Low Complexity Scheduling and Coding for Wireless Networks

THÈSE N^{O} 6512 (2015)

PRÉSENTÉE LE 24 FÉVRIER 2015

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

LABORATOIRE D'ALGORITHMIQUE POUR L'INFORMATION EN RÉSEAUX

PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Siddhartha BRAHMA

acceptée sur proposition du jury:

Prof. B. Faltings, président du jury
Prof. C. Fragouli, directrice de thèse
Prof. O. N. A. Svensson, rapporteur
Prof. D. Tuninetti, rapporteuse
Prof. A. Özgür Aydin, rapporteuse

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

To my father...

...and to all from whom I have learnt.

Abstract

The advent of wireless communication technologies has created a paradigm shift in the accessibility of communication. With it has come an increased demand for throughput, a trend that is likely to increase further in the future. This has given rise to several challenges in developing new communication schemes. A key aspect of these challenges is to develop low complexity algorithms and architectures that can take advantage of the nature of the wireless medium like broadcasting and physical layer cooperation.

In this thesis, we consider several problems in the domain of low complexity coding, relaying and scheduling for wireless networks. We begin with the formulation of the Pliable Index Coding problem that models a server trying to send one or more new messages over a noiseless broadcast channel to a set of clients that already have a subset of messages as side information. We show through theoretical upper bounds and algorithms, that it is possible to design very short length codes, poly-logarithmic in the number of clients, to solve this problem. The length of the codes are exponentially better than those possible in a traditional index coding setup.

We next turn our attention to several aspects of low complexity relaying in half-duplex diamond networks. In such networks, the source transmits information to the destination through n half-duplex intermediate relays arranged in a single layer. The half-duplex nature of the relays implies that they can either be in a listening or transmitting state at any point of time. To achieve high rates, there is an additional complexity of optimizing the schedule (i.e. the relative time fractions) of the relaying states, which can be 2^n in number. Using approximate capacity expressions derived from the quantize-map-forward scheme for physical layer cooperation, we show that for networks with $n \leq 6$ relays, the optimal schedule has atmost $n+1$ active states. This is an exponential improvement over the possible 2^n active states in a schedule. We also show that it is possible to achieve at least half the capacity of such networks (approximately) by employing simple routing strategies that use only two relays and two scheduling states. These results imply that the complexity of relaying in half-duplex diamond networks can be significantly reduced by using fewer scheduling states or fewer relays without adversely affecting throughput.

Both these results assume centralized processing of the channel state information of all the relays. We take the first steps in analyzing the performance of relaying schemes where each relay switches between listening and transmitting states randomly and optimizes their relative fractions using only local channel state information. We show that even with such simple scheduling that avoids centralized communication, we can achieve a

significant fraction of the capacity of the network. Next, we look at the dual problem of selecting the subset of relays of a given size that has the highest capacity for a general layered full-duplex relay network. We formulate this as an optimization problem and derive efficient approximation algorithms to solve them.
We end the thesis with the design, analysis and implementation of a practical relaying scheme called QUILT. As a key component of the scheme, the relay opportunistically decodes or quantizes its received signal and transmits the resulting sequence in cooperation with the source. To keep the complexity of the system low, we use LDPC codes at the source, interleaving at the relays and belief propagation decoding at the destination. We show through over-the-air (WiFi) experiments on Warplab testbeds that the scheme performs better than existing state-of-the-art physical layer cooperation schemes, achieving improved frame error rates by a factor of 5 for some topologies.

Key words: wireless networks, pliable index coding, relay networks, half-duplex relay scheduling, relay selection, physical layer cooperation, quantize-map-forward, belief propagation decoding.

