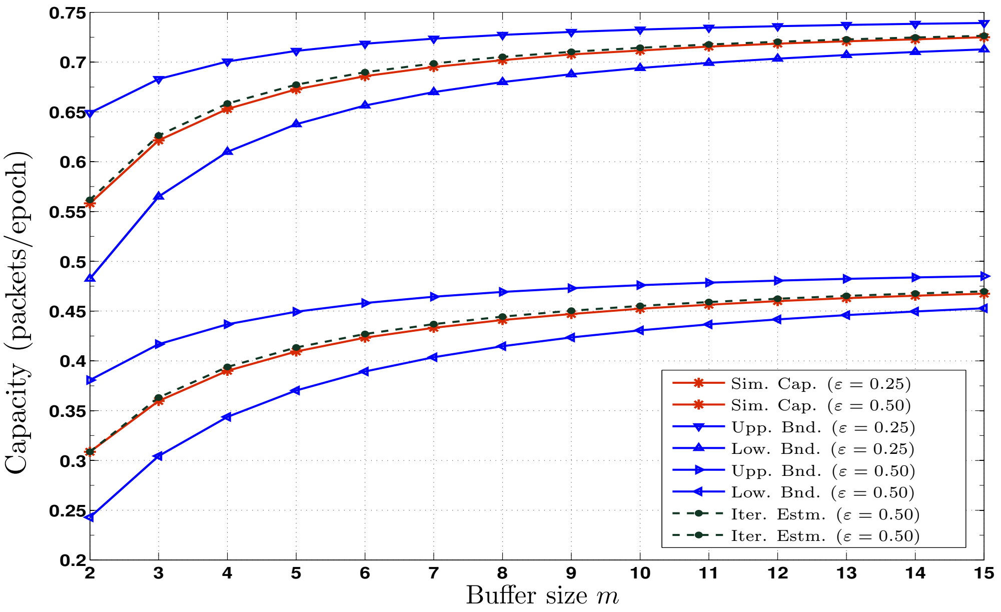

Figure 11.8: Capacity of a line network with $h = 8$ as a function of the buffer size m.

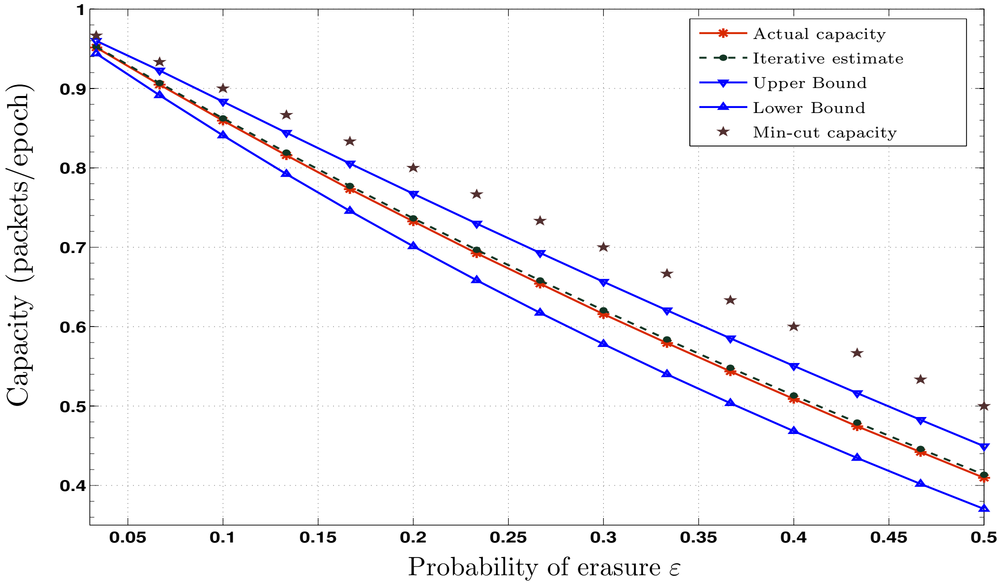

Figure 11.9: Capacity of a line network with $h = 8$, $m = 5$ as a function of the erasure probability ε.

simulation setting of Figure 11.9, the drop in capacity varied from 1.5% at $\varepsilon = 0.033$ to 18.1% at $\varepsilon = 0.5$ in a near-linear fashion.

11.4.2 Results on General Networks

We consider three networks to describe the results of simulation and inferences. Figures 11.10, 11.11 and 11.12 present the three networks that are referred henceforth as Network 1, Network 2 and Network 3, respectively. Since no node in Network 1 has more than one outgoing edge, splitting of information streams does not occur. However, both splitting and joining of information streams can potentially occur in Network 2 and Network 3.

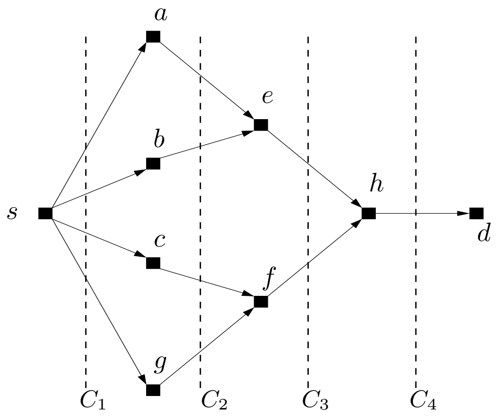

Figure 11.10: An acyclic directed network chosen for simulation.

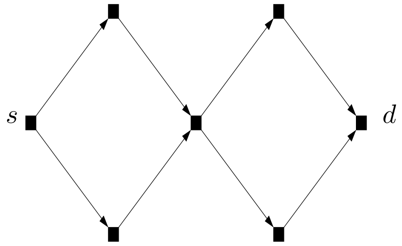

Figure 11.11: An acyclic directed network chosen for simulation.

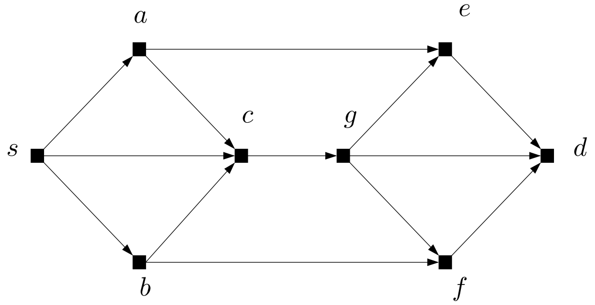

Figure 11.12: An acyclic directed network chosen for simulation.

In Network 1, the erasure probabilities for the edges chosen for simulation are as follows. All edges in cuts C_1 and C_2 in Figure 11.10 have $\varepsilon = 0.75$, whereas those of C_3 and C_4 were set to $\varepsilon = 0.5$ and $\varepsilon = 0.1$, respectively. Under such a choice, the min-cut capacity that is achieved in the presence of infinite buffer size is seen to be 0.9 packets/epoch. In Network 2 and Network 3, all the edges have the same probability of erasure of $\varepsilon = 0.5$ and $\varepsilon = 0.25$, respectively. Therefore, the min-cut capacity of these two networks can be seen to be 1 packets/epoch and 2.25 packets/epoch, respectively. In all the networks, every intermediate nodes is assumed to have the same buffer size. In order to simulate the exact capacity of the networks, the steady-state rate at which innovative packets arrive at the destination was computed when random linear network coding over the field $\mathbb{F}_{50021}$ is employed at each intermediate node.

Table 11.1: Variation of the capacity with the size of the buffer for Network 1.

Buffer Size	Actual capacity	Estimated capacity	Upper bound	Lower Bound
1	0.4685	0.4308	0.5474	0.3604
3	0.7451	0.7311	0.7920	0.6890
5	0.8313	0.8213	0.8644	0.7903
8	0.8803	0.8747	0.8986	0.8519
10	0.8902	0.8894	0.9000	0.8714
15	0.9000	0.8990	0.9000	0.8925

Table 11.1 presents the variation of actual capacity and our estimate and bounds from Section 11.3 (in packets/epoch). It is noted that the estimate predicts the actual capacity within an error of 8%. Also, the actual capacity is always noticed to be higher than that of the estimated capacity. This can be attributed to the fact that the RLNC scheme uses all paths to deliver a packet and when the one of the copy reaches the destination successfully, all other copies are effectively deleted. On the other hand, in the estimate, each packet is sent only on one link and it has to rely on this link to deliver the packet.

Figure 11.13 presents the variation of the actual capacity and our results with the size of the buffers at intermediate nodes of Network 2. The trends that are noticed in Network 1 also hold here. Notice that this network has an *articulation point* [15]. This intermediate node is where maximum congestion is experienced. As a result of this congestion, a large

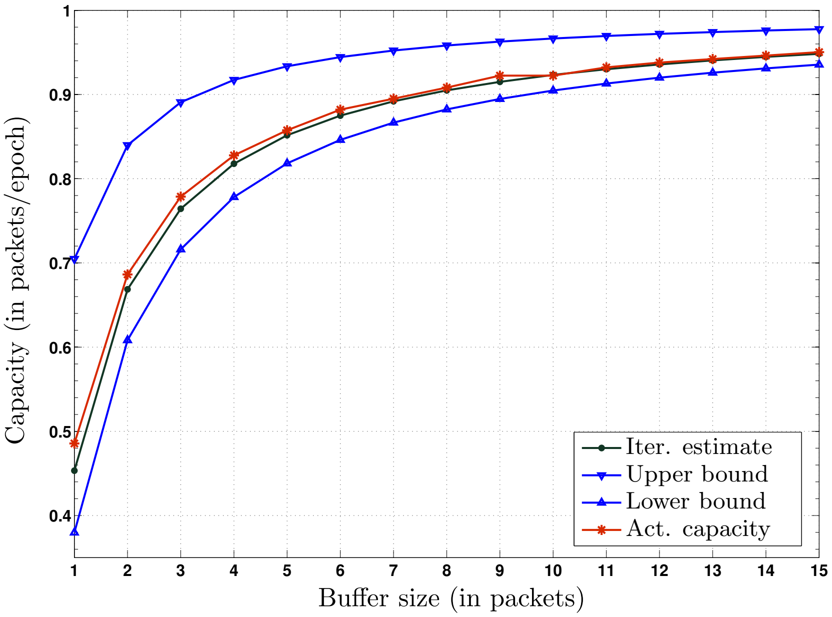

Figure 11.13: Variation of the capacity with the size of the buffer for Network 2.

buffer is required at this node for the actual capacity to approach very close to the min-cut capacity. For example, in this network, we need a buffer size of about 80 packets at intermediate nodes for the actual capacity of the network to approach within 1% of the min-cut capacity *i.e.,* to have a capacity of 0.99 packets/epoch. Notice that in both Network 1 and Network 2, as the buffer size is increased, the estimate and the bounds approach the min-cut capacity of the network. This is because the operation of nodes assumed in Section 11.3.2 is optimal as the buffers become large. However, as we shall note, this is not the case, in general.

Consider the variation of the simulated capacity and our results for Network 3 with the size of the buffer presented in Figure 11.14. In this network, it is seen that no matter how large the estimate, the estimate does not approach the min-cut capacity of 2.25 packets/epoch. As the buffer size is made large, the upper bound, the estimate and the lower bound approach 2.25, 1.815 and 1.815 packets/epoch, respectively. This is attributable to the fact that the mode of operation, wherein nodes send packets over each outgoing link

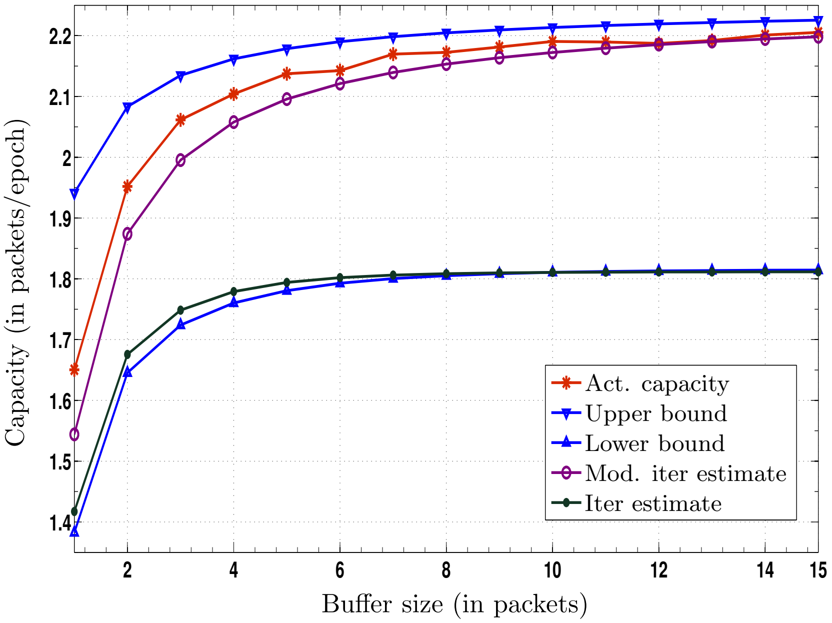

Figure 11.14: Variation of the capacity with the size of the buffer for Network 3.

when possible, is not asymptotically[15] optimal for Network 3. The asymptotic optimal mode of operation for the nodes is to ignore the edges $(a, c), (b, c), (g, e)$ and (g, f) and transmit packets on three node-disjoint paths in the graph. If one were to use this information from the asymptotically optimal flow allocation and modify the iterative estimate based on this allocation, we would obtain the modified iterative estimate that is also presented in the figure. Note that this estimate approximates the actual performance of the RLNC scheme very well.

Finally, we present the variation of the actual capacity and our results with the channel parameter for both Network 2 and Network 3. For this simulation, we set the buffer size at five packets for all intermediate nodes. It is noticed in both networks that the estimate becomes tighter when the channel erasure probability is close to either zero or one. In Network 1, the estimate predicts accurately (within an error of 1%) over the whole range of simulated channel conditions.

In Network 3, it is noticed that for erasure probabilities sufficiently away from both

[15] As the sizes of the buffers approach infinity.

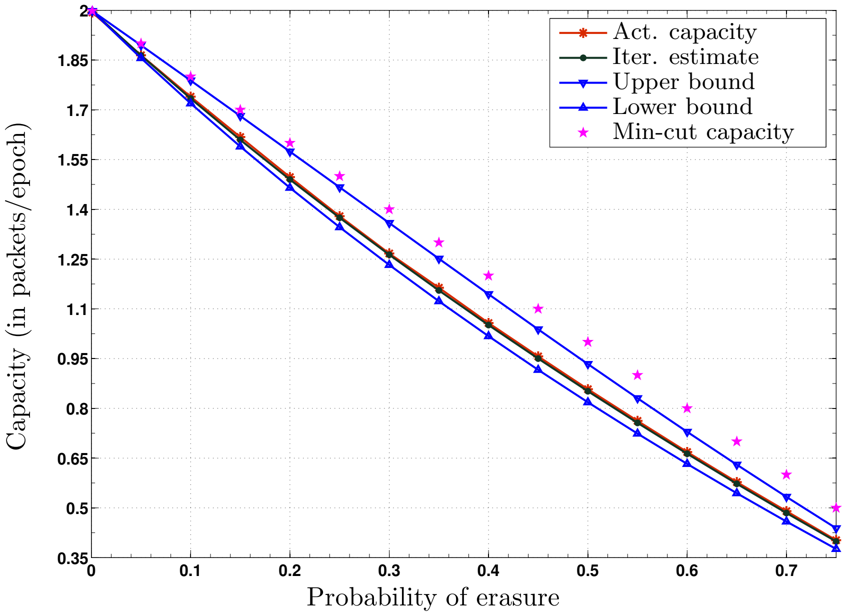

Figure 11.15: Variation of the capacity with erasure probability for Network 2.

zero and unity, there is significant error in the iterative estimate. The error becomes more drastic as the buffer size is increased. The difference between the estimate and the actual capacity for large buffer sizes attains a maximum around a channel erasure probability of $\varepsilon = 0.45$. However, this difference in prediction for large buffer sizes is minimized by the modified iterative estimate that exploits the packet-routing information from the optimal scheme for the infinite buffer scenario.