Résumé

L'avènement des technologies de communication sans fil a créé un changement de paradigme dans l'accès à la communication. Avec lui est venu une demande accrue de débit, une tendance qui est susceptible d'augmenter encore à l'avenir. Cela a donné naissance à plusieurs défis dans le développement de nouveaux systèmes de communication. Un aspect clé de ces défis est de développer des algorithmes et des architectures de faible complexité qui peuvent tirer avantage de la nature du support sans fil comme la radiodiffusion et la coopération avec la couche physique.

Dans cette thèse, nous considérons plusieurs problèmes dans le domaine de codage à faible complexité, relais et la planification des réseaux sans fil. Nous commençons avec la formulation du problème de Pliable Index Coding qui modèle un serveur tentant d'envoyer un ou plusieurs nouveaux messages sur un canal de diffusion non-bruité à un ensemble de clients qui ont déjà un sous-ensemble de messages comme information supplémentaire. On montre par le biais de bornes supérieures et d'algorithmes théoriques, qu'il est possible de concevoir des codes de longueurs très courtes, poly-logarithmique dans le nombre de clients, pour résoudre ce problème. La longueur des codes est exponentiellement meilleure que celle possible dans une configuration traditionnelle de codage d'index.

Nous tournons ensuite notre attention sur plusieurs aspects de relayage à faible compléxité dans les réseaux semi-duplex en configuration diamant. Dans ces réseaux, la source émet des informations vers la destination à travers n relais intermédiaires semi-duplex disposés en une seule couche. La nature semi-duplex des relais implique qu'ils ne peuvent être que soit en reception, soit en transmission, mais pas les deux simultanément. Pour atteindre des débits élevés, il y da une complexité supplémentaire pour l'optimisation de la planification (c.à.d. les fractions de temps relatif) des états des relais, au total 2^n combinaisons différentes. Par l'utilisation d'expressions approximatives de la capacité provenant de la technique quantize-map-forward pour la coopération dans la couche physique, nous montrons que pour les réseaux avec $n \leq 6$ relais, la planification optimale a au plus $n+1$ états actifs . Ceci est une amélioration exponentielle du nombre d'états actifs 2^n possibles dans la planification. Nous montrons également qu'il est possible de réaliser au moins la moitié de la capacité de ces réseaux (approximativement) en utilisant des stratégies de routage simples qui utilisent seulement deux relais et les deux états de programmation. Ces résultats impliquent que la complexité de relayer dans les réseaux en diamant semi-duplex peut être considérablement réduite en utilisant moins d'états ou moins de relais sans affecter le débit.

Ces deux résultats supposent un traitement centralisé de l'information de tous les états des relais du canal. Nous faisons les premiers pas dans l'analyse de la performance des régimes où chaque relais commute entre l'écoute et et la transmission au hasard et optimise sa fraction relative en utilisant uniquement les informations d'état local du canal. Nous montrons que, même avec un tel ordonnancement simple qui évite la communication centralisée, on peut atteindre une fraction significative de la capacité du réseau. Ensuite, nous regardons le double problème de la sélection du sous-ensemble de relais d'une taille donnée qui a la plus grande capacité pour un réseau général de relais full-duplex en couches. Nous formulons cela comme un problème d'optimisation et déduisons des algorithmes d'approximation efficaces pour le résoudre.
Nous terminons la thèse avec la conception, l'analyse et la mise en œuvre d'un système de relais pratique appelée QUILT. Comme élément clé de cette technique, le relais décode ou quantifie le message reçu de façon opportuniste, et transmet la séquence résultante en coopération avec la source. Pour garder la complexité du système faible, nous utilisons des codes LDPC à la source, l'entrelacement au décodage relais et du décodage à propagation de croyance (belief propagation) à la destination. Nous montrons par des expériences sur les ondes (WiFi), sur un banc d'essai Warplab, que le système fonctionne mieux que les programmes de coopération de la couche physique de pointe existants avec une amélioration des taux d'erreur de trame jusqu'à un facteur cinq pour certaines topologies.

Mots clés : réseaux sans fil, pliable index coding, les réseaux de relais, planification de relais semi-duplex, sélection de relais, oopération de la couche physique, quantize-map-forward, décodage à propagation de croyance.

Acknowledgements

Successfully completing my PhD is one of the most important milestones in my life. I have been able to accomplish this only because of the support of numerous people, not just during my stay at EPFL but from times much before that. I am grateful to every one of them, including those who do not know me.

Foremost, I am indebted to my advisor, Christina Fragouli. I still remember the day when I wrote an email approaching her for a PhD position. She was kind enough to invite me for an interview and the interaction I had with her during the interview convinced me to come to EPFL. During the PhD, Christina showed an amazing ability to maintain a balance between guiding me towards important research problems and giving me freedom to choose my own problems. She taught me the art of persevering with a problem, trying out different ideas and reasoning out why a certain approach may or may not work. Despite her busy schedule, she always made sure to talk to me every week and guide me whenever I got stuck. I am grateful for her support and encouragement to attend conferences and workshops to showcase my work. She is also a very good teacher and I thoroughly enjoyed doing the Network Coding course under her. In fact, the first chapter of the thesis is based on an open ended problem that she had proposed during the course. I am also very grateful for her constant and generous personal support on numerous occasions - she never let me feel without a guardian.

I had the good fortune of interacting with several other faculty members in the IC department. A significant part of my work was done in collaboration with Suhas Diggavi - whose knowledge of network information theory and guidance really helped me in contributing positively to the work on physical layer cooperation. I am also grateful to Ayfer Özgür, whose initial work on half-duplex diamond networks forms the basis of much of my work on the same topic. As a teaching assistant, I had the chance to work with Rüdiger Urbanke, Olivier Lévêque and Nicolas Macris - a very enjoyable and rewarding experience. My research at EPFL received financial support from the ERC Project NOWIRE-ERC-2009-StG-240317, which I appreciate a lot.

My life at the IC department and EPFL was made very smooth because of two people. First, I would like to express my deepest thanks to Francoise Behn, who single handedly managed all affairs of the ARNI lab. She graciously arranged for all my travel and helped in other administrative and personal issues and made me feel at home. I will miss the neatly prepared envelopes with tickets and hotel reservations that she gave me before every conference. Second, I am grateful to Damir Laurenzi for giving me all the

computational resources required to run numerous simulations. I will cherish his great sense of humor and instant replies to my emails even in the middle of the night.
The members of our ARNI lab have been some of the most friendly and helpful group of people I have come across in my life. I thank Emre for help with the WARP boards, Javad for being the first person to show me around Lausanne and Lorenzo, Iris, Marios and I-Hsiang. Three more people stand out. I would like to thank Shirin for being my office mate for two years and sharing many a discussion on random topics. Thanks also goes to Melissa for being such a great coworker. Without her patience, effort and diligence, our work on physical layer cooperation would not have come to fruition. Most importantly, I would like to thank Ayan for all the support, encouragement and help in the past four years. It was a matter of great comfort that we shared a remarkably similar background and hence could speak about everything under the sun without barriers. There was a free flow of ideas between us and a lot of the work we did together came as a result of this. We complemented each other in research and worked wonderfully as a team, something that I will cherish forever. Outside ARNI, I would like to extend my thanks to Robin for translating my thesis abstract, and to Hamed, Vahid, Saeid, Rammohan, Shovan, Mukul and Mayur for both professional and personal help.
I also take this opportunity to thank my colleagues and professors at Princeton and IIT Kharagpur who contributed to my growth as a researcher, especially Sudebkumar Pal who was my undergraduate advisor and from whom I learnt the first steps of doing research. I also thank Robert Tarjan for being such a great inspiration.
We are all born alone in this world, if not for our families. My greatest debt is to my parents who sacrificed everything for my well-being and always provided the right environment to pursue my academic passions. I have been influenced greatly by my mother's insistence on truth and correctness - two qualities that have always helped me. I am very thankful to my sister for always being caring and supportive through good and bad times. I am also thankful to my in-laws, especially my late father-in-law who always encouraged me to scale newer heights. A very special note of thanks goes to my wife who has been by my side through all the frustration and elation of doing research. She has often turned my pessimism into optimism, encouraged me to strike out new paths, tolerated my idiosyncracies with humor and been a constant companion even when I was away from her.
I end by expressing my deepest gratitude to my father, who left this world less than a year before I started my PhD. A lot of what I am is because of him. From an early age, he was a friend, philosopher and guide to me, silently moulding me to become a good humanist. He made the greatest sacrifices for my well-being and yet magnanimously accepted the choices I made in life. A part of him will forever remain in my existence. This thesis is a homage to my father.