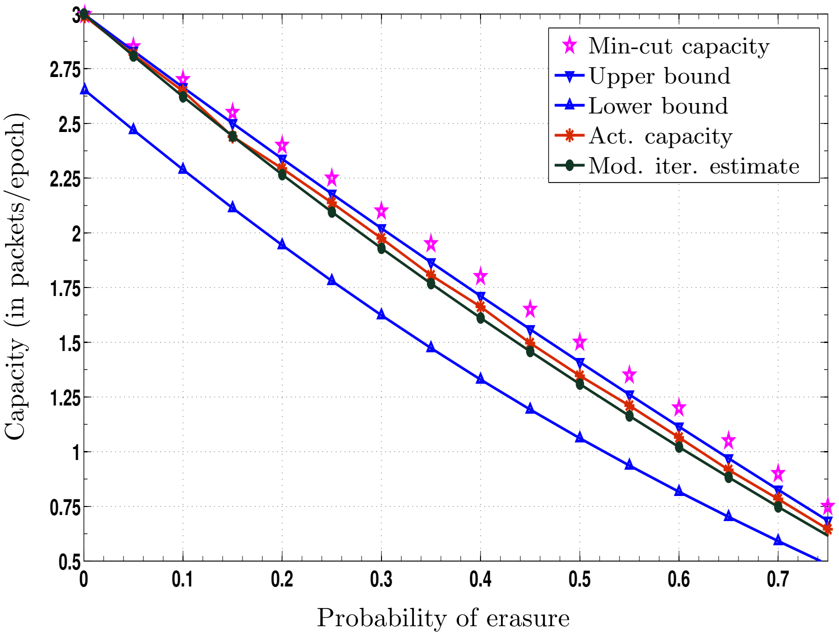

Figure 11.16: Variation of the capacity with erasure probability for Network 3.

CHAPTER XII

CONCLUSION OF THE THESIS

In this dissertation, we investigated several aspects of modern codes such as low-density parity-check (LDPC) codes and rateless codes. The dissertation spans a broad range of issues ranging from the theoretical computation of information-theoretic capacity of networks and the design of improved decoders for LDPC codes to the practical design of data dissemination schemes. In the following, we summarize the contribution of our thesis.

Part I of the thesis studies the theoretical aspects of the design of improved algorithms for decoding and puncturing of finite-length LDPC codes. Although considerable work is available in the literature regarding the design of good decoders for LDPC codes over the binary erasure channel that offer near-optimal performance for very long codes, the design of good decoders for finite-length LDPC codes has been fairly limited. Our work focuses on the class of bit-guessing algorithms. A large portion of the literature on improved decoders that employ bit-guessing are qualitative. However, our approach presented a complete matrix characterization of the structure of stopping sets that are decoded when a fixed number of bits are guessed. Using the characterization, we estimated the number of bits that need to be guessed to complete the decoding of a stopping set. We then introduced two interesting qualitative observations on the distribution of stopping sets that are then exploited in the design of the contractive message-passing (CMP) decoder over the BEC. The proposed decoder offers an improvement of several orders of magnitude over other improved decoders of relevance while maintaining an acceptable computational complexity. Moreover, the number of bits guessed in our scheme is fewer than most other similar schemes.

The second portion of Part I concerns the design of practical rate-compatible schemes for LDPC codes. In this work, we focussed on the problem of constructing a family of punctured codes from a given mother code. We first noticed that by spreading the punctured codes uniformly across the Tanner graph, we can expect minimal degradation in the performance of the code during puncturing. We employed this intuitive criterion to synthesize a novel and simple punctured family of codes. Our puncturing scheme is then extended using an additional layer of puncturing to improve the range of rate compatibility. We noticed that our scheme outperforms other puncturing schemes that have the same approach to puncturing. It was noticed that the proposed scheme results in punctured codes that are, at their respective rates, as good as the mother code is at its rate.

In the last portion of Part I, we address the challenging problem of evaluating the degradation in bit-error rate of the code due to random puncturing. Most approaches to this problem would be aimed at the distribution of stopping sets of the mother code. However, we proposed the unique approach of using the performance of the mother code to estimate that of the punctured codes at different channel parameters. We noticed that our bounds provided a good estimate of the performance of punctured codes over a wide range of channel parameters for all ensembles that were simulated.

part II of the dissertation deals with the design of efficient communication strategies for data dissemination in various networks. We explored three issues. First, we investigated the design of an energy-efficient and reliable broadcasting scheme in multihop wireless networks. We proposed the fractional transmission scheme (FTS), an energy-efficient and reliable broadcasting scheme for networks wherein nodes are aware of their neighborhoods. Our scheme exploits the information available at nodes to decide the minimum fraction that each node must transmit so that all its neighbor can potentially receive all the packets that are broadcasted. Further, rateless codes are employed at each node in the network for a two-fold purpose – to improve reliability in the presence of channel losses, and to guarantee the innovativeness of packets arriving from different nodes in the network. Via simulations, we verified that our scheme outperforms several popular schemes such as multipoint relaying and dominant pruning. In fact, the performance of our scheme is seen to be close to that offered by network coding in random deployment network models.

The second problem studies path enumeration as a means to understand the reliability of probabilistic broadcast (PBcast) in wireless grid networks. The challenging problem of the study of finite grid networks is addressed using combinatorial, algebraic and iterative techniques. Our work reveals that reasonably close estimates can be provided for understanding the variation of reliability of the scheme with the probability of forwarding in spite of the simplifying assumptions that are made during the analysis.

The last problem studies the application of modern of codes in delay-tolerant networks (DTNs). We consider the design of an efficient unicast scheme in DTNs. In such networks, the lack of end-to-end paths and the constantly evolving topology of the network make traditional routing-based approaches to data delivery infeasible. Our pioneering work introduced the rateless code-based (RCb) scheme that employs coding at source to ensure both higher reliability and energy-efficiency. Through simulations, it was verified that our scheme outperformed all realistic coded and uncoded schemes that were simulated.

The final part of the thesis studied two interesting areas of application of Markov chains in the estimation of unicast capacity of finite-buffer networks. The first problem that we investigated is that of the design of a Markov-chain based approach to compute the throughput capacity of a network of nodes performing random walks on grid paths. Our work in this area is one of the first works to consider the finiteness of buffers. Our approach yielded an accurate prediction of the dynamics and throughput of the two-hop scheme when the source and destination are immobile. The approach also incorporated realistic effects of contention in the network into the model. However, when the source and destination nodes are assumed to be mobile, we noticed that the exact chain grows too huge to support exact analysis. In this case, a novel approximate method that greatly reduces the complexity was introduced to compute the throughput. Even for this setting, we noticed that our predictions were more accurate than those made using standard approaches such as the Poisson approximation.

The final problem investigated in this dissertation is the study of the information-theoretic capacity of finite-buffer wired networks. As a first step, the study of line network was performed. We derived an exact Markovian setup for the computation of capacity. We noticed that the Markov chain grows to unmanageable sizes even for moderate line networks. Bounds on the capacity of these networks were then derived. A novel iterative technique considering the phenomenon of packet blocking was then presented. This technique was noticed to provide an accurate prediction of the capacity of a line network. We then extend our results using the traditional approach of viewing information as flows through a network. It was understood that the study of the optimal scheme does not fit directly in a Markovian setting due the replication of packets at nodes. Therefore, only schemes that involved no replication were considered. Again, bounds and estimates on the capacity of these networks were introduced. Though the estimates for general networks were not as accurate as in the case of line networks, the results were very encouraging given the extremely challenging aspect of finite-buffer networks.

APPENDIX A

PROOFS OF RESULTS IN PART I

A.1 Proof of Theorem 4.2.1

Let $e_1, e_2, \dots, e_{|S|}$ represent the values of the erased bits. Suppose that H_S satisfies the similarity condition. Then $\exists\, \pi \in \mathfrak{S}_{|S|}$, $\sigma \in \mathfrak{S}_{n-nR}$ such that

$$\left(\begin{array}{c|c|c|c|c|c|c|c}
\multicolumn{7}{c|}{X_t} & \\ \cline{1-7}
I_{s_t} & \multicolumn{6}{c|}{X_{t-1}} & \\ \cline{1-7}
\mathbf{0} & I_{s_{t-1}} & \multicolumn{5}{c|}{X_{t-2}} & \\ \cline{1-7}
\mathbf{0} & \mathbf{0} & I_{s_{t-2}} & \multicolumn{4}{c|}{X_{t-3}} & X_0 \\ \cline{1-7}
\multicolumn{7}{c|}{\vdots} & \\ \cline{1-7}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & I_{s_2} & X_1 & \\ \cline{1-7}
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} & I_{s_1} &
\end{array}\right)
\begin{pmatrix} e_{\pi(1)} \\ e_{\pi(2)} \\ \vdots \\ e_{\pi(|S|)} \end{pmatrix} = \mathbf{0}. \tag{A.1}$$

Suppose that we correctly guess the value of the g bits $e_{\pi(|S|-g+1)}, \dots, e_{\pi(|S|)}$ that correspond to the columns of X_0. Then, at the end of the first iteration, we will recover bits indexed by the columns of I_{s_1}, namely $e_{\pi(|S|-g-s_1+1)}, \dots, e_{\pi(|S|-g)}$. In the next iteration, we will identify bits $e_{\pi(|S|-g-s_1-s_2+1)}, \dots, e_{\pi(|S|-s_2-g)}$. Inductively, at the i^{th} iteration we will identify the bits $e_{\pi(|S|-g-\sum_{j=1}^{i} s_j+1)}, \dots, e_{\pi(|S|-g-\sum_{j=1}^{i-1} s_j)}$, and all the bits will be identified in exactly t iterations over and above the iterations of the standard MP decoder.

Now suppose that the stopping set is g-solvable. Then there exists a free set of size g. Let t denote the number of iterations taken to completely dissolve S. Denote for each $i \in \{1, \dots, t\}$, S_i to be the set of bit nodes that are freed at the completion of the i^{th} iteration. Also, denote S_0 to be the set of bit nodes that are guessed. Using an appropriate column permutation, one can group the bit nodes of each set together such that the index of a bit freed at a later iteration is lower than that of one freed in an earlier iteration. Finally, a suitable row permutation then renders the matrix in the described form.

A.2 Proof of Observation 4.3.1

Before we begin the proof of the observation, we present the following result.

Lemma A.2.1. *For a code of size n, let V represent the set of bit nodes and $f : 2^V \longrightarrow 2^V$ represent the mapping of each $A \subseteq V$ to the unique maximal stopping set contained in it (assuming that the bits corresponding to nodes in A are erased). Let for each $0 \leq i \leq n$, $\mathcal{N}_i \triangleq \{S \subseteq V : |S| = i, f(S) \neq \emptyset\}$. Then*

$$\sum_{Z_i \in \mathcal{N}_i} \frac{|f(Z_i)|}{|\mathcal{N}_i|} \geq \sum_{Z_{i-1} \in \mathcal{N}_{i-1}} \frac{|f(Z_{i-1})|}{|\mathcal{N}_{i-1}|}, \qquad i \in \{2, \ldots, n\}. \tag{A.2}$$

Proof. Partition $\mathcal{N}_i$ into two $\mathcal{N}_i^a = \{w \in \mathcal{N}_i : v \subsetneq w \Rightarrow f(v) = \emptyset\}$ and $\mathcal{N}_i^b = \mathcal{N}_i \setminus \mathcal{N}_i^a$. Note that each $w \in \mathcal{N}_i^a$ satisfies $f(w) = w$. Thus, we see that

$$\sum_{w \in \mathcal{N}_i} \frac{|f(w)|}{|\mathcal{N}_i|} = \frac{i|\mathcal{N}_i^a| + \sum_{w \in \mathcal{N}_i^b} |f(w)|}{|\mathcal{N}_i^a| + |\mathcal{N}_i^b|} \geq \frac{\sum_{w \in \mathcal{N}_i^b} |f(w)|}{|\mathcal{N}_i^b|}. \tag{A.3}$$

We now divide the proof into two cases based on the cardinalities of $\mathcal{N}_{i-1}$ and $\mathcal{N}_i^b$. First, suppose that $|\mathcal{N}_{i-1}| \leq |\mathcal{N}_i^b|$. Construct a directed graph G with a vertex set $V_G = \{n_v : v \in \mathcal{N}_{i-1}\} \cup \{n_w : w \in \mathcal{N}_i^b\} \cup \{n_s, n_t\}$. The set of edges E_G in the directed graph is given by

$$E_G = \{(n_s, n_v) : v \in \mathcal{N}_{i-1}\} \cup \{(n_w, n_t) : w \in \mathcal{N}_i^b\} \cup \{(n_v, n_w) : v \in \mathcal{N}_{i-1}, w \in \mathcal{N}_i^b\}. \tag{A.4}$$

Note that here we use the notation that an edge $e = (v, w)$ starts from v and ends in w. Define a capacity function $C : E_G \longrightarrow \mathbb{R}$ over the set of edges which takes on values as defined by:

$$C(e) = \begin{cases} 1 & \text{if } e = (n_s, y) \text{ and } y \in V_G \\ \frac{|\mathcal{N}_{i-1}|}{|\mathcal{N}_i^b|} & \text{if } e = (x, n_t) \text{ and } x \in V_G \\ \infty & \text{if } e = (n_v, n_w), n_v \in \mathcal{N}_{i-1} \text{ and } v \subseteq w \in \mathcal{N}_i^b \\ 0 & \text{otherwise} \end{cases}. \tag{A.5}$$

The n_s-n_t cut with the smallest capacity is seen to allow a flow of $|\mathcal{N}_{i-1}|$. Therefore, from the celebrated *max-flow min-cut theorem* [15], we see that there exists a flow g of value

$|\mathcal{N}_{i-1}|$ satisfying the following.

$$\begin{aligned}
\sum_{w \in \mathcal{N}_i^b} g(n_v, n_w) &= 1, & v \in \mathcal{N}_{i-1}. \qquad &\text{(A.6)} \\
\sum_{v \in \mathcal{N}_{i-1}} g(n_v, n_w) &= \frac{|\mathcal{N}_{i-1}|}{|\mathcal{N}_i^b|}, & w \in \mathcal{N}_i^b. \qquad &\text{(A.7)}
\end{aligned}$$

Therefore, one sees that

$$\begin{aligned}
\sum_{v \in \mathcal{N}_{i-1}} \frac{|f(v)|}{|\mathcal{N}_{i-1}|} &= \sum_{v \in \mathcal{N}_{i-1}} \left(\sum_{w \in \mathcal{N}_i^b} g(n_v, n_w) \right) \frac{|f(v)|}{|\mathcal{N}_{i-1}|} & \text{(A.8)} \\
&= \sum_{\substack{v \in \mathcal{N}_{i-1} \\ w \in \mathcal{N}_i^b}} \frac{g(n_v, n_w)|f(v)|}{|\mathcal{N}_{i-1}|} & \text{(A.9)} \\
&\leq \sum_{\substack{v \in \mathcal{N}_{i-1} \\ w \in \mathcal{N}_i^b}} \frac{g(n_v, n_w)|f(w)|}{|\mathcal{N}_{i-1}|} & \text{(A.10)} \\
&= \sum_{w \in \mathcal{N}_i^b} \left(\sum_{v \in \mathcal{N}_{i-1}} g(n_v, n_w) \right) \frac{|f(w)|}{|\mathcal{N}_{i-1}|} & \text{(A.11)} \\
&= \sum_{w \in \mathcal{N}_i^b} \frac{|f(w)|}{|\mathcal{N}_i^b|} \leq \sum_{w \in \mathcal{N}_i} \frac{|f(w)|}{|\mathcal{N}_i|}. & \text{(A.12)}
\end{aligned}$$

In the case that $|\mathcal{N}_{i-1}| > |\mathcal{N}_i^b|$, use the same graph with directions reversed on all the edges. Defining suitably a new capacity function (such that the min n_s-n_t cut has a capacity of $|\mathcal{N}_i^b|$) on this direction-reversed graph, one can construct a flow that can be used similarly to show that the claim holds even in this case. ❑

To begin the proof of Observation 4.3.1, let f be defined as in Lemma A.2.1. Also, let $\mathcal{S}$ denote the set of stopping sets of the code and let X be the random variable representing the set of erasures in the word received at the output of the BEC with erasure probability ϵ. Also, let $k = \min_{S \in \mathcal{S}} |S|$. Then one can show that

$$\begin{aligned}
g(\epsilon) &= E[|f(X)| \mid f(X) \neq \emptyset] & \text{(A.13)} \\
&= \frac{\sum\limits_{S \in \mathcal{S} \setminus \{\emptyset\}} |S| \sum\limits_{T \in f^{-1}(S)} \epsilon^{|T|}(1-\epsilon)^{n-|T|}}{\sum\limits_{S \in \mathcal{S} \setminus \{\emptyset\}} \sum\limits_{T \in f^{-1}(S)} \epsilon^{|T|}(1-\epsilon)^{n-|T|}} & \text{(A.14)} \\
&= \frac{\sum\limits_{i \geq k} M_i \epsilon^i (1-\epsilon)^{n-i}}{\sum\limits_{i \geq k} m_i \epsilon^i (1-\epsilon)^{n-i}} = \frac{\sum\limits_{i \geq k} M_i h_i(\epsilon)}{\sum\limits_{i \geq k} m_i h_i(\epsilon)}, & \text{(A.15)}
\end{aligned}$$

where for each $i = k, \dots, n,\ M_i, m_i$ are the number of times the term $h_i(\epsilon) \triangleq \epsilon^i(1-\epsilon)^{n-i}$ appears in the numerator and denominator, respectively. Then, one sees that

$$g'(\epsilon) = \frac{\sum\limits_{n \geq i > j \geq k} \left(\frac{M_i}{m_i} - \frac{M_j}{m_j}\right) m_i m_j h_i(\epsilon) h_j(\epsilon)}{\epsilon(1-\epsilon)\left(\sum\limits_{i \geq k} m_i h_i(\epsilon)\right)^2}. \tag{A.16}$$

Finally, one can use the inequality from Lemma A.2.1 that $\frac{M_i}{m_i} \geq \frac{M_j}{m_j}$ for $n \geq i > j \geq k$ to show explicitly that the derivative g' is non-negative in $(0,1)$.