Lausanne, 19 December 2014

Contents

List of Figures

Introduction

The development and widespread use of wireless communication technologies has revolutionised the way we communicate and stay in touch with each other. According to data from the International Telecommunication Union (ITU) [1] , the number of mobile phone subscriptions in the world is expected to reach almost 7 billion by the end of 2014. During the same period, with widespread availability of smartphones, the number of people who access the internet from their mobile devices is expected to exceed 2.3 billion. This has created a huge demand for higher bandwidth and throughputs, which in turn has created new challenges for developing more elaborate communication technologies and architectures that can satisfy this demand.

Most of the wireless communication now takes place through point-to-point links between two communicating nodes. In the past few decades, extensive research in information and coding theory combined with their practical application has provided us with a good understanding of point-to-point communication [2]. However, to achieve further gains we must look towards more complicated networked architectures. Such architectures take advantage of two key aspects of the wireless medium - broadcast and multiple access. In broadcast, a single source can transmit information to multiple receivers over a wireless channel, while in multiple access, two or more nodes can cooperate to send information to a destination node. Together they constitute physical layer cooperation in wireless, which is distinct from the "bit-pipe" view of wired communication or point-to-point wireless communication [3]. The study of network information theory [4] shows that it is possible to attain higher throughputs by using techniques that take advantage of physical layer cooperation and some of these techniques have already started to be included in official standards [5].

A key challenge in making physical layer cooperation more useful is to devise *low complexity* schemes. Depending on the exact scenario or problem, there can be many interpretations of the term "low complexity". It can refer to simpler codes (e.g. in terms of length), lesser operational and computational complexity or using fewer network resources. In this thesis, we look at several different problems in wireless networks with an aim of discovering structures or devising techniques that have reduced complexity in one or more ways.

We start off with a coding problem in noiseless wireless broadcast networks. In the well-known *Index Coding* with side-information problem, a server holds m messages, and can broadcast over a noiseless channel to a set of n receivers or clients. Each client has as side information some subset of the m messages, and requests from the server a *specific* message she does not have. The objective is to devise a coding strategy that minimizes the number of broadcast transmissions the server makes to satisfy the demands of all the clients [6]. The index coding problem has been studied extensively in the literature and has deep connections with the problem of network coding [7]. We formulate a new variant of index coding where the clients are *pliable* and are happy to receive *any* t messages they do not already have. We term the new formulation *Pliable Index Coding (t)* (or **PICOD(t)**). There are several applications that motivate this formulation - specifically scenarios where the clients are interested in receiving with low delay any information that is not a part of their side information sets. The goal is then to compute the shortest code that satisfies all the clients. Although we show that computing the shortest code in PICOD(t) is NP-hard, we derive the surprising result that codes of length $O(\log^2 n)$ are sufficient for any instance of PICOD(1). This is an exponential improvement over index coding, where the length of the code is $\Omega(n)$ in the worst case. We also present general results for any t and also for a scenario where the server has only knowledge about the cardinality of the side information sets of the clients. The theoretical upper bounds are accompanied by simple greedy algorithms that perform very well on random instances of PICOD(t), as shown through extensive simulation results.