A.3 Proof of Theorem 4.3.1

Let

$$\mathcal{B} \triangleq \left\{ \underline{x} \in \{0, 2, \dots, d_c\}^{\overline{R}n} : \sum_i x_i \in \left(\alpha d_v(n - \sqrt{n}), \alpha d_v(n + \sqrt{n})\right) \right\}, \tag{A.17}$$

$$\mathcal{A} \triangleq \left\{ \underline{x} \in \mathcal{B} : |\{i : x_i = 2\}| \leq \beta \overline{R} n \right\}. \tag{A.18}$$

Then, from (4.13), it follows that

$$p_n^{\alpha,\beta} \triangleq P[T_S \leq \beta \mid |S| \in \left(\alpha d_v(n - \sqrt{n}), \alpha d_v(n + \sqrt{n})\right)] \tag{A.19}$$

$$= \Theta\left(\frac{\sum_{\underline{r}' \in \mathcal{A}} \prod_i \binom{d_c}{r'_i}}{\sum_{\underline{r}' \in \mathcal{B}} \prod_i \binom{d_c}{r'_i}} \right) \tag{A.20}$$

$$= \Theta\left(\frac{\sum\limits_{j=-\lceil \alpha d_v \sqrt{n} \rceil}^{\lfloor \alpha d_v \sqrt{n} \rfloor} \sum\limits_{k=0}^{\beta \overline{R} n} \binom{\overline{R}n}{k} \binom{d_c}{2}^k \mathscr{C}\left(\left(\sum\limits_{i \neq 1,2} \binom{d_c}{i} x^i \right)^{\overline{R}n - k}, x^{\alpha n d_v - 2k + j} \right)}{\sum\limits_{j=-\lceil \alpha d_v \sqrt{n} \rceil}^{\lfloor \alpha d_v \sqrt{n} \rfloor} \mathscr{C}\left(\left((1+x)^{d_c} - d_c x \right)^{\overline{R}n}, x^{\alpha n d_v + j} \right)} \right). \tag{A.21}$$

For our convenience, let us define

$$L_n^\lambda \triangleq \max_{\substack{c_3, \dots, c_{d_c} \\ |\sum i c_i - n(\alpha d_v - 2\lambda \overline{R})| \leq \alpha d_v \sqrt{n}}} \left[\binom{m}{c_3, \dots, c_{d_v}, m - \sum\limits_{k>2} c_k} \prod_{j=3}^{d_c} \binom{d_c}{j}^{c_j} \right], \tag{A.22}$$

$$M_n \triangleq \max_{\substack{c_2, \dots, c_{d_c} \\ \sum |i c_i - \alpha n d_v| | \leq \alpha d_v \sqrt{n}}} \left[\binom{m'}{c_2, \dots, c_{d_v}, m' - \sum\limits_{k>1} c_k} \prod_{j=2}^{d_c} \binom{d_c}{j}^{c_j} \right], \tag{A.23}$$

where $m = \overline{\beta R} n$, $m' = \overline{R} n$. Here, it must be noted that (A.22) is suitably defined only for $\lambda \in \{\delta \in \mathbb{Q} \cap [0, \beta] : \delta \overline{R} n \in \mathbb{N}\}$. Since every term in the numerator of (A.20) is exponential in n and since the number of terms in the numerator is polynomial in n, the term with the fastest exponential growth will dominate the sum for very large n. Therefore, one sees that

$$\lim_{n\to\infty} \frac{\log p_n^{\alpha,\beta}}{n} \leq \sup_{\lambda\in[0,\beta]} \left\{ h(\lambda) + \lambda\overline{R}\log\binom{d_c}{2} + \lim_{n\to\infty}\left[\frac{\log L_n^{\lambda}}{n} - \frac{\log M_n}{n}\right]\right\}. \tag{A.24}$$

Here, we would like to mention that henceforth in the proof, we interpret the corresponding limits to be evaluated over only those indices n for which L_n^{λ} and M_n are defined. Using Lagrange multipliers for (A.22), we can show that

$$\lim_{n\to\infty} \frac{\log L_n^{\lambda}}{m} = h(t_3^*, \ldots, t_{d_v}^*, 1 - \sum_{k>2} t_k^*) + \sum_{j>2} t_j^* \log\binom{d_c}{j}, \tag{A.25}$$

where

$$t_i^* = \frac{\binom{d_c}{i}x_0^i}{1 + \sum_{k>2}\binom{d_c}{k}x_0^k}, \qquad i = 3, \ldots, d_c, \tag{A.26}$$

and x_0 is the unique solution to

$$\frac{x((1+x)^{d_c-1} - 1 - (d_c-1)x)}{(1+x)^{d_c} - d_c x - \binom{d_c}{2}x^2} = \frac{\alpha d_v - 2\lambda\overline{R}}{(1-\lambda)\overline{R}d_c}. \tag{A.27}$$

Lagrange multipliers for (A.23) yields

$$\lim_{n\to\infty} \frac{\log M_n}{m'} = h(t_2', \ldots, t_{d_v}', 1 - \sum_{k>1} t_k') + \sum_{j>1} t_j' \log\binom{d_c}{j}, \tag{A.28}$$

where

$$t_i' = \frac{\binom{d_c}{i}x_1^i}{1 + \sum_{k>1}\binom{d_c}{k}x_1^k}, \qquad i = 2, \ldots, d_c, \tag{A.29}$$

and x_1 is the unique solution to

$$\frac{x((1+x)^{d_c-1} - 1)}{(1+x)^{d_c} - d_c x} = \frac{\alpha d_v}{\overline{R}d_c}. \tag{A.30}$$

At $\beta = 0$, one sees that $m = m'$. One then finds the limiting exponent of the denominator to be the maximum of a $(d_c - 1)$-variate strictly concave function (of $t_2, \ldots, t_{d_c}$) and the limiting exponent of the numerator to be the maximum of the same concave function when the variable t_2 is set to 0. Hence,

$$\lim_{n\to\infty} \frac{\log L_n^0}{m} - \lim_{n\to\infty} \frac{\log M_n}{m'} < 0 \Rightarrow \lim_{n\to\infty} \frac{\log p_n^{\alpha,0}}{n} < 0. \tag{A.31}$$

Equations (A.24)-(A.31) imply that $\lim\limits_{n\to\infty} \frac{\log p_n^{\alpha,\beta}}{n}$ is bounded above by a continuous function of β that is strictly negative in an interval $(0, \beta^*)$ for some $0 < \beta^* < 1$. This concludes the proof.

A.4 Proof of Lemma 6.1.1

The number of times $g_1 - g_2$ changes signs is equal to the number of times $g(i) \triangleq \log \frac{g_1(i)}{g_2(i)}$ changes signs. Therefore, g must change signs at least once, else, it will lead to a contradiction because of the following.

$$\sum_{p \leq i \leq n} (g_1(i) - g_2(i)) \geq 0 \Rightarrow 1 > 1 - \sum_{0 \leq i < p} g_1(i) \geq \sum_{p \leq i \leq n} g_1(i) = 1. \tag{A.32}$$

Further, it is clear that g can change sign for $i \in \{p, p+1, \ldots, n\}$. For $p < i < n$, we see that

$$g(i) = \underbrace{\log \frac{\binom{n}{i}}{\binom{n-p}{i-p}}}_{I} \underbrace{- i \log \frac{\varepsilon_1(\overline{\varepsilon}_2)}{\varepsilon_2(\overline{\varepsilon}_1)} + n \log \frac{\overline{\varepsilon}_1}{\overline{\varepsilon}_2} - p \log \varepsilon_2,}_{II} \tag{A.33}$$

where the first term I is a convex and strictly decreasing function of i and II is a term linear in i. The first term can be extended to a continuous, strictly convex, decreasing function $\tilde{w} : [f, n] \longrightarrow \mathbb{R}$ by linear extrapolation for non-integer values. Similarly, the linear part can be extended to a linear function $\tilde{v} : [f, n] \longrightarrow \mathbb{R}$. Since a continuous, strictly convex function and a straight line can intersect it at most two places, we see that $\tilde{w} - \tilde{v}$ and hence g changes sign at most twice.

In the case of matched means i.e., $n\varepsilon_1 = p + (n-p)\varepsilon_2$, one uses Stirling's approximation [107] to show that

$$\begin{aligned} g(p) &= \log \frac{\binom{n}{p}\varepsilon_1^p(\overline{\varepsilon}_1)^{n-p}}{(\overline{\varepsilon}_2)^{n-p}} = \log \binom{n}{p} + p \log \varepsilon_1 - (n-p) \log \frac{\overline{\varepsilon}_1}{\overline{\varepsilon}_2} && \text{(A.34)} \\ &\geq p \log \frac{n}{p} - \frac{1}{2} \log(\pi n e^{\frac{1}{3}}) + p \log(\varepsilon_1) = p\left(\log \frac{n\varepsilon_1}{p} - \frac{\log(\pi n e^{\frac{1}{3}})}{2p} \right) && \text{(A.35)} \\ &\geq p\left(\frac{n\varepsilon_1}{p} - 1 - \frac{\log(\pi n e^{\frac{1}{3}})}{2p} \right) > 0. && \text{(A.36)} \end{aligned}$$

Moreover,

$$g(n) = \log \frac{\varepsilon_1^n}{\varepsilon_2^{n-p}} = n\left(\log(\frac{p}{n} + \frac{n-p}{n}\varepsilon_2) - \frac{p}{n} \log 1 - \frac{n-p}{n} \log \varepsilon_2 \right) > 0 \tag{A.37}$$

follows from Jensen's inequality. Finally since g must change sign at least once, we conclude from (A.36) and (A.37) that g changes sign twice.

A.5 Proof of Lemma 6.1.2

For notational convenience, define $\lambda \triangleq n^{-1}p$. Define two random variables $X \sim B(n, \varepsilon\overline{\lambda}+\lambda)$ and $Y \sim B(n-p, \varepsilon)$. Then, we see that $E(Y+p) = E(Y)+p = E(X)$. Then,

$$\begin{aligned} \sum_i if(i,n,\varepsilon\overline{\lambda}+\lambda) &= p + \sum_j jf(j-p, n-p, \varepsilon) && \text{(A.38)} \\ \Rightarrow \sum_i (n-i)f(i,n,\varepsilon\overline{\lambda}+\lambda) &= p + \sum_j (n-p-j)f(j-p,n-p,\varepsilon) && \text{(A.39)} \\ \Rightarrow \sum_{i=0}^{n} F(i,n,\varepsilon\overline{\lambda}+\lambda) &= \sum_{j=0}^{n} F(j-p,n-p,\varepsilon). && \text{(A.40)} \end{aligned}$$

Since the indices in (A.40) run from 0 to n, (A.40) is exact. However, since the sum in the lemma is only from 0 up to $\overline{R}n$, we must approximate the residual part ρ that corresponds to the sum for indices between $\overline{n}$ and n. Notice that this sum is given by

$$\begin{aligned} \rho &\triangleq \sum_{i=\overline{R}n}^{n} \left| F(i,n,\varepsilon\overline{\lambda}+\lambda) - F(i-p,n-p,\varepsilon) \right| && \text{(A.41)} \\ &= \sum_{i=\overline{R}n}^{n} \left| \overline{F(i,n,\varepsilon\overline{\lambda}+\lambda)} - \overline{F(i-p,n-p,\varepsilon)} \right| && \text{(A.42)} \\ &\leq Rn\Big(\max\left(\overline{F(\overline{R}n,n,\varepsilon\overline{\lambda}+\lambda)}, \overline{F(\overline{R}n-p,n-p,\varepsilon)} \right)\Big). && \text{(A.43)} \end{aligned}$$

Since binomial random variables are generated by a sum of binary i.i.d. random variables, one can use Azuma-Hœffding's inequality [2] to conclude that

$$F(\overline{R}n-p,n-p,\varepsilon) \leq e^{-2\alpha^2\overline{\lambda}n} \qquad \text{(A.44)}$$

$$F(\overline{R}n,n,\varepsilon\overline{\lambda}+\lambda) \leq e^{-2\alpha^2\overline{\lambda}^2 n} \qquad \text{(A.45)}$$

for some $\alpha > 0$. Combining equations (A.44), (A.45) with (A.43), we conclude the result.

A.6 Proof of Lemma 6.1.3

Let V denote the set of code bits. Let $f : 2^V \longrightarrow 2^V$ denote the function mapping each set $A \subset V$ to the maximal stopping set contained in A. Also, for every $i = 1, .., n$, denote X_i to be a random variable selecting uniformly at random a subset of size i from V. Let Y_{X_i}

denote the random variable that picks at random an element of X_i. Then, we see that

$$E[|f(X_i \setminus Y_{X_i})|] = \sum_{\substack{X \subseteq V \\ |X|=i}} \frac{\left\{ \sum_{v \in X} \frac{|f(X \setminus \{v\})|}{|X|} \right\}}{\binom{n}{i}} = \sum_{\substack{X \subseteq V \\ |X|=i \\ v \in X}} \frac{|f(X \setminus \{v\})|}{i\binom{n}{i}} \tag{A.46}$$

$$= \sum_{\substack{X \subseteq V \\ |X|=i-1}} \frac{|f(X)|}{\binom{n}{i-1}} = E[|f(X_{i-1})|] = S_{i-1}(\mathcal{C}). \tag{A.47}$$

$$\therefore \quad S_i(\mathcal{C}) - S_{i-1}(\mathcal{C}) = \sum_{\substack{X \subseteq V \\ |X|=i \\ v \in X}} \frac{|f(X)| - |f(X \setminus \{v\})|}{i\binom{n}{i}} \tag{A.48}$$

$$= \sum_{\substack{X \subseteq V \\ |X|=i}} \frac{\sum_{v \in f(X)} [|f(X)| - |f(X \setminus \{v\})|]}{i\binom{n}{i}} \tag{A.49}$$

$$\geq \sum_{\substack{X \subseteq V \\ |X|=i}} \frac{|f(X)|}{i\binom{n}{i}} = \frac{S_i(\mathcal{C})}{i}. \tag{A.50}$$

A.7 Proof of Theorem 6.1.1

Just as in the proof of Lemma 6.1.2, we denote $\lambda = n^{-1}p$. We notice that

$$E_{\mathcal{C},p}(\varepsilon) - E_{\mathcal{C},0}(\varepsilon(1-\lambda)+\lambda) = \sum_{0 \leq i \leq n} \frac{S_i(\mathcal{C})}{n}[f(i-p, n-p, \varepsilon) - f(i, n, \varepsilon(1-\lambda)+\lambda)] \tag{A.51}$$

$$= \sum_{0 \leq i \leq n} \frac{\Delta S_{i+1}(\mathcal{C})}{n}[F(i, n, \varepsilon(1-\lambda)+\lambda) - F(i-p, n-p, \varepsilon)] \tag{A.52}$$

$$\approx \sum_{0 \leq i \leq \overline{R}n} \underbrace{\frac{\Delta S_{i+1}(\mathcal{C})}{n}[F(i, n, \varepsilon(1-\lambda)+\lambda) - F(i-p, n-p, \varepsilon)]}_{\triangleq \Sigma} \tag{A.53}$$

where we have used $\Delta S_{i+1}(\mathcal{C})$ as a shorthand for $(S_{i+1}(\mathcal{C}) - S_i(\mathcal{C}))$. Also, in moving from (A.52) to (A.53), we use Lemma 6.1.2 and the fact that $\frac{S_i(\mathcal{C})}{n} \leq 1$ to drop/neglect the contribution from the non-convex regions (namely $i > \overline{R}n$) (since they amount to a term that is exponentially small in n). Moreover, by convexity of $S_i(\mathcal{C})$ for $i = 1, \ldots, \overline{R}n$ we are guaranteed the monotonic increase of $\Delta S_{i+1}(\mathcal{C}) > 0$ with the index i. Corollary 2 guarantees that $\Psi(i) \triangleq (F(i, n, \varepsilon(1-\lambda)+\lambda) - F(i-p, n-p, \varepsilon))$ changes sign at most once as i increases from 0 to n. Let L be the smallest non-zero index for which the term $\Psi(\cdot)$ is non-positive.