A canonical example of physical layer cooperation is a relay network, where a source node sends information to a destination node through one or more intermediate relays. In the next topic, we consider the problem of low complexity relaying in half-duplex diamond networks from several different perspectives. In such networks, the source communicates with the destination using n half-duplex relays arranged in a single layer, with there being no inter-relay communication. Being half-duplex, each relay can either be in the listening (L) or transmitting (T) state at any point of time, giving rise to 2^n relaying states for the whole network. To achieve high rates close to capacity, the schedule of relaying states (i.e. their relative time fractions) needs to be optimized. The complexity of relaying in such networks, among other things, can arise from – (i) the number of active states in the schedule (which can potentially be 2^n), (ii) the number of relays being used and (iii) the use of global channel state information in computing relaying schedules. We investigate each of these sources of complexity in isolation and derive results on how to reduce it. Since the exact capacity of half-duplex diamond networks is not known, we use simple approximations [8, 9, 10] to the capacity that are a constant gap away from the true value, i.e., the gap from capacity is only a function of n and independent of channel strengths. These approximations are based on the quantize-map-forward relaying scheme that utilizes physical layer cooperation. Further, these approximations can be expressed as a linear program, whose coefficients are a function of the individual point to point link capacities in the network.

First, we show the surprising result that for $n \leq 6$, the approximately optimal schedule has atmost $n+1$ active states. This is an exponential improvement over the possible 2^n scheduling states. In fact, we conjecture this to be true for any n. To prove the result we develop a computational proof strategy that crucially uses *submodularity properties* of information flow across cuts in the network and *linear programming duality* to derive contradictions for optimal schedules having more than $n+1$ states. Second, we show that it is possible to achieve high rates even when using only a subset of relays. More concretely, we show that very simple *routing strategies* that use only two relays and two relaying states can achieve rates that are at least half the capacity of the network (approximately). For 2-relay networks, we show that routing strategies achieve at least 8/9 of the capacity (approximately). These can also be seen as network simplification results for half-duplex diamond networks in the same vein as those derived for full-duplex diamond networks by Nazaroglu et. al. [10]. The proof uses linear programming duality to derive upper bounds to the capacity expressions and defines an algorithmic procedure to select a pair of relays with the claimed property in $O(n \log n)$ time. Third, we investigate the performance we can achieve if we restrict ourselves to schedules that can be derived only using local channel state information. We propose a randomized strategy in which each relay switches between the L and T state multiples times, at the same time ensuring that the relative fraction of L and T states obey a local optimality condition. We show that using this approach for the 2-relay diamond network, we can achieve at least 3/4 of the capacity of the network (approximately) as the number of switches increases to infinity. The expected rates achieved by our strategy increases with the number of switches, with most of the gain achieved using only two switches. To summarize, each of the three results show that it is possible to reduce the complexity of relaying in half-duplex diamond networks without adversely affecting the throughput.

We next consider the following problem in full-duplex layered relay networks – how to select the subset of relays of a given size k that has the highest capacity in a computationally efficient manner? Using approximate capacity expressions similar to ones used for half-duplex diamond networks, we formulate this as an integer optimization problem. We then relax it to a real-valued problem and use the real (or fractional) optimum to generate approximate solutions for the original problem. For single layered or diamond networks, using properties of submodular functions and convex optimization, we show that the real-valued optimization problem can be solved in polynomial time. Simulation results show that our algorithm achieves high accuracy rates and is significantly faster than an exhaustive search algorithm. For n-relay full-duplex diamond networks, Nazaroglu et.al. [10] showed that for any $1 \leq k < n$ there exists a subnetwork of relays that achieves at least $k/(k+1)$ fraction of the capacity approximately. Thus our algorithm can be used to compute the best such network efficiently.

The theoretical results on half-duplex diamond networks presented above hint at the possibility of low complexity relaying schemes based on physical layer cooperation that use a few relays and relaying states and yet achieve high throughput. This motivates

us to our final contribution in this thesis. We consider the simplest scenario of a source communicating with a destination with the help of a single half-duplex relay. We propose and evaluate QUILT, a system for practical physical-layer relaying based on the quantize-map-forward scheme. We make several design choices, each of which is justified theoretically, driven by the need to keep the complexity of the system low. In particular, we use LDPC codes at the source, symbol level quantization and interleaving at the relay and a belief propagation algorithm at the destination for fast decoding. The relay works on the principle of opportunistic decoding, i.e., it decodes whenever possible and quantizes otherwise. We implement the system over the Warplab software radio testbed and perform over-the-air (WiFi) experiments to evaluate its performance. We show that our system outperforms existing approaches to physical layer relaying, achieving frame error rates five times lower than the next best scheme in some topologies.

Main Contributions

The main contributions presented in this thesis are as follows.

- We propose and study the Pliable Index Coding problem, a novel variant of the standard Index Coding problem where each client is happy to receive any t messages it does not have in its side information sets. We show that although finding the optimum code is NP-hard, any instance of the problem for $t = 1$ can be solved using codes of length that grows poly-logarithmically in the number of clients, which is an exponential improvement over the worst case length of codes in index coding. The results are generalized to arbitrary t and also for the scenario where the server only knows the cardinality of the side information sets of the clients.

- We show that for half-duplex diamond networks with $n \leq 6$ relays, there exist approximately optimal schedules that have atmost $n + 1$ active states. We design and implement a computational proof strategy using submodularity properties of approximate capacity expressions and linear programming duality to prove this result.

- We show that simple routing strategies employing only two relays and only two relaying states that avoid multiple access and broadcast, can achieve at least half the capacity of half-duplex diamond networks (approximately). We use techniques from linear programming duality to derive this result.

- For half-duplex diamond networks, we show that randomized switching strategies using only local channel state information can achieve a significant fraction of the capacity of the network.

- We derive efficient algorithms for selecting the subnetwork of relays of a given size with the highest capacity for full-duplex diamond networks and also generalize it

to arbitrary layered networks. The algorithms use submodularity properties of approximate capacity expressions and convex programming.

- We design, analyze and implement QUILT - a practical, low complexity implementation of physical relay cooperation for single relay networks. Key features of the system include - LDPC codes at the source, symbol level quantization followed by bit-wise interleaving at the relays and belief propagation based joint decoding at the destination.

Outline

The thesis is divided into six chapters. The contents in each of the chapters is as follows.

- In Chapter 1, we present our results on Pliable Index Coding which include the NP-Hardness proof, upper bounds on the length of codes, algorithms and simulation results on the performance of the algorithms.
- In Chapter 2, we present the computational proof strategy for showing that approximately optimal schedules in half-duplex diamond networks have atmost a linear number of states. The implementation details and numerical results are also presented.
- In Chapter 3, we present our result on the performance of simple routing strategies for half-duplex diamond networks. We first develop the proof technique for the special class of antisymmetric networks and then generalize it to arbitrary ones.
- In Chapter 4, we consider randomized relay schedules using only local channel state information for half-duplex diamond networks. We show theoretical lower bounds and simulation results on their performance.
- In Chapter 5, is devoted to the problem of relay selection in full-duplex layered networks. Efficient approximation algorithms are developed and their performance is evaluated using simulations.
- Finally Chapter 6 contains our results on QUILT – the physical layer cooperation scheme for single relay networks, including its design, implementation details and results obtained from over-the-air experiments on Warplab testbeds.

We conclude with a discussion of open problems and possible directions of future work.

1 Pliable Index Coding

In the well-known *Index Coding* problem, a server holds m messages and can broadcast over a noiseless channel to a set of n receivers or clients. Each client has as side information some subset of the m messages and requests from the server a *specific* message she does not have. The objective is to devise a coding strategy that minimizes the number of broadcast transmissions the server makes to satisfy the demands of all the clients.