Then, one sees that

$$\begin{align}
\Sigma &\leq \sum_{i=1}^{\overline{R}n} \frac{\Delta S_{i+1}(\mathcal{C})}{n} \Psi(i) \tag{A.54}\\
&\leq \sum_{i=0}^{L} \frac{\Delta S_{L+1}(\mathcal{C})}{n} |\Psi(i)| - \sum_{i=L+1}^{\overline{R}n} \frac{\Delta S_{L+2}(\mathcal{C})}{n} |\Psi(i)| \tag{A.55}\\
&= \underbrace{\frac{\Delta S_{L+1}(\mathcal{C})}{n} \sum_{i=0}^{\overline{R}n} \Psi(i)}_{\sigma_1} - \underbrace{\left(\frac{\Delta S_{L+2}(\mathcal{C}) - \Delta S_{L+1}(\mathcal{C})}{n} \right) \sum_{i=L+1}^{\overline{R}n} |\Psi(i)|}_{\sigma_2} \tag{A.56}
\end{align}$$

Finally, note the first term is guaranteed to be exponentially small in n due to Lemma 6.1.2 and (A.40) thereby establishing the negativity of Σ for almost the whole range of the hypothesis.

APPENDIX B

PROOFS OF RESULTS IN PART II

B.1 Elementary Results for Chapter 7

Lemma B.1.1. *Let $l \in \mathbb{N}$, $l \geq 2$ and let*

$$M = \{\underline{\alpha} \in [0,1]^l : \alpha_1 = 1, \alpha_l = 1, \text{ and } \alpha_i + \alpha_{i+1} \geq 1 \;\; i \in 1, \ldots, l-1\}. \tag{B.1}$$

Then, $T_l \triangleq \min_{\underline{\alpha} \in M} \sum_i \alpha_i = \lceil \frac{l+1}{2} \rceil$.

Proof. Suppose that l is even. Then,

$$T_l = \min_{\underline{\alpha} \in M} (\alpha_1 + (\alpha_2 + \alpha_3) + \ldots + (\alpha_{l-2} + \alpha_{l-1}) + \alpha_l) \geq \frac{l+2}{2}. \tag{B.2}$$

It can be seen that this is attained at $\underline{\alpha}^* = (\alpha_1^*, \ldots, \alpha_l^*)$, where $\alpha_i^* = 1$ if $i = 1$ and $i = l$ and $\alpha_i^* = 0.5$ otherwise.

In the case that l is odd, we see that

$$T_l = 2 + \min_{\underline{\alpha} \in M} ((\alpha_2 + \alpha_3) + \ldots + (\alpha_{l-3} + \alpha_{l-2}) + \alpha_{l-1}) \geq \frac{l+1}{2}. \tag{B.3}$$

Clearly, the equality is attained at $\underline{\alpha}^* = (\alpha_1^*, \ldots, \alpha_l^*)$, where $\alpha_i^* = 1$ if i is odd and $\alpha_i^* = 0$ otherwise. ❑

Lemma B.1.2. *Let $l \in \mathbb{N}$, $l > 1$. Let $M = \{\underline{\alpha} \in [0,1]^l : \alpha_i + \alpha_{i+1} \geq 1, \; i = 1, \ldots, l-1\}$. Then* $T_l^{'} \triangleq \min\limits_{\underline{\alpha} \in M} \sum\limits_{i=1}^{l} \alpha_i = \lfloor \frac{l}{2} \rfloor$.

Proof. Let r be even. Then

$$T_r^{'} = \min_{\underline{\alpha} \in M} (\alpha_1 + \ldots + \alpha_r) = \min_{\underline{\alpha} \in M} \Big((\alpha_1 + \alpha_2) + \ldots (\alpha_{r-1} + \alpha_r) \Big) \geq \frac{r}{2}. \tag{B.4}$$

It can be verified that this minimum is attained at $\underline{\alpha} = \underline{\alpha}^* = (\alpha_1^*, \ldots, \alpha_r^*)$, where $\alpha_i^* = \frac{1}{2}$, $i = 1, \ldots, r$. In the case that r is odd,

$$T_r^{'} \geq \min_{\underline{\alpha} \in M} \Big((\alpha_1 + \alpha_2) + \ldots (\alpha_{r-2} + \alpha_{r-1}) \Big) + \min_{\underline{\alpha} \in M} \alpha_r \geq \frac{r-1}{2}. \tag{B.5}$$

Moreover, it can be seen that the bound is reached at $\underline{\alpha}^* = (\alpha_1^*, \ldots, \alpha_r^*)$, where $\alpha_i^* = 0$ if i is odd and $\alpha_i^* = 1$ otherwise. ❑

Lemma B.1.3. *Let* $n \in \mathbb{N}$ *and* $a_1, \ldots, a_n, b_1, \ldots, b_n \in (0, \infty)$ *and* $\beta = \min\{\frac{\sum_k b_k}{b_i} : k = 1, \ldots, n\}$. *Then,*

$$\sum_k \frac{a_k}{b_k} \geq \beta \frac{\sum_k a_k}{\sum_k b_k}. \tag{B.6}$$

Proof.

$$\sum_k \frac{a_k}{b_k} = \frac{1}{\sum_j b_j} \sum_k \frac{a_k(b_1 + \ldots + b_n)}{b_k} \geq \frac{\beta \sum_k a_k}{\sum_j b_j}. \tag{B.7}$$

❑

B.2 Proof of Theorem 7.2.2

Since the radius of communication r_n is chosen to be the asymptotic threshold for connectivity, we notice that a random deployment of n nodes will be connected with high probability. Notice that from [88] that this transmission radius guarantees that the maximum node degree M_n and the average node degree $\overline{M}_n$ satisfy the following with high probability.

$$M_n = nr_n^2(1 + o(1)) = \log n(1 + o(1)) \tag{B.8}$$

$$\overline{M}_n = nr_n^2(1 + o(1)) = \log n(1 + o(1)). \tag{B.9}$$

In FTS, every pair of neighboring nodes are assigned as a parent-child pair. After the first two phases of FTS, the parent sets and the children are identified along with the fractions that each node transmits in the third phase. Construct a directed graph $\overrightarrow{G_n}$ with these n nodes in the following manner. Using the assigned children and parent sets, designate a direction to each edge of the network such that the starting node is a parent of the ending node. Equivalently, the ending node of each edge is a child of the starting node. For each node v, allow d_v^+, d_v^- to denote the number of incoming and outgoing edges incident on v in $\overrightarrow{G_n}$. Note that for all nodes $v \neq s$, $d_v^+ > 0$, however d_v^- can be zero. After the first phase of the FTS, the fraction determined by each node $v \neq s$ for which $d_v^- > 0$ is given by

$$\alpha_v = \max\left\{\frac{1}{d_w^+} : (v, w) \in \overrightarrow{E}(\overrightarrow{G_n})\right\} \geq \frac{\sum\limits_{(v,w) \in \overrightarrow{E}(\overrightarrow{G_n})} \frac{1}{d_w^+}}{d_v^-}. \tag{B.10}$$

Note that $\min\{\frac{\sum_{v\in V} d_v^-}{d_u^-} : u \in V\} \geq \frac{n\overline{M}_n}{2M_n} = \frac{n}{2}(1+o(1))$. Also, nodes that have no outgoing edge do not transmit packets. Therefore, the average cost of transmission per packet per node is given by

$$\mathcal{N}_{/p/n} = \frac{1+\sum_{v\neq s}\alpha_v}{n} = \frac{1}{n}\left(1 + \sum_{\substack{v\neq s \\ d_v^- > 0}} \frac{\sum\limits_{(v,w)\in\overrightarrow{E}(\overrightarrow{G_n})} \frac{1}{d_w^+}}{d_v^-}\right) \tag{B.11}$$

By Lemma B.1.3, the proof is completed by the following argument.

$$\begin{aligned}
\mathcal{N}_{/p/n} &\geq \frac{1}{n} + \left(\frac{1}{2}\frac{\sum\limits_{v\neq s}\sum\limits_{(v,w)\in\overrightarrow{E}(\overrightarrow{G_n})}\frac{1}{d_w^+}}{\sum_{v\neq s} d_v^-}\right)(1+o(1)) && \text{(B.12)}\\
&= \frac{1}{n} + \left(\frac{\sum\limits_{w\in V}\sum\limits_{(v,w)\in\overrightarrow{E}(\overrightarrow{G_n})}\frac{1}{d_w^+}}{n\overline{M}_n}\right)(1+o(1)) && \text{(B.13)}\\
&= \frac{1}{n} + \frac{1}{\overline{M}_n}(1+o(1)) = \frac{1}{\log n}(1+o(1)). && \text{(B.14)}
\end{aligned}$$

B.3 Proof of Theorem 8.2.3

We begin the proof with an illustration of a classification of the various sections of a pair of paths. Consider two paths from s to (i,j) as presented in Fig. B.1. Clearly, it can be seen that two paths have six different sections. In some sections, the paths are edge-disjoint (called as *L-sections*) and in others, the paths use the same edges (called as P_1-*sections*). Here, sections $2,3,5$ are L-sections and $1,4,6$ are P$_1$-sections. If we include a single node as the trivial P$_1$-section, we can insert one such trivial section between the two adjacent L-sections 2 and 3. We then notice that the P$_1$-sections and L-sections alternate. In fact, we can decompose every subgraph generated by the union of two distinct paths from the $\binom{i+j}{i}$ paths into alternating L-sections and P_1 sections. Thus, every pair of paths from source to (i,j) will be of the form $P_1[LP_1]^k$ for some $k\in\mathbb{N}$. Also, it must be noted that a subgraph generated by a pair of paths that has a decomposition comprising of k L-sections can be generated by 2^{k-1} different selections of pairs of paths. Thus, in mathematical terms, we have

$$\binom{\binom{i+j}{j}}{2} = \sum_{k\in\mathbb{N}} \sum_{\substack{\sum_\iota t_\iota^{p,k}+\sum_\kappa t_\kappa^{l,k}=i \\ \sum_\iota s_\iota^{p,k}+\sum_\kappa s_\kappa^{l,k}=j}} 2^{k-1}\prod_{\iota=1}^{k+1}\binom{t_\iota^{p,k}+s_\iota^{p,k}}{t_\iota^{p,k}}\prod_{\kappa=1}^{k} l(t_\kappa^{l,k}, s_\kappa^{l,k}), \tag{B.15}$$

where the coefficients $t_\iota^{p,k}$s and $s_\iota^{p,k}$s are non-negative integers where as $t_\kappa^{l,k}$s and $s_\kappa^{l,k}$s are positive integers. It can be immediately seen that the inner sum in (B.15) is a $2k+1$-fold

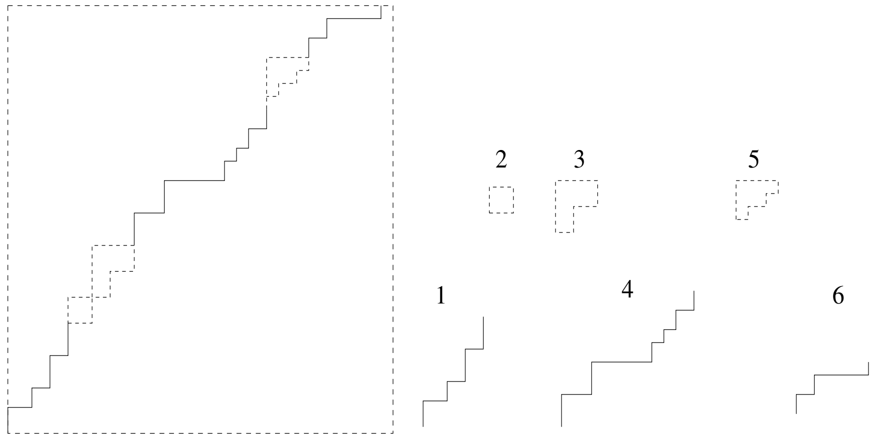

Figure B.1: Different sections of a pair of paths from the source to a node in the grid.

convolution, which in a generalized power series notation will reduce to multiplications in the following manner.

$$\binom{\binom{i+j}{j}}{2} x^i y^j = \sum_{k\in\mathbb{N}} \sum_{\substack{\sum_\iota t_\iota^{p,k} + \sum_\kappa t_\kappa^{l,k} = i \\ \sum_\iota s_\iota^{p,k} + \sum_\kappa s_\kappa^{p,k} = j}} 2^{k-1} \prod_{\iota=1}^{k+1} \left[\binom{t_\iota^{p,k}+s_\iota^{p,k}}{t_\iota^{p,k}} x^{t_\iota^{p,k}} y^{s_\iota^{p,k}} \right] \prod_{\kappa=1}^{k} \left[l(t_\kappa^{l,k}, s_\kappa^{l,k}) x^{t_\kappa^{l,k}} y^{s_\kappa^{l,k}} \right].$$

Summing over all powers of x and y results in the following.

$$\tilde{\mathcal{P}}_2 = \tilde{\mathcal{P}}_1 \sum_{k\in\mathbb{N}} 2^{k-1} [\tilde{\mathcal{P}}_1 \tilde{\mathcal{L}}]^k \tag{B.16}$$

$$\therefore\ 2\tilde{\mathcal{P}}_2 + \tilde{\mathcal{P}}_1 = \tilde{\mathcal{P}}_1 \sum_{k\in\mathbb{Z}_{\geq 0}} [2\tilde{\mathcal{P}}_1 \tilde{\mathcal{L}}]^k. \tag{B.17}$$

Note that $\mathcal{L}$ is a power series whose smallest non-zero coefficient is that of xy and hence the infinite sums in (B.16) and (B.17) converge. The same fact guarantees that the infinite sum is, in fact, invertible in the ring of power series $\mathbb{R}[[x, y]]$ and that $(1 - 2\tilde{\mathcal{L}}\tilde{\mathcal{P}}_1)$ is its unique inverse. Multiplying both the sides of (B.17) with $(1 - 2\tilde{\mathcal{L}}\tilde{\mathcal{P}}_1)$ yields the result.

The uniqueness of the solution of (8.8) can be easily shown by induction. The proof uses the fact that the equation generated by comparing the coefficients of $x^{i_1}y^{i_2}$ on both the sides of (B.17) involve only variables $l(j_1, j_2)$ whose indices j_1, j_2 satisfy $j_1 \leq i_1$ and $j_2 \leq i_2$.

B.4 Proof of Theorem 8.2.4

Two events $X_\omega^{ij}, X_{\omega'}^{ij}$ are not independent if and only if the paths $W_\omega^{ij}, W_{\omega'}^{ij}$ share at least one node. Equivalently, events $X_\omega^{ij}, X_{\omega'}^{ij}$ are not independent if and only if the subgraph generated by $W_\omega^{ij}, W_{\omega'}^{ij}$ has at least one non-trivial P$_1$-section. Also, each event $X_\omega^{ij} \cap X_{\omega'}^{ij}$ derived from internally intersecting pair of paths contributes to $\delta^{ij}(p)$ a quantity equal to p raised to the total number of distinct nodes that these paths use excluding the end nodes s and (i, j). Let S_ω^{ij} denote the intermediate nodes that lie in the path W_ω^{ij}. Then it follows that

$$\delta^{ij}(p) = \sum_{\omega,\omega' : |S_\omega^{ij} \cap S_{\omega'}^{ij}| > 2} p^{|S_\omega^{ij} \cup S_{\omega'}^{ij}|}. \tag{B.18}$$

Just as in (B.15), we note that every pair of distinct paths that are not independent can be generated by alternating P$_1$-sections and L-sections, *i.e.*, as P$_1$LP$_1$, P$_1$LP$_1$LP$_1$ and so on. Suppose that $p = 1$. Then, counting all subgraphs generated by pairs of suitable paths (*i.e.*, those containing at least one non-trivial P$_1$-section) taking into consideration their correct multiplicity factor, we see that

$$\delta^{ij}(1) = \Bigg[\sum_{k\in\mathbb{N}} \sum_{\substack{\sum_\iota t_\iota^{p,k} + \sum_\kappa t_\kappa^{l,k} = i \\ \sum_\iota s_\iota^{p,k} + \sum_\kappa s_\kappa^{l,k} = j}} 2^{k-1} \prod_{\iota=1}^{k+1} \tbinom{t_\iota^{p,k} + s_\iota^{p,k}}{t_\iota^{p,k}} \prod_{\kappa=1}^{k} l(t_\kappa^{l,k}, s_\kappa^{l,k})\Bigg] - l(i,j), \tag{B.19}$$

where, again, the coefficients $t_\iota^{p,k}$'s and $s_\iota^{p,k}$'s are non-negative integers where as $t_l^{l,k}$'s and $s_l^{l,k}$'s are positive integers. To evaluate the quantity $\delta^{ij}(p)$, one must use appropriate powers of p as weights for various P$_1$- and L-sections, which as is described in (B.18) is the number of nodes in the corresponding subsection. Using the appropriate weights yields

$$\begin{aligned}\delta^{ij}(p) = p^{-1}\Bigg\{ \sum_{k\in\mathbb{N}} 2^{k-1} \sum_{\substack{\sum_\iota t_\iota^{p,k} + \sum_\kappa t_\kappa^{l,k} = i \\ \sum_\iota s_\iota^{p,k} + \sum_\kappa s_\kappa^{l,k} = j}} &\Bigg[\prod_{\iota=1}^{k+1} \left(p^{t_\iota^{p,k} + s_\iota^{p,k}} \tbinom{t_\iota^{p,k} + s_\iota^{p,k}}{t_\iota^{p,k}} \right) \times \\ &\prod_{\kappa=1}^{k} \left(l(t_\kappa^{l,k}, s_\kappa^{l,k}) p^{2t_\kappa^{l,k} + 2s_\kappa^{l,k} - 1} \right) \Bigg] \Bigg\} - l(i,j)p^{2i+2j-2}.\end{aligned} \tag{B.20}$$

As before, multiplying the above equation by $x^i y^j$ and summing over all indices yields

$$\Delta(x,y) = p^{-1}\left(\frac{1}{2} \sum_{k\in\mathbb{N}} \mathcal{P}_1(p,x,y)[2\mathcal{P}_1(p,x,y)\mathcal{L}(p,x,y)]^k - \mathcal{L}(p,x,y) \right). \tag{B.21}$$

Again, noting that the minimum degree of $\mathcal{L}$ is xy (provided $p > 0$), we conclude both the invertibility and the convergence of the infinite sum in (B.21). Multiplying both sides by the inverse $(1 - 2\mathcal{P}_1(p,x,y)\mathcal{L}(p,x,y))$ yields the result. Lastly, the uniqueness of the solution for Δ follows from the same argument of uniqueness as that of the Theorem 8.2.3.

APPENDIX C

PROOFS AND DETAILS FOR CHAPTER 10

C.1 Proof of Theorem 10.4.1

The theorem is proved by means of the following lemmas. We first obtain a solution to the steady-state equation of the ISD_2 Markov chain. Since the ISD_2 chain is irreducible, this solution is indeed the unique steady-state probability distribution of the same chain.

Lemma C.1.1. *There exists a solution to the steady state of the ISD_2 chain such that the steady-state probabilities of the active states, and those of the passive states belonging to layers $b = 1, 2, \ldots, B-1$ in the ISD_2 Markov chain are equal. In other words, $\forall\ k \in \{1, \ldots, B\}, l, b \in \{0, \ldots, B-1\}, x \notin \mathcal{D} \triangleq \Big\{(0,0),\ \textit{and}\ (\sqrt{N}-1, \sqrt{N}-1)\Big\}$,*

$$\psi_{S_k} = \psi_{D_l} = \psi_{(x,b)}. \tag{C.1}$$

Proof. Let us assume that $\psi_{S_1} = V_0$. By symmetry, $\psi_{D_{B-1}} = V_0$, since the distributions of contact times with the source and destination are equal. Consider the following set of states.

$$\widehat{S} = \{S_k : 2 \le k \le B\} \bigcup \{D_l : 0 \le l \le B-2\} \bigcup \{(x,b) : x \notin \mathcal{D}, 1 \le b \le B-1\}. \tag{C.2}$$

It can be shown that the steady-state probabilities of each state A in the above set of $N(B-1)+2$ states satisfy equations of the following form.

$$\psi_A \quad = \sum_{A' \in \widehat{S} \bigcup \{S_1, D_{B-1}\}} \alpha_A(A')\psi_{A'}. \tag{C.3}$$

Further, the coefficients in the above can be easily determined and satisfy

$$\sum_{A' \in \widehat{S} \bigcup \{S_1, D_{B-1}\}} \alpha_A(A') = 1. \tag{C.4}$$

From (C.4), we see that $\psi_A = V_0$ for each $A \in \widehat{S} \cup \{S_1, D_{B-1}\}$ satisfies (C.3), thereby completing the proof. ❑

Thus, we have determined the steady-state distribution for $N(B-1)$ out of the $N(B+1)$ states in the system. Hence, we only need to determine the steady-state distributions for the remaining $2N$ states belonging to the layers $b = 0$ and $b = B$. We do this by means of the following lemma.

Lemma C.1.2. *For the ISD_2 Markov chain, the steady-state distributions satisfy the following property*

$$\sum_{b=0}^{B} \psi_{(x,b)} = \psi_{\widetilde{S}} + \sum_{k=1}^{B} \psi_{S_k} = \psi_{\widetilde{D}} + \sum_{l=0}^{B-1} \psi_{D_l} = \frac{1}{N}, \qquad x \notin \mathcal{D} \tag{C.5}$$

Proof. This follows from the fact that each of the above sums represent the *marginal distributions* of the steady state with respect to the location in the grid. Since the random walk is independent of the buffer states, these marginal probabilities are exactly the same as the steady-state probability distribution for the random walk itself, each of which is $\frac{1}{N}$ in our model. ❑

It now follows from the above lemmas that we only need to calculate the probability distribution for the state $\widetilde{D}$ in order to estimate the throughput. Due to the well-known connection between reversible Markov chains and electrical networks [11], the problem reduces to the computation of potentials in an electric mesh formed by a grid of unit resistances, as shown in Figure C.1. Hence, we use the *electrical potential method* on the $b = 0$ layer to obtain the steady-state distributions of the remaining states in the ISD_2 chain, having known that the probabilities for states S_1 and D_0 are V_0 each. Knowing this, one can easily see that $\psi_{(0,1)} = \psi_{(1,0)} = \frac{5V_0}{2}$. We just need to calculate the potential at the location D_0 in the figure. Observing that the equivalent resistance grid is symmetric, we can discard one half of the network, as shown by dotted lines in Figure C.1. Moreover, the resistive mesh between the nodes X and Y itself can be reduced to its equivalent resistance $g(N)$ as shown in the figure. Let V_Y be the potential at Y. The corresponding steady-state equations then reduce to the following.

$$\left(V_Y - \frac{5V_0}{2}\right) g(N) \quad + \quad \left(V_Y - \psi_{\widetilde{D}} - V_0\right) = 0 \tag{C.6}$$

$$\psi_{\widetilde{D}} \quad = \quad \frac{3V_0 + 2V_Y}{2}. \tag{C.7}$$

The solution for the above system is given by the following.

$$V_Y = 5\frac{1+\gamma(N)}{2}V_0, \tag{C.8}$$

$$\psi_{\widetilde{D}} = \frac{5\gamma(N)+8}{2}V_0, \tag{C.9}$$

where, $\gamma(N) = \frac{1}{g(N)}$. Exact value of $\gamma(N)$ can be obtained using Laplacian methods described in [59], and the difference equations involved are summarized in Theorem 10.4.1. For a fairly large grid, 25×25 or more, it can be shown (either by exact analysis or by using numerical tools) that $\left[\frac{5\gamma(N)+8}{2}\right] \approx 3\log N$. Finally, we can compute the exact expression for

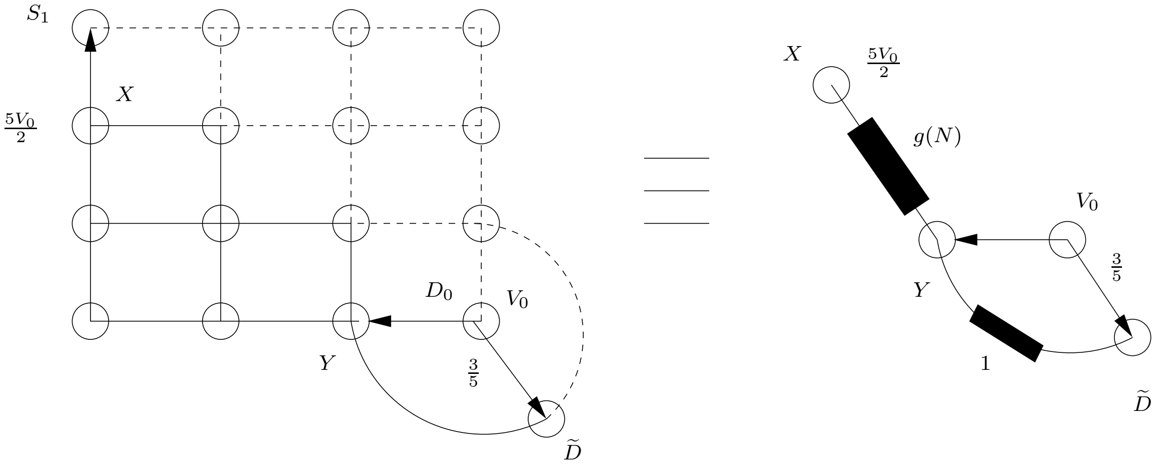

Figure C.1: Solution of the ISD_2 chain by Potential method.

the throughput due to a single node from Lemmas C.1.1, C.1 and (C.9) using the following.

$$\frac{\mathcal{C}_{s,d}}{n} = \sum_{l=0}^{B-1}\psi_{D_l} = \frac{1}{N}\frac{\sum_{l=0}^{B-1}\psi_{D_l}}{\psi_{\widetilde{D}} + \sum_{l=0}^{B-1}\psi_{D_l}}. \tag{C.10}$$

C.2 Proof of Theorem 10.5.1

For the sake of clarity, we just provide the outline of the proof. Note that the computation of throughput follows from the identification of the steady-state probabilities for the MSD_2 chain. So, let $\psi : \{0, 1, 2, 3\} \times \{0, 1, \ldots, B\} \longrightarrow [0, 1]$ to denote the steady-state probabilities of the MSD_2 chain. Since the state transition for $\mathcal{R}$ uses the steady-state distribution probabilities for the locations, we can verify that

$$\Pr[\mathcal{R} = r] = \begin{cases} N^{-2}(N-1) & r = 1, 2 \\ N^{-2} & r = 3 \\ N^{-2}(N-1)^2 & r = 4 \end{cases} \tag{C.11}$$

Also, using the properties of $\mathbf{\Phi}$ and the (C.11), we can conclude the following.

$$\sum_b \psi(r, b) = \Pr[\mathcal{R} = r] \tag{C.12}$$

$$\psi(3, b) = \frac{\psi(1, b) + \psi(1, b)}{2(N-1)} \tag{C.13}$$

$$\psi(4, b) = \frac{(N-1)}{2}(\psi(1, b) + \psi(2, b)) \tag{C.14}$$

Note that (C.12) simply follows from the fact that the marginal for $\mathcal{R}$ can be obtained by summing over all possible buffer occupancies. Also, while the scaling factors for (C.13) and (C.14) are obtained from the marginal distribution, the actual relation follows from the structure of $\mathbf{\Phi}$ and they can be intuitively understood thus. Since the distribution of a node does not vary during times when the state is at $\mathcal{R} = 3, 4$, the distribution of the buffer distribution depends on the probabilities of arriving at these states from those that correspond to $\mathcal{R} = 1, 2$. However, the probabilities of arriving at states with $\mathcal{R} = 3, 4$ from those with $\mathcal{R} = 1$ is the same as that from states with $\mathcal{R} = 2$. Thus, we can expect $\psi(3, \cdot)$ and $\psi(4, \cdot)$ to be scaled versions of each other. Other properties that can be shown exploiting the centrosymmetric structure of $\mathbf{\Phi}$ but are not detailed here include the following.

$$\psi(b, 1) = \psi(B - b, 2) \tag{C.15}$$

$$\psi(1, 1) = (1 - p_c)\psi(2, 1) \tag{C.16}$$

$$\psi(2, 1) = \psi(3, 1) = \ldots = \psi(B - 1, 1) \tag{C.17}$$

$$\psi(B, 1) = \frac{1 - \alpha(N)(1 - p_c)}{\beta(N)}\psi(2, 1). \tag{C.18}$$

Note that in (C.18), the constants $\alpha(N), \beta(N)$ are computed solely using the knowledge of the underlying probability transition matrix P pertaining to the random walk of the nodes on the network and are independent of the buffer occupancy parameter. The above properties suffice to uniquely identify $\phi(r, b)$ for all values of $\mathcal{M}$ and $\mathcal{R}$. As an illustration,

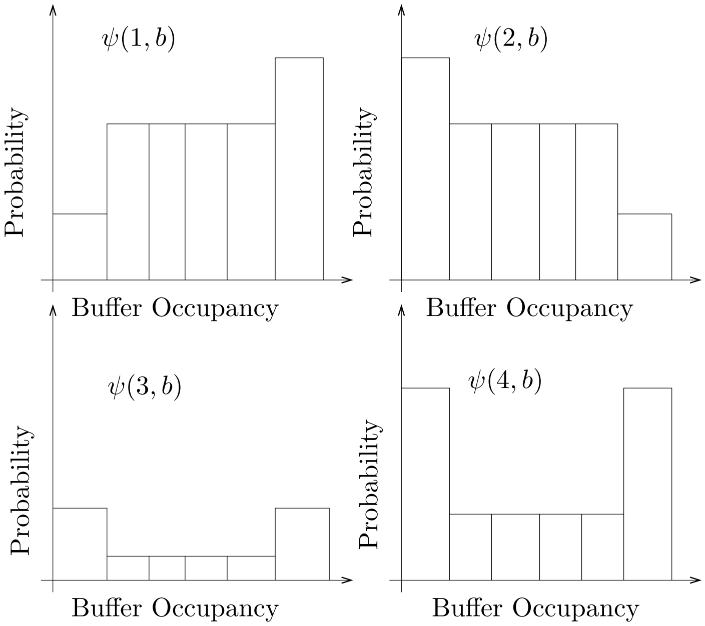

Figure C.2: Illustration of the Steady-State Distribution Solution obtained for the two-hop mobile source mobile destination unicast problem.

Figure C.2 presents the typical plot of the steady-state distribution as a function of the buffer occupancy and the partial location information $\mathcal{R}$. Finally, using the computed ψ, one can compute the throughput just as we did for the ISD_2 model.

APPENDIX D

PROOFS AND DETAILS FOR CHAPTER 11

D.1 Proof of Lemma 11.2.1

Consider Γ_i^- for some $i > 1$ and the states of the system at some time $l \in \mathbb{N}$. Γ_i^- represents transitions from a states that have the form $(n_1(l), n_2(l), \ldots, n_{h-1}(l) = i)$ to states of the form $(n_1(l+1), n_2(l+1), \ldots, n_{h-1}(l+1) = i-1)$. Since $n_{h-1}(l+1) = n_{h-1}(l) - 1$, it must be that $Y_{i-1}(l) = 0$ and that the channel must have erased the packet transmitted by v_{h-2} (else, $n_{h-1}(l+1)$ would been equal to $n_{h-1}(l)$). Therefore, for all channel realizations of $\{X_i(l)\}_{i=0}^{h-3}$, it is true that the state transition must obey

$$1 + n_1(l) + \sum_{i=2}^{h-2} n_i(l) \prod_{j=1}^{i-1}(m_j + 1) \leq 1 + n_1(l+1) + \sum_{i=2}^{h-2} n_i(l+1) \prod_{j=1}^{i-1}(m_j + 1). \tag{D.1}$$

However, $\left(1 + n_1(l) + \sum_{i=2}^{h-2} n_i(l) \prod_{j=1}^{i-1}(m_j + 1)\right)$ is the index of the row corresponding to the state $\mathbf{n}(l)$ within Γ_i^- and $\left(1 + n_i(l+1) + \sum_{i=2}^{h-2} n_1(l+1) \prod_{j=1}^{i-1}(m_j + 1)\right)$ is the index of the column corresponding to $\mathbf{n}(l+1)$ within Γ_i^-. Therefore, all possible transitions in Γ_i^- correspond to transitions from states to other state that involve a non-positive change in the row-index. Therefore, Γ_i^- is upper triangular. Finally, since each diagonal term of Γ_i^- is bounded below by $\overline{\varepsilon}_h \prod_{i=0}^{h-1} \varepsilon_i$, we conclude that

$$\det(\Gamma_i^-) \geq \left(\overline{\varepsilon}_h \prod_{i=0}^{h-1} \varepsilon_{i+1}\right)^{\prod_{j=1}^{h-2}(m_j+1)} > 0. \tag{D.2}$$

Now consider a transition under Γ_i^+ for $i < h-1$ from a state that has the form $(n_1(l), n_2(l), \ldots, n_{h-1}(l) = i)$ to another that has the form $(n_1(l+1), n_2(l+1), \ldots, n_{h-1}(l+1) = i+1)$ after an epoch. Since $n_{h-1}(l+1) = n_{h-1}(l) + 1$, it must be that the packet transmitted during this epoch on the link (v_{h-2}, v_{h-1}) must have reached successfully, *i.e.*, $Y_{h-2}(l) = 1$. By an argument similar to the above one, we can show that Γ_i^+ is lower triangular. However, its singularity for $h \geq 4$ follows from the fact that certain diagonal terms are zero. For example, consider the probability of transition from state $(n_1(l) =$

$0, \ldots, n_{h-2}(l) = 0, n_{h-1}(l) = i)$ to the state $(n_1(l) = 0, \ldots, n_{h-2}(l) = 0, n_{h-1}(l) = i+1)$ which corresponds to the $(\Gamma_i^+)_{11}$. However, this transition is impossible when $h > 2$, since the node v_{h-2} has no packets to send during this epoch. Thus, $\det(\Gamma_i^+) = 0$ provided $h \geq 3$.