In this chapter we formulate the *Pliable Index Coding (t)* (or **PICOD(t)**) problem, where the clients are *pliable* and are happy to receive *any* t messages they do not already have. Although PICOD(t) is more unconstrained than traditional index coding, we show that computing the shortest code remains NP-Hard. When the size of the side information sets of all the clients is $s < m$ and each client wants one new message ($t = 1$), we show that codes of length $O(\log n)$ are sufficient. This is an exponential improvement over index coding, where the length of the code is $\Omega(n)$ in the worst case. In this scenario, for a general value of t, we show that codes of length $O(\min(t \log n, t + \log^2 n))$ are sufficient. That is, if $t \gg \log n$, then the number of broadcast transmissions needed grows *linearly* with t.

We also consider the *Oblivious Pliable Index Coding Problem (t)* (or **OB-PICOD(t)**), where the server only knows the cardinality of the side information sets of the clients and each client would like to know any t new messages it does not have. When these cardinalities are all equal to s, we show that codes of length $\min(s + t, m - s)$ are both sufficient and necessary for linear codes.

All the results for PICOD(t) and OB-PICOD(t) are also generalized to the case when the side information sets are of different cardinalities. As a final contribution, we propose heuristic approximation algorithms for PICOD(t). These are based on a natural bipartite graph representation of the problem and employ coverings of the graph. We show through extensive simulation results that a simple greedy covering algorithm performs very well in practice and that the length of the codes closely follow the behavior predicted by the theoretical upper bounds.

1.1 Related Work

Over the past few years, there has been a significant amount of work on the theory of index coding, especially for linear codes. The problem was introduced by Birk *et. al.* [6] in the context of an application in satellite communication networks. Bar-Yossef *et.al.* [11] presented the first theoretical analysis of the problem. They showed that the optimal length for a *scalar linear* index code is given by a graph functional called the *minrk*. They conjectured this to be true even for non-linear codes, which was later disproved by Lubetzky *et.al.* [12]. New graph parameters were introduced in [13] showing the strict separation of optimal solutions for different field sizes.

Building on the work of [14, 7] which investigate the connections between index coding and network coding and using information theoretic linear programs, Blasiak et.al [15] prove some of the tighest known bounds for the index coding problem. The work of Blasiak et.al. [15, 16] also shows several separation results between the optimal linear and non-linear index codes. These results can also be used to come up with instances in network coding that have large gaps between linear and non-linear coding rates. Techniques from interference alignment have also been used to analyse index codes [17]. There have been other investigations dealing with several aspects of the index coding problem including the complementary index coding problem [18], index codes with near extreme rates [19], secure index coding [20], index codes in presence of error [21] and index coding with outerplanar side information [22].

In terms of research looking at slightly different formulations from the core one, although as far as we know pliable index coding as we define it has not been examined before our work, the following are some representative contributions. The so called bipartite index coding problem is analysed in [23, 24] where multiple clients may "want" a specific message. A special case of this problem where the side information sets are of size one was completely characterized in [25]. Finally, instantly decodable network codes were investigated in [26] where the clients want all the messages that they do not have and want them to be instantly decodable.

1.2 Problem Formulation

Suppose that the server has m messages $b_1, \cdots, b_m$ and there are n clients $c_1, \cdots, c_n$. The messages are assumed to lie in a field $(\mathcal{F}, +, \cdot)$ that is large enough (this will be clarified later) and all the encoding and decoding functions are linear. Each client c_i knows a subset of messages $b_{N_c[i]}$, where $N_c[i]$ is a strict subset of $[m]$. Here $b_{N_c[i]}$ denotes the set $\{b_j,\, j \in N_c[i]\}$ and $[m] = \{1, 2, \cdots, m\}$. Thus $N_c[i]$ represents the indices of the messages that client c_i has as side information. Let t be the number of new messages each client wants to know. We will assume that $|N_c[i]| \leq m - t$ for all $i \in [n]$. The (linear) *Pliable Index Coding Problem* PICOD(t) problem is to devise a linear code $\mathcal{C}$