Finally, the fact that $I - \Lambda_i$ is non-singular follows from the fact that $(I - \Lambda_i)$ is diagonal dominant [48] since $(I - \Lambda_i)_{kk} \geq \sum_{k' \neq k} |(I - \Lambda_i)_{kk'}|$. On the other hand, there exists at least one k for which the inequality is strict, which is sufficient to guarantee the non-singularity of such a matrix.

D.2 Proof of Theorem 11.2.1

We proceed by mathematical induction on the time index l. Clearly, the condition holds for $l = 0$. Suppose that the result is true for all node indices and for all times $l = 0, \ldots, k-1$ for some $k \in \mathbb{N}$. We consider the states of the node v_i for some $i = 2, \ldots, h-1$ in both the chains at the time instant $k-1$. One of the two following cases must apply.

1. $\underline{n_i(k-1) = \tilde{n}_i(k-1)}$: In this case we note that

$$\tilde{n}_i(k) - \tilde{n}_i(k-1) = \sigma(\tilde{n}_{i-1}(l))X_i(l)\sigma(m_i - \tilde{n}_i(l)) - \sigma(\tilde{n}_i(k-1))X_i(k-1). \tag{D.3}$$

From (11.3) and (11.9), the first term signifying the packet arrival is seen to satisfy

$$\begin{aligned} \sigma(\tilde{n}_{i-1}(l))X_i(l)\sigma(m_i - \tilde{n}_i(l)) &\leq \sigma(n_{i-1}(l))X_i(l)\sigma(m_i - n_i(l)) & \text{(D.4)} \\ &\leq \sigma(n_{i-1}(l))X_i(l)\sigma(m_i - n_i(l) + Y_{i+1}(k-1)) & \text{(D.5)} \\ &= Y_i(k-1). & \text{(D.6)} \end{aligned}$$

Similarly, (11.3) and (11.9) bound the term corresponding to the packet departure by

$$\sigma(\tilde{n}_i(k-1))X_i(k-1) \geq Y_{i+1}(k-1). \tag{D.7}$$

Comparing (D.3), (D.6), (D.7) and (11.4), we see that

$$\tilde{n}_i(k) - \tilde{n}_i(k-1) \leq Y_i(k-1) - Y_{i+1}(k-1) = n_i(k) - n_i(k-1). \tag{D.8}$$

Rearranging terms in (D.8), we conclude the result.

2. $\underline{n_i(k-1) \geq \tilde{n}_i(k-1)+1}$: In this case, we note that

$$\begin{aligned}
\tilde{n}_i(k) &= \tilde{n}_i(k-1) + \sigma(\tilde{n}_{i-1}(l))X_i(l)\sigma(m_i - \tilde{n}_i(l)) - \sigma(\tilde{n}_i(k-1))X_i(k-1) && \text{(D.9)}\\
&\leq \tilde{n}_i(k-1) + 1 - \sigma(\tilde{n}_i(k-1))X_i(k-1) && \text{(D.10)}\\
&\leq \tilde{n}_i(k-1) + 1 - Y_{i+1}(k-1) && \text{(D.11)}\\
&\leq n_i(k-1) - Y_{i+1}(k-1) && \text{(D.12)}\\
&\leq n_i(k-1) + Y_i(k-1) - Y_{i+1}(k-1) = n_i(k). && \text{(D.13)}
\end{aligned}$$

Thus, we have the following.

$$n_i(k) \geq \tilde{n}_i(k), \qquad i = 2, \dots h-1. \tag{D.14}$$

The proof is then complete by following the above procedure exactly for the node v_1.

D.3 Proof of Theorem 11.2.4

Consider two approximate solutions $(\mathbf{r}^a, \mathbf{p_b}^a)$ and $(\mathbf{r}^b, \mathbf{p_b}^b)$ such that $r_h^a = r_h^b = \delta$ with $0 < \delta < 1$. The important idea is to notice that $\varphi(0|r_{h-1}, \varepsilon_h, 0)$ is a strictly decreasing function of r_{h-1} when the other parameters are kept fixed. This follows from the fact that

$$\frac{\partial \varphi(0|r_{h-1}, \varepsilon_h, 0)}{\partial r_{h-1}} \propto \frac{-\left(\frac{1}{1-r_{h-1}}\right)^2}{\left[1 + \frac{\alpha_0}{\beta}\left(\sum_{l=0}^{m_{h-1}-1} \frac{\alpha^l}{\beta^l}\right)\right]^2} < 0. \tag{D.15}$$

An easy way in understand this behavior is to notice that α, α_0 increase with r_{h-1}, while β decreases with r_{h-1}. Therefore, from (11.25), it follows that

$$r_h^a = r_h^b \;\Rightarrow\; \varphi(m_{h-1}|r_{h-1}^a, \varepsilon_{i+1}, 0) = \varphi(m_{h-1}|r_{h-1}^b, \varepsilon_{i+1}, 0) \;\Rightarrow\; r_{h-1}^a = r_{h-1}^b. \tag{D.16}$$

Therefore, from (11.24), it must follow that ${p_b}_{h-1}^a = {p_b}_{h-1}^b$. We can then use the monotonicity of $\varphi(0|r_{h-2}, \varepsilon_{h-1}, {p_b}_{h-1})$ in conjunction with the already shown results that $r_{h-2}^a = r_{h-2}^b$ and ${p_b}_{h-2}^a = {p_b}_{h-2}^b$. We can extend this inductively to show that $\mathbf{r}^a = \mathbf{r}^b$ and $\mathbf{p_b}^a = \mathbf{p_b}^b$. Therefore, for each $\delta > 0$, there is at most one unique solution satisfying $r_h = \delta$.

Now, consider two approximate solutions $(\mathbf{r}^a, \mathbf{p_b}^a)$ and $(\mathbf{r}^b, \mathbf{p_b}^b)$ such that $\delta_a = r_h^a < r_h^b = \delta_b$ with $0 < \delta_a < \delta_b < 1$. By monotonicity of $\varphi(0|r_{h-1}, \varepsilon_h, 0)$, we have $r_{h-1}^a < r_{h-1}^b$. From (11.24), we notice that ${p_b}_i$ is also a strictly increasing function in both its variables r_i

and p_{bi+1}. Therefore, $p_{b}{}_{h-1}^{a} < p_{b}{}_{h-1}^{b}$. Again, proceeding inductively from the last node to the first each time noticing the monotonic growth of (11.25) and (11.25), we conclude that

$$\begin{array}{lcl} r_i^a & < & r_i^b \\ p_{b}{}_i^a & < & p_{b}{}_i^b \end{array}, \qquad i = 1, \dots, h. \tag{D.17}$$

On the contrary, since $(\mathbf{r}^a, \mathbf{p_b}^a)$ and $(\mathbf{r}^b, \mathbf{p_b}^b)$ are both approximate solutions, $r_1^a = r_1^b = \overline{\varepsilon}_1$, which contradicts (D.17). Therefore, there is at most one unique solution to the system of equations.

To identify the unique solution, we construct a sequence of tuples $\{(\mathbf{r}(l), \mathbf{p_b}(l))\}_{l \in \mathbb{N}}$ in the following manner.

1. Set $\mathbf{p_b}(1) = (0, \dots, 0)$ and $i = 1$.

2. Compute $\mathbf{r}(i)$ using (11.25) with $\mathbf{p_b}(i)$ and $r_1(i) = \overline{\varepsilon}_1$.

3. Compute $\mathbf{p_b}(i+1)$ using (11.24) with the computed $\mathbf{r}(i)$ and $\mathbf{p_b}(i)$.

4. Increment i by 1 and perform 2.

By the monotonic property of the non-linear system of equations, the following results can be established.

$$\begin{array}{lcl} r_i(l) & < & r_i(l+1) \\ p_{bi}(l) & < & p_{bi}(l+1) \end{array}, \qquad l \in \mathbb{N}. \tag{D.18}$$

However, each component of $\mathbf{r}$ and $\mathbf{p_b}$ is individually bounded by 1[1]. Therefore, the sequence of numbers for each component of these vectors must converge. Denote the component-wise limit as $(\mathbf{r}^*, \mathbf{p_b}^*)$. Denote $\Xi : [0,1]^h \times [0,1]^h \longrightarrow [0,1]^h \times [0,1]^h$ to be the following map. For each $\mathbf{r}, \mathbf{p_b} \in [0,1]^h$, denote $\Xi(\mathbf{r}, \mathbf{p_b})$ to be the pair, whose first component is the vector of rates computed from (11.25) and the second component is the vector of blocking probabilities computed from (11.24). Then, it can be seen that Ξ is a continuous map and that $\Xi((\mathbf{r}(l), \mathbf{p_b}(l))) = (\mathbf{r}(l+1), \mathbf{p_b}(l+1))$ for each $l \in \mathbb{N}$. Also, for

[1] In fact, it is possible to bound both the rates and the blocking probabilities uniformly away from unity.

this sequence of rates and blocking probabilities, we note that

$$\begin{aligned}\lim_{l\to\infty} \|\Xi((\mathbf{r}^*, \mathbf{p_b}^*)) - (\mathbf{r}^*, \mathbf{p_b}^*)\|_\infty \leq & \lim_{l\to\infty} \|\Xi(\mathbf{r}^*, \mathbf{p_b}^*) - \Xi((\mathbf{r}(l), \mathbf{p_b}(l)))\|_\infty \\ & + \lim_{l\to\infty} \|\Xi((\mathbf{r}(l), \mathbf{p_b}(l))) - (\mathbf{r}(l), \mathbf{p_b}(l))\|_\infty \,\cdot \\ & + \lim_{l\to\infty} \|(\mathbf{r}(l), \mathbf{p_b}(l)) - (\mathbf{r}^*, \mathbf{p_b}^*)\|_\infty \end{aligned} \tag{D.19}$$

However, the three limits on the right-hand side of (D.19) are zero because of the continuity of Ξ and the convergence of $\{(\mathbf{r}(l), \mathbf{p_b}(l))\}_{l\in\mathbb{N}}$ to $(\mathbf{r}^*, \mathbf{p_b}^*)$. Hence, we see that $(\mathbf{r}^*, \mathbf{p_b}^*)$ is a fixed point of the map and hence the unique solution of the system of non-linear equations. Finally, it is trivial to verify that the unique solution satisfies the information flow conservation principle.

$$r_h^* = r_i^*(1 - p_{b_i}^*) = r_j^*(1 - p_{b_j}^*) \qquad 1 \leq i < j < h. \tag{D.20}$$

D.4 Proof of Theorem 11.2.3

The proof is based on mathematical induction on the time index l. At each time, we compare the state of the EMC with that of the modified AMC. Let the extended state of the EMC at an instant $l \in \mathbb{Z}_{\geq 0}$ be denoted by $\mathbf{n}^e(l) = (n_1(l), \ldots, n_h(l))$, where all notations are identical to that in Section 11.2 with the addition that $n_h(l)$ being the number of packets that the destination has received by the l^{th} epoch. Similarly define the state of the AMC with modified buffer sizes at an instant $l \in \mathbb{Z}_{\geq 0}$ by $\mathbf{q}^e(l)$. Define an ordering of vectors of $\mathbb{Z}^h_{\geq 0}$ in the following manner. For two vectors $\mathbf{v}, \mathbf{v'} \in \mathbb{Z}^h_{\geq 0}$, $\mathbf{v} \succeq \mathbf{v'}$ if $\sum_{k=i}^h v_i \geq \sum_{k=i}^h v_i'$ for each $i = 1, \ldots, h$. We track the system starting from initial rest (all buffers being empty) using an instance of channel realizations. Clearly $\mathbf{q}^e(0) \succeq \mathbf{n}^e(0)$.

Suppose that $\mathbf{q}^e(l) \succeq \mathbf{n}^e(l)$ for $l = 0, \ldots k-1$. Consider $l = k$. One of the following two situations may arise.

1. $\underline{\{i < h : q_i^e(k-1) = \sum_{j=1}^i m_i\} = \emptyset}$: Then, we see that no node is saturated in the AMC and hence can potentially accept packets provided both the node preceding it has packets to send and the channel allows it. Consider the number of packets that are in the buffers of nodes $v_j, \ldots, v_h$ for some $0 < j \leq h$ in both the Markov chains.

If $n_{j-1}(k-1) = 0$ of if both $n_{j-1}(k-1) > 0$ and $X_j(k) = 0$ are true, then we see that

$$\sum_{s=j}^{h} q_s^e(k) \geq \sum_{s=j}^{h} q_s^e(k-1) \geq \sum_{s=j}^{h} n_s^e(k-1) = \sum_{s=j}^{h} n_s^e(k). \tag{D.21}$$

However, if $n_{j-1}(k-1) > 0$ and $X_j(k) = 1$ and additionally $q_{j-1}(k-1) = 0$ then

$$\sum_{s=j}^{h} q_s^e(k-1) > \sum_{s=j}^{h} n_s^e(k-1). \tag{D.22}$$

Therefore,

$$\sum_{s=j}^{h} q_s^e(k) \geq \sum_{s=j}^{h} q_s^e(k-1) \geq \sum_{s=j}^{h} n_s^e(k-1) + 1 \geq \sum_{s=j}^{h} n_s^e(k). \tag{D.23}$$

On the other hand, if $n_{j-1}(k-1) > 0$ and $X_j(k) = 1$ and additionally $q_{j-1}(k-1) > 0$ then

$$\sum_{s=j}^{h} q_s^e(k) = \sum_{s=j}^{h} q_s^e(k-1) + 1 \geq \sum_{s=j}^{h} n_s^e(k-1) + 1 \geq \sum_{s=j}^{h} n_s^e(k). \tag{D.24}$$

Since j was arbitrary, it follows that $\mathbf{q}^e(k) \succeq \mathbf{n}^e(k)$.

2. $\underline{\{i < h : q_i^e(k-1) = \sum_{j=1}^{i} m_i\} \neq \emptyset}$: Then, let $I = \max\{i < h : q_i^e(k-1) = \sum_{j=1}^{i} m_i\}$. In this case, nodes $n_{I+1}, \ldots, v_h$ are not saturated and can accept packets. The argument for $\sum_{s=j}^{h} q_s^e(k) \geq \sum_{s=j}^{h} n_s^e(k)$ follows for $j = I+2, \ldots, h$ identically as in the previous case. However, for $j = I+1$, two situations arise depending on the channel realizations at the k^{th} epoch. If $X_{I+1}(k) = 1$, then

$$\sum_{s=I+1}^{h} q_s^e(k) = \sum_{s=j}^{h} q_s^e(k-1) + 1 \geq \sum_{s=j}^{h} n_s^e(k-1) + 1 \geq \sum_{s=j}^{h} n_s^e(k). \tag{D.25}$$

However, if $X_{I+1}(k) = 0$, then

$$\sum_{s=I+1}^{h} q_s^e(k) = \sum_{s=j}^{h} q_s^e(k-1) \geq \sum_{s=j}^{h} n_s^e(k-1) = \sum_{s=j}^{h} n_s^e(k). \tag{D.26}$$

Therefore, the claim is true for $j = I, \ldots, h$. Now for $1 \leq j < I$, we have

$$\sum_{s=j}^{h} q_s^e(k) \geq \sum_{s=I+1}^{h} q_s^e(k) + \sum_{\iota=1}^{I} m_\iota \geq \sum_{s=I+1}^{h} n_s^e(k) + \sum_{\iota=1}^{I} n_\iota^e(k) \geq \sum_{s=j}^{h} n_s^e(k) \tag{D.27}$$

Thus, the claim is true for all indices $j = 1, \ldots, h$ and $\mathbf{q}^e(k) \succeq \mathbf{n}^e(k)$. Here, it must be mentioned that if the buffer sizes for the nodes of AMC are not modified as in the hypothesis, (D.27) will not hold.

Finally, by mathematical induction, the claim follows for all times and we see that

$$\overline{\mathcal{C}}(\mathcal{E}, \mathcal{M}; \infty) = \lim_{l \to \infty} \frac{q_h^e(l)}{l} \geq \lim_{l \to \infty} \frac{n_h^e(l)}{l} = \mathcal{C}(\mathcal{E}, \mathcal{M}; \infty). \tag{D.28}$$

APPENDIX E

ON A CLASS OF DISCRETE-TIME G/GEO/1/K QUEUES

The main aim of this section is to understand the behavior of the Approximated Markov Chain via a formal setup for the discrete-time equivalent of the $G/M/1/k$ queue. More specifically, we interested in the following problem.

Given a single-node server with m customer slots with the service time distribution $\mathbb{G}(\mathbb{t}_N)$, and an arrival process whose inter-arrival times follow a generic distribution that is a weighted sum[1] *of distinct geometric distributions given by $\sum_{i=1}^{N-1} p_i \mathbb{G}(\mathbb{t}_i)$, what is the distribution of the inter-departure times ?*

To illustrate this complications in this problem, Figure E.1 presents an instance of an inter-arrival period for this queue. The number of customers in the queue of the server just before an arrival or a departure is presented on the axis. The arrival and departure of customers is marked by incoming and outgoing arrows, respectively. In Scenario A, we see that the queue is never starved and as a result all the inter-departure times are instances of the service process.

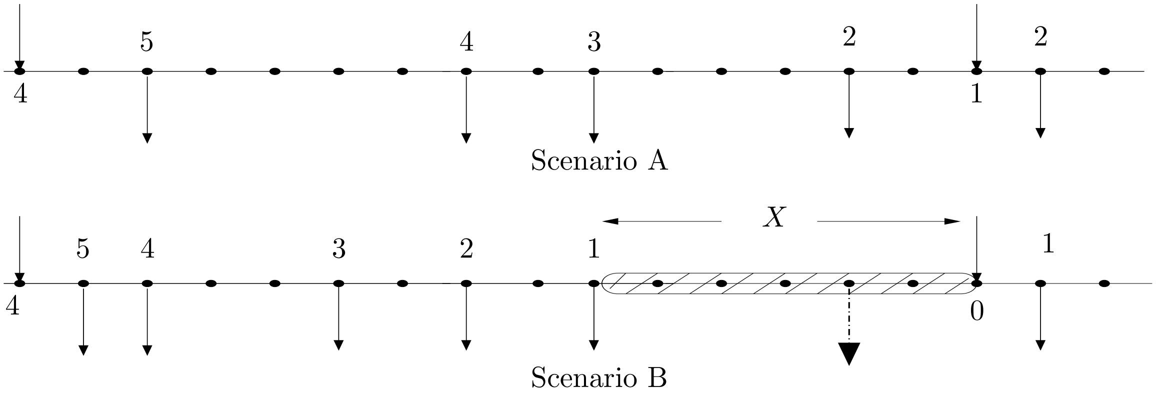

Figure E.1: An instance of the inter-arrival duration at a server with five customer slots.

However, in Scenario B, we notice that all the five customers that are in the queue after

[1] *Note that we do not require that the weights be positive. We only need that their sum be unity and that they generate a valid probability distribution.*

the arrival are serviced much ahead of the next arrival and hence there is a period of time that the queue is starved. If the queue were not starved, it could have possibly serviced a customer at the instance marked by the outgoing dotted arrow. Hence, this duration of time denoted by X in the figure, adds a delay to the inter-duration time. Thus, if we are able to extract the distribution $\{f_i^X\}_{i\in\mathbb{N}}$ of this duration, we can expect the inter-departure time to be a weighted sum of $f_X \otimes \mathbb{G}(\mathbb{t}_N)$ and $\mathbb{G}(\mathbb{t}_N)$.

In order to identify the distribution f_X, we need to identify the probability distribution π of the number of customers that is seen by an arriving customer. The first step in identifying π from the imbedded Markov chain is to construct the distribution $\{D_j\}_{j\in\mathbb{Z}_{\geq 0}}$ of the number of packets that could potentially transmitted during an inter-arrival duration T_A provided the queue is infinite. Note that Scenario A and B are instances where, potentially, the server could service 4 and 6 customers, respectively. This distribution can be computed from the arrival and departure processes in the following manner.

$$\begin{aligned} D_j &= \sum_{k=1}^{\infty} \Pr[T_A = k]\binom{k}{j}\mathbb{t}_N^{k-j}\overline{\mathbb{t}}_N^{j} && \text{(E.1)}\\ &= \sum_{k=1}^{\infty}\Big(\sum_{l=1}^{N-1} p_l\overline{\mathbb{t}}_l\mathbb{t}_l^{k-1}\Big)\binom{k}{j}\mathbb{t}_N^{k-j}\overline{\mathbb{t}}_N^{j} && \text{(E.2)}\\ &= \sum_{l=1}^{N-1} p_l\frac{\overline{\mathbb{t}}_l}{\mathbb{t}_l}\left[\frac{\overline{\mathbb{t}}_N}{\mathbb{t}_N}\right]^j\Big(\sum_{k=1}^{\infty}\binom{k}{j}(\mathbb{t}_l\mathbb{t}_N)^k\Big) && \text{(E.3)}\\ &= \left[\frac{\overline{\mathbb{t}}_N}{\mathbb{t}_N}\right]^j\sum_{l=1}^{N-1}\frac{p_l\overline{\mathbb{t}}_l\eta(\mathbb{t}_l\mathbb{t}_N, j)}{\mathbb{t}_l}, && \text{(E.4)}\end{aligned}$$

where $\eta(y,j) \triangleq \frac{y^j}{(1-y)^{j+1}}(y\sigma(1-j)+\sigma(j))$. For each $i,j\in\{1,\ldots,m\}$, the i,j^{th} entry of the probability transition matrix P_π for the imbedded Markov chain can then be computed by

$$(P_\pi)_{i,j} = \sigma(2-j)\Big(\sum_{k=\min(i,m)}^{\infty} D_k\Big) + \sigma(j-1)D_{j-1-\min(i,m)} \qquad \text{(E.5)}$$

To make (E.5) accurate, we set $D_{j-1-\min(i,m)} = 0$ when $j-1-\min(i,m) < 0$. The distribution π can then be solved from the eigenvector relation

$$\pi(I-P_\pi) = \mathbf{0} \qquad \text{(E.6)}$$

Once we identify π from (E.6), we can identify the distribution of X by conditioning on the number of customers just after a customer arrival. Let M denote the number of customers just after an arrival. Then,

$$\begin{aligned}
\Pr[X = i|M = k] &= \sum_{j=1}^{\infty} \Pr[\text{the queue is emptied at time } j] \Pr[T_A = i + j] && \text{(E.7)}\\
&= \sum_{j=1}^{\infty} \binom{j-1}{k-1} \mathbb{t}_N^k \overline{\mathbb{t}}_N^{j-k} \Big(\sum_{l=1}^{N-1} p_i \overline{\mathbb{t}}_l \mathbb{t}_l^{i+j-1} \Big) && \text{(E.8)}\\
&= \sum_{l=1}^{N-1} p_l [\overline{\mathbb{t}}_l \mathbb{t}_l^{i-1}] \Big(\frac{\overline{\mathbb{t}}_N^k}{\mathbb{t}_N^k} \sum_{j=1}^{\infty} \binom{j-1}{k-1} (\mathbb{t}_l \overline{\mathbb{t}}_N)^j \Big) && \text{(E.9)}\\
&= \sum_{l=1}^{N-1} \Big(p_l \frac{(\mathbb{t}_l \overline{\mathbb{t}}_N)^k}{(1 - \mathbb{t}_l \mathbb{t}_N)^k} \Big) (\overline{\mathbb{t}}_l \mathbb{t}_l^{i-1}). && \text{(E.10)}
\end{aligned}$$

From (E.10), we notice that the distribution of X conditioned on $M = k$ is a weighted sum of geometric distributions. The distribution of X can then be computed as follows.

$$\begin{aligned}
f_i^X &= \frac{\sum_{k=0}^{m} \pi_k \Pr[X = i|M = k]}{\sum_{k=0}^{m} \pi_k \Pr[X \geq 1|M = k]} && \text{(E.11)}\\
&= \Bigg(\sum_{\substack{k \in \{0,\ldots,m\}\\ l = \{1,\ldots,N-1\}}} \Big(\pi_k p_l \frac{(\mathbb{t}_l \overline{\mathbb{t}}_N)^k}{(1 - \mathbb{t}_l \mathbb{t}_N)^k} \Big) \Bigg)^{-1} \sum_{\substack{k \in \{0,\ldots,m\}\\ l = \{1,\ldots,N-1\}}} \Big(\pi_k p_l \frac{(\mathbb{t}_l \overline{\mathbb{t}}_N)^k}{(1 - \mathbb{t}_l \mathbb{t}_N)^k} \Big) (\overline{\mathbb{t}}_l \mathbb{t}_l^{i-1}) && \text{(E.12)}
\end{aligned}$$

Also, we notice that the inter-duration period is either an instance of $f^X \otimes \mathbb{G}(\mathbb{t}_N)$ or that of $\mathbb{G}(\mathbb{t}_N)$, and hence can be written as a $\alpha f^X \otimes \mathbb{G}(\mathbb{t}_N) + (1 - \alpha)\mathbb{G}(\mathbb{t}_N)$ for some $\alpha \in [0, 1]$. The last step in constructing the inter-departure distribution is to identify α. This is done by noticing the mean duration between departures. The expected number of departures in an inter-arrival duration is given by

$$\mathcal{D} = \sum_{i=0}^{m} \pi_i \Big(\sum_{k=0}^{\infty} \min(k, i + 1, m) D_k \Big). \tag{E.13}$$

Therefore, one can identify α by comparing the average rate of departure in the following manner.

$$\frac{1}{\mathcal{D}} \sum_{l=1}^{N-1} \frac{p_l}{1 - \mathbb{t}_l} = \alpha f^X + \frac{1}{1 - \mathbb{t}_N}. \tag{E.14}$$

Finally, we conclude this section by noting that if $\mathbb{t}_i \neq \mathbb{t}_j$ for $1 \leq i < j \leq N$, the distribution of inter-departure times is also a weighted sum of distinct geometric distributions due to the following identity.

$$\mu \neq \lambda \Rightarrow \mathbb{G}(\lambda) \otimes \mathbb{G}(\mu) = \frac{1 - \lambda}{\mu - \lambda} \mathbb{G}(\lambda) + \frac{1 - \mu}{\lambda - \mu} \mathbb{G}(\mu) \qquad (\mu \neq \lambda). \tag{E.15}$$

REFERENCES

[1] Al-Hanbali, A., Kherani, A. A., and Nain, P., "Simple models for performance evaluation of a class of two-hop relay protocols," in *IFIP Networking 2007*, pp. 235–244, 14-18 May 2007.

[2] Alon, N. and Spencer, J., *The Probabilistic Method.* John Wiley, 1992.

[3] Altiok, T., "Approximate analysis of exponential tandem queues with blocking," *European Journal of Operational Research*, vol. 11, no. 4, pp. 390–398, 1982.

[4] Altiok, T., "Approximate analysis of queues in series with phase-type service times and blocking," *Operations Research*, vol. 37, pp. 301–310, July 1989.

[5] Appel, M. J. B. and Russo, R. P., "The connectivity of a graph on uniform points in $[0, 1]^d$," *Statistics and Probability Letters*, no. 60, pp. 351–357, 2002.

[6] Appenzeller, G., Keslassy, I., and McKeown, N., "Sizing router buffers," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 281–292, 2004.

[7] Arfken, G. B. and Weber, H. J., *Mathematical Methods for Physicists.* Elsevier Science and Technology Books, sixth ed., 2005.

[8] Arumugam, R., Subramanian, V., and Minai, A. A., "Intelligent broadcast for large-scale sensor networks," *in Proc. the 4th International Conference on Complex Systems*, 2002.

[9] Babich, F., Montorsi, G., and Vatta, F., "Design of rate-compatible punctured turbo (RCPT) codes," in Proc. of IEEE International Conference on Communications, ICC 2002, Vol. 3, May 2002, pp. 1701–1705.

[10] Békéssy, A., Békéssy, P., and Kolmós, J., "Asymptotic enumeration of regular matrices," *Studia scientiarum mathematicarum Hungarica*, vol. 7, pp. 343–353, 1972.

[11] Berman, K. A. and Konsowa, M. H., "Random paths and cuts, electrical networks, and reversible markov chains," *SIAM J. Discret. Math.*, vol. 3, no. 3, pp. 311–319, 1990.

[12] Berrou, C., Glavieux, A., and Thitimajshima, P., "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *IEEE International Conference on Communications (ICC 93)*, vol. 2, pp. 1064–1070, May 1993.

[13] Bettstetter, C., Hartenstein, H., and Pérez-Costa, X., "Stochastic properties of the random waypoint mobility model," *Wireless Networks*, vol. 10, no. 5, pp. 555–567, 2004.

[14] Bi, D. and Pérez, L. C., "Rate-compatible low-density parity-check codes with rate-compatible degree profiles," *IEE Electronic Letters*, vol. 42, pp. 41–43, Jan. 2006.

[15] Bollobás, B., *Modern Graph Theory.* Springer-Verlag, first ed., 2002.

[16] Brandwajn, A. and Jow, Y.-L. L., "An approximation method for tandem queues with blocking," *Operations Research*, vol. 36, no. 1, pp. 73–83, 1988.

[17] Burgess, J. and Levine, B. N., "CRAWDAD data set umass/diesel (v. 2006-01-17)." Downloaded from http://crawdad.cs.dartmouth.edu/umass/diesel, Jan. 2006.

[18] Burshtein, D. and Miller, G., "Asymptotic enumeration methods for analyzing LDPC codes," *IEEE Trans. Inform. Theory*, vol. 50, pp. 1115–1131, June 2004.

[19] Chen, L.-J., Yu, C.-H., Sun, T., Chen, Y.-C., and hua Chu, H., "A hybrid routing approach for opportunistic networks," in *CHANTS '06: Proceedings of the 2006 SIGCOMM workshop on Challenged networks*, (New York, NY, USA), pp. 213–220, ACM Press, 2006.

[20] Cooper, R. B., *Introduction to Queueing Theory (2nd edition).* Elsevier, 1981.

[21] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms.* The MIT Press, 2nd ed., 2001.

[22] Dallery, Y. and Frein, Y., "On decomposition methods for tandem queueing networks with blocking," *Operations Research*, vol. 41, no. 2, pp. 386–399, 1993.

[23] Dana, A. F., Gowaikar, R., and Hassibi, B., "On the capacity region of broadcast over wireless erasure networks," in *Forty-Second Annual Allerton Conference on Communication, Control and Computing*, October 2004.

[24] Dana, A. F., Gowaikar, R., Palanki, R., Hassibi, B., and Effros, M., "Capacity of wireless erasure networks," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 789–804, 2006.

[25] Di, C., Proietti, D., Telatar, I. E., Richardson, T., and Urbanke, R., "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, pp. 1570–1579, June 2002.

[26] Diggavi, S., Grossglauser, M., and Tse, D., "Even one-dimensional mobility increases ad hoc wireless capacity," *Proceedins of the 2002 IEEE International Symposium on Information Theory*, p. 352, June 2002.

[27] E., O. M., "Algebraic construction of sparse matrices with large girth," *IEEE Trans. on Inform. Theory*, vol. 52, pp. 718–727, February 2006.

[28] Elias, P., "Coding for two noisy channels," *Third London Symposium on Information Theory*, pp. 61–74, October 1955.

[29] Feller, W., *An introduction to probability theory and its applications.* John Wiley & Sons, 2nd ed., 1957.

[30] Gallager, R. G., *Low-density parity-check codes.* PhD thesis, Massachussetts institute of technology, 1963.

[31] Garcy, M. R. and Johnson, D. S., *Computers and Tractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Company, New York, 1979.

[32] Grimmett, G. and Stirzaker, D., *Probability and Random Processes.* USA: Oxford University Press, 2nd ed., September 1992.

[33] Groenevelt, R., Nain, P., and Koole, G., “The message delay in mobile ad hoc networks,” *Perform. Eval.*, vol. 62, no. 1-4, pp. 210–228, 2005.

[34] Grossglauser, M. and Tse, D. N. C., “Mobility increases the capacity of ad hoc wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 10, no. 4, pp. 477–486, 2002.

[35] Gupta, P. and Kumar, P., “Critical power for asymptotic connectivity in wireless networks,” *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W.M. McEneaney, G. Yin and Q. Zhang (Eds.)*, 1998.

[36] Ha, J., Kim, J., Klinc, D., and McLaughlin, S. W., “Rate-compatible punctured low-density parity-check codes with short block lengths,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 728–738, February 2006.

[37] Ha, J., Kim, J., and McLaughlin, S. W., “Puncturing for finite-length low-density parity-check codes,” in Proc. of Intl. Symp. Inform. Theory 2004, Chicago, June-July 2004, pp. 152.

[38] Ha, J., Kim, J., and McLaughlin, S. W., “Rate-compatible puncturing of low-density parity-check codes,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 2824–2836, Nov. 2004.

[39] Ha, J. and McLaughlin, S. W., “Optimal puncturing distributions for rate-compatible low-density parity-check codes,” in Proc. of 2003 IEEE International Symposium on Information Theory (ISIT), Yokohama, Japan, 2003, pp. 233.

[40] Ha, J. and McLaughlin, S. W., “Optimal puncturing of irregular low-density parity-check codes,” in Proc. of IEEE International Conference on Communications (ICC), Anchorage, Alaska, May 2003, pp. 3110–3114.

[41] Ha, J., Kim, J., Klinc, D., and McLaughlin, S. W., “Rate-compatible punctured low-density parity-check codes with short block lengths,” *IEEE Trans. Inform. Theory*, vol. 52, pp. 728–738, Feb. 2006.

[42] Haas, Z. J., Halpern, J. Y., and Li, L., “Gossip-based ad hoc routing,” in *The 21st Annual Joint Conference of IEEE Computer and Communications Societies (IEEE INFOCOM 2002)*, pp. 1707–1716, June 2002.

[43] Hagenauer, J., “Rate-compatible punctured convolutional codes (rcpc codes) and their applications,” *IEEE Trans. on Communications*, vol. 36, pp. 389–399, April 1988.

[44] Hamming, R. W., “Error detecting and error correcting codes,” *The Bell Systems Technical Journal*, vol. 26, pp. 147–161, April 1950.

[45] Hanbali, A. A., Nain, P., and Altman, E., “Performance of ad hoc networks with two-hop relay routing and limited packet lifetime,” in *valuetools ’06: Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, (New York, NY, USA), p. 49, ACM, 2006.

[46] Heinzelman, W. R., Kulik, J., and Balakrishnan, H., "Adaptive protocols for information dissemination in wireless sensor networks," *Proceedings of ACM International Conference on Mobile Computing and Networking (MOBICOM 99).*, (1999).

[47] Ho, T., Médard, M., Koetter, R., Karger, D., Effros, M., Shi, J., and Leong, B., "A random linear network coding approach to multicast," *IEEE Trans. on Inform. Theory*, vol. 52, pp. 4413–4430, October 2006.

[48] Horn, R. A., *Topics in matrix analysis.* New York, NY, USA: Cambridge University Press, 1986.

[49] Hu, X.-Y., Eleftheriou, E., and Arnold, D.-M., "Regular and irregular progressive edge-growth tanner graphs.," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.

[50] Jain, S., Demmer, M. J., Patra, R., and Fall, K. R., "Using redundancy to cope with failures in a delay tolerant network," in *SIGCOMM*, pp. 109–120, 2005.

[51] Jain, S., Fall, K. R., and Patra, R., "Routing in a delay tolerant network," in *Proceedings of the ACM SIGCOMM 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 2004, Portland, Oregon, USA*, pp. 145–158, 2004.

[52] Jindal, A. and Psounis, K., "Performance analysis of epidemic routing under contention," in *Proceedings of IWCMC'06*, 2006.

[53] Jun, H., Ammar, M., and Zegura, E., "Power management in delay tolerant networks: A framework and knowledge-based mechanisms," in *2nd IEEE Conference on Sensor and Ad Hoc Communication and Networks*, Sept 2005.

[54] Kemeny, J. G., S. J. L., *Finite Markov Chains.* New York, USA: Springer-Verlag, 2nd ed., 1976.

[55] Kim, J.-S., Scott, D. J., and Yasinsac, A., "Probabilistic broadcasting based on coverage area and neighbor confirmation in mobile ad hoc networks," *Proceedings of the 2004 IEEE Global Telecommunications Conference Workshops*, pp. 96–101, Nov-Dec 2004.

[56] Koetter, R. and Médard, M., "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, 2003.

[57] Kuai, H., Alajaji, F., and Takahara, G., "A lower bound on the probability of a finite union of events," *Discrete Math.*, vol. 215, no. 1-3, pp. 147–158, 2000.

[58] Lan, L., Tai, Y., Chen, L., Lin, S., and Abdel-Ghaffar, K., "A trellis-based method for removing cycles from bipartite graphs and construction of low density parity check codes," *IEEE Communication Letters*, vol. 8, pp. 443–445, July 2004.

[59] Lavatelli, L., "The Resistive Net and Finite-Difference Equations," *American Journal of Physics*, vol. 40, pp. 1246–1257, Sept. 1972.

[60] Li, J. and Narayanan, K., "Rate-compatible low density parity check codes for capacity-approaching ARQ schemes in packet data communications," in the Proc. of International Conference on Communications, Internet and Information Technology (CIIT), US Virgin Islands, pp. 201–206, Nov. 2002.

[61] Li, L., Ramjee, R., Buddhikot, M., Miller, S., and Hill, M., "Network coding-based broadcast in mobile ad hoc networks," *Proceedings of the 26th IEEE International Conference on Computer Communications (IEEE INFOCOM 2007)*, pp. 1739–1747, May 2007.

[62] Li, S.-Y. R., Yeung, R. W., and Cai, N., "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, pp. 371–381, February 2003.

[63] Lim, H. and Kim, C., "Multicast tree construction and flooding in wireless ad hoc networks," in *The 3rd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 61–68, August 2000.

[64] Lindgren, A., Doria, A., and Scheln, O., "Probabilistic routing in intermittently connected networks," in *The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003)*, June 2003.

[65] Luby, M., Mitzenmacher, M., Shokrollahi, M., and Spielman, D., "Efficient erasure correcting codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 569–584, February 2001.

[66] Luby, M., "LT codes," in *43rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–282, 2002.

[67] Luby, M. G., Mitzenmacher, M., and Shokrollahi, M. A., "Analysis of random processes via and-or tree evaluation," in *SODA '98: Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, (Philadelphia, PA, USA), pp. 364–373, Society for Industrial and Applied Mathematics, 1998.

[68] Lun, D. S., Médard, M., and Effros, M., "On coding for reliable communication over packet networks," *in Proc. Allerton conference on communication, control, and computing*, 2004.

[69] Lun, D. S., Médard, M., and Koetter, R., "Efficient operation of wireless packets networks using network coding," *International Workshop on Convergent Technology*, 2005.

[70] Lun, D. S., Pakzad, P., Fragouli, C., Médard, M., and Koetter, R., "An analysis of finite-memory random linear coding on packet streams," *in Proc. of the 2nd Workshop on Network Coding, Theory, and Applications (NetCod 2006), Boston, MA, April 3-7, 2006.*

[71] Lun, D. S., Ratnakar, N., Koetter, R., Médard, M., Ahmed, E., and Lee, H., "Achieving minimum-cost multicast: A decentralized approach based on network coding," *in Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM*, vol. 3, Mar. 2005.

[72] MacWilliams, F. J. and Sloane, N. J., *The theory of error correcting codes.* North Holland Mathematical Library, 3rd ed., 1988.

[73] Maymounkov, P., "Online codes," *NYU Technical Report TR2003-883*, 2002.

[74] Maymounkov, P. and Mazieres, D., "Rateless codes and big downloads," *in Proc. 2nd International Workshop on Peer-toPeer Systems*, 2003.

[75] McKay, B. D., "Asymptotics for 0-1 matrices with prescribed line sums," in Enumeration and Design (Academic Press, 1984), pp. 225-238.

[76] McKay, B. D., "Asymptotics for symmetric 0-1 matrices with prescribed row sums," *Ars Combinatoria*, vol. 19A, pp. 15–26, 1985.

[77] Melamed, B. and Yao, D., *Advances in Queueing: Theory, Methods and Open Problems*, ch. The ASTA Property in Queueing, invited chapter, pp. 195–224. CRC Press, 1995.

[78] Merugu, S., Ammar, M., and Zegura, E., "Routing in space and time in networks with predictable mobility." Technical Report GIT-CC-04-7, Georgia Institute of Technology, 2004.

[79] Ni, S.-Y., Tseng, Y.-C., Chen, Y.-S., and Sheu, J.-P., "The broadcast storm problem in a mobile ad hoc network," in *5th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking*, pp. 151–162, 1999.

[80] Norris, J. R., *Markov Chains.* Cambridge University Press, 1998.

[81] Obraczka, K., Viswanath, K., and Tsudik, G., "Flooding for reliable multicast in multi-hop ad hoc networks.," *Wireless Networks*, vol. 7, no. 6, pp. 627–634, 2001.

[82] Orecchia, L., Panconesi, A., Petrioli, C., and Vitaletti, A., "Localized techniques for broadcasting in wireless sensor networks.," in *DIALM-POMC*, pp. 41–51, 2004.

[83] Orlitsky, A., Viswanathan, K., and Zhang, J., "Stopping set distribution of LDPC code ensembles," *IEEE Trans. Inform. Theory*, vol. 51, pp. 929–953, March 2005.

[84] Oswald, P. and Shokrollahi, M. A., "Capacity-achieving sequences for the erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, pp. 3017–3028, December 2002.

[85] Pakzad, P., Fragouli, C., and Shokrollahi, A., "Coding schemes for line networks," *IEEE International Symposium on Inform. Theory*, Sep. 2005.

[86] Papoulis, A. and Pillai, S. U., *Probability, Random Variables and Stochastic Processes.* 2001.

[87] Park, S.-J., Vedantham, R., Sivakumar, R., and Akyildiz, I. F., "A scalable approach for reliable downstream data delivery in wireless sensor networks," *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, May 2004.

[88] Penrose, M., *Random Geometric Graphs.* Oxford University Press, 2003.

[89] Penrose, M. D., "A strong law for the longest edge of the minimal spanning tree," *The Annals of Probability*, vol. 1, pp. 246–260, 1999.

[90] Peterson, W. W. and Brown, D. T., “Cyclic codes for error detection,” *Proceedings of the IRE*, vol. 49, no. 1, pp. 228–235, 1961.

[91] Pishro-Nik, H. and Fekri, F., “Results on punctured low-density parity-check codes and improved iterative decoding techniques,” in Proc. of 2004 IEEE Information Theory Workshop, San Antonio, TX, October 2004, pp. 24–29.

[92] Pishro-Nik, H. and Fekri, F., “Improved decoding algorithms for low-density parity-check codes,” *3rd International Symposium on Turbo Codes and Related Topics, Brest, France*, September 2003.

[93] Pishro-Nik, H. and Fekri, F., “On decoding of low-density parity-check codes over the binary erasure channel,” *IEEE Trans. Inform. Theory*, vol. 50, pp. 439–454, March 2004.

[94] Pishro-Nik, H. and Fekri, F., “Results on punctured low-density parity-check codes and improved iterative decoding technicques,” *IEEE Trans. Inform. Theory*, vol. 53, pp. 599–614, February 2007.

[95] Prékopa, A. and Gao, L., “Bounding the probability of the union of events by aggregation and disaggregation in linear programs,” *Discrete Appl. Math.*, vol. 145, no. 3, pp. 444–454, 2005.

[96] Qayyum, A., Viennot, L., and Laouiti, A., “Multipoint relaying for flooding broadcast message in mobile wireless networks,” in *35th Annual Hawaii Int. Conf. on System Science*, 2002.

[97] Rahnavard, N. and Fekri, F., “CRBcast: A collaborative rateless scheme for reliable and energy-efficient broadcasting in wireless sensor networks,” *in Proc. 5th ACM/IEEE International Conference on Information Processing in Sensor Networks, Nashville, TN*, pp. 276–283, Apr. 2006.

[98] Rahnavard, N., Vellambi, B. N., and Fekri, F., “Collaborative rateless broadcast (CRBcast) in wireless sensor networks,” *accepted to IEEE Trans. on Wireless Communications*, 2008.

[99] Richardson, T. J., Shokrollahi, A., and Urbanke, R. L., “Design of capacity-approaching irregular low-density parity-check codes,,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, February 2001.

[100] Richardson, T. J., Shokrollahi, M. A., and Urbanke, R. L., “Finite-length analysis for various low-density-parity-check ensembles for the binary erasure channel,” in Proc. of 2002 IEEE International Symposium on Information Theory (ISIT), Lausanne, Switzerland, June-July 2002, pp. 1.

[101] Richardson, T. J. and Urbanke, R. L., “The capacity of low-density parity-check codes under message-passing decoding,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 599–618, February 2001.

[102] Richardson, T. and Urbanke, R., “Efficient encoding of low-density parity-check codes,” *IEEE Transactions on Inform. Theory*, vol. 47, pp. 638–656, February 2001.

[103] Scott, D. J. and Yasinsac, A., "Dynamic probabilistic retransmission in ad hoc networks," *Proceedings of the International Conference on Wireless Networks*, June 2004.

[104] Serfozo, R., *Introduction to Stochastic Networks.* New York: Springer-Verlag, 1st ed., 1999.

[105] Shannon, C. E., "A mathematical theory of communication," *The Bell Systems Technical Journal*, vol. 27, pp. 379–423, July 1948.

[106] Shannon, C. E., "A mathematical theory of communication," *The Bell Systems Technical Journal*, vol. 27, pp. 623–656, October 1948.

[107] Shiryaev, A. N., *Probability.* Springer-Verlag, second ed., 1995.

[108] Shokrollahi, A. and Storn, R., "Design of efficient erasure codes with differential evolution," in Proc. of the 2000 IEEE International Symposium on Information Theory ISIT, pp. 5, June 2000.

[109] Shokrollahi, A., "Raptor codes," *IEEE Trans. on Inform. Theory*, vol. 52, pp. 2551–2567, June 2006.

[110] Shokrollahi, M. A., "New sequences of linear time erasure codes approaching the channel capacity," *Proc. of AAECC-13.* Lecture Notes in Compute Science 1719, pp. 65–76, 1999.

[111] Shokrollahi, M. A., "Capacity-achieving sequences," *in IMA Volumes in Mathematics and its Applications*, vol. 123, pp. 153–166, 2000.

[112] So, K. C. and tsai E. Chin, K., "Performance bounds on multiserver exponential tandem queues with finite buffers," *European Journal of Operational Research*, vol. 63, no. 3, pp. 463–477, 1992.

[113] Spyropoulos, T., Psounis, K., and Raghavendra, C., "Single-copy routing in intermittently connected mobile networks," in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON'04)*, pp. 235–244, 4-7 October 2004.

[114] Spyropoulos, T., Psounis, K., and Raghavendra, C., "Efficient routing in intermittently connected mobile networks: The single-copy case," *ACM/IEEE Transactions on Networking*, 2007. to appear.

[115] Stanley, R. P., *Enumerative combinatorics*, vol. 1. Cambridge University Press, 2 ed., 2002.

[116] Stojmenovic, I., Seddigh, M., and Zunic, J., "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Trans. on Parallel and distributed systems*, vol. 13, pp. 14–25, Jan. 2002.

[117] Tanenbaum, A. S., *Computer Networks.* Upper Saddle River, NJ, USA: Prentice-Hall PTR, 4th ed., 2002.

[118] Tariq, M. M. B., Ammar, M. H., and Zegura, E. W., "Message ferry route design for sparse ad hoc networks with mobile nodes," in *MobiHoc*, pp. 37–48, 2006.

[119] Tian, T., Jones, C., Villasenor, J., and Wesel, R., "Selective avoidance of cycles in irregular ldpc code construction," *IEEE Trans. on Communications*, vol. 52, pp. 1242–1247, August 2004.

[120] Vahdat, A. and Becker, D., "Epidemic routing for partially connected ad hoc networks," *Technical Report CS-200006, Duke University*, April 2000.

[121] Vellambi, B. N. and Fekri, F., "On stopping sets and an improved decoding algorithm for low-density parity-check codes over the binary erasure channel," in *Forty-Second Annual Allerton Conference on Communication, Control and Computing*, October 2004.

[122] Wang, Y., Jain, S., Martonosi, M., and Fall, K., "Erasure-coding based routing for opportunistic networks," in *WDTN '05: Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 229–236, ACM Press, 2005.

[123] Widmer, J. and Boudec, J. Y. L., "Network coding for efficient communication in extreme networks," *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 284–291, 2005.

[124] Wieselthier, J. E., Nguyen, G. D., and Ephremides, A., "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks and Applications*, vol. 7, pp. 481–492, 2002.

[125] Wischik, D. and McKeown, N., "Part i: buffer sizes for core routers," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 3, pp. 75–78, 2005.

[126] Yazdani, M. R. and Banihashemi, A. H., "On construction of rate-compatible low-density parity-check codes," *IEEE Comm. Letters*, vol. 8, pp. 159–161, March 2004.

[127] Zhang, Q. and Agrawal, D. P., "Dynamic probabilistic broadcasting in mobile ad hoc networks," *IEEE 58th Vehicular Technology Conference*, vol. 5, pp. 2860–2864, Oct. 2003.

[128] Zhang, X., Neglia, G., Kurose, J., and Towsley, D., "Performance modeling of epidemic routing," *Comput. Netw.*, vol. 51, no. 10, pp. 2867–2891, 2007.

[129] Zhao, W. and Ammar, M. H., "Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks," in *9th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2003)*, pp. 308–314, May 2003.

[130] Zhao, W., Ammar, M. H., and Zegura, E. W., "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *5th ACM Interational Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2004)*, pp. 187–198, May 2004.

VITA

Badri N. Vellambi was born on January 6, 1982 in the small city of Indore in the state of Madhya Pradesh at the heart of India. He spent most of his schooling years in the southern metropolis of Chennai. He completed his junior college diploma from Adarsh Senior Secondary School in 1998 and obtained his undergraduate degree from the prestigious Indian Institute of Technology-Madras at Chennai in 2002. He moved to Atlanta to commence his graduate studies at the Georgia Institute of Technology shortly thereafter. He was awarded the Master of Science and Master of Science in Mathematics degrees in 2005 and 2008, respectively. During his stay in Georgia Tech, he was awarded both the outstanding research award and the outstanding service award from the Center of Signal and Image Processing. He will be working as a research fellow at the Institute of Telecommunications Research, Adelaide starting Spring 2008 (September 2008).

His broad research interests include coding and information theory, and communication theory. Pure mathematics (analysis and algebra) also interest him to a great extent.