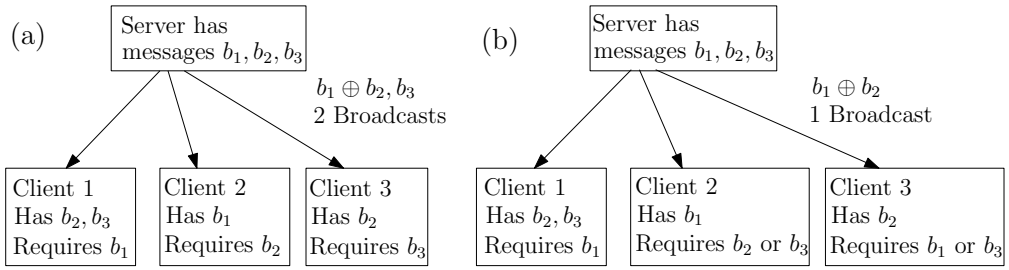


Figure 1.1 – (a) Index coding instance needs 2 broadcast transmissions and (b) PICOD(1) instance needs just one broadcast transmission

which consists of

1. A linear encoding function E mapping $x \in \mathcal{F}^m$ to $E(x) \in \mathcal{F}^l$, where $l = |\mathcal{C}|$ is the length of the code.

2. Linear decoding functions $D_1, \cdots, D_n$ for the n clients such that $D_i(E(x), b_{N_c[i]}) = \{b_{k_{i,1}}, \ldots, b_{k_{i,t}}\}$ for *some* t *distinct* indices $k_{i,1}, \ldots, k_{i,t} \in \overline{N_c[i]} = [m] \setminus N_c[i]$.

The goal then is to find a code with the minimum length which is also the number of broadcast transmissions.

To illustrate the difference between index coding and PICOD, consider the scenario shown in Figure 1.1. The server has three messages (in this case bits) b_1, b_2, b_3 and there are three clients with the side information sets shown in the figure. In a specific index coding instance, client 1 wants message b_1, client 2 wants b_2 and client 3 wants b_3. For solving this, at least 2 broadcast transmissions are needed. Client 1 can decode b_1 from $b_1 + b_2$ as she knows b_2, where $+$ here denotes addition in the binary field $GF(2)$. Client 2 can decode b_2 from $b_1 + b_2$ as well, and client 3 gets b_3 directly. It is easy to see that one transmission does not suffice in this case. On the other hand, in PICOD, it is sufficient to send just $b_1 + b_2$ as clients 1 and 3 can decode $b_1 = b_2 + (b_1 + b_2)$ while client 3 can decode $b_2 = b_1 + (b_1 + b_2)$.

The *Oblivious Pliable Index Coding Problem* OB-PICOD(t) models a situation where the server has limited information about the side information sets $N_c[i]$. More concretely, we assume that the server only knows the cardinalities of the side information sets $|N_c[i]|$. The (linear) OB-PICOD(t) problem then is to construct a linear code $\mathcal{C}$ which consists of

1. A linear encoding function E mapping $x \in \mathcal{F}^m$ to $E(x) \in \mathcal{F}^l$, where $l = |\mathcal{C}|$ is the length of the code.

2. Linear decoding functions $D_1, \cdots, D_n$ for the n clients such that $D_i(E(x), b_{N_c[i]}) = \{b_{k_{i,1}}, \ldots, b_{k_{i,t}}\}$ for *some* t *distinct* indices $k_{i,1}, \ldots, k_{i,t} \in \overline{N_c[i]} = [m] \setminus N_c[i]$.

Note that since the server does not have the exact side information sets but only their cardinalities, an encoding scheme should be able to deal with all possible such sets with the given cardinalities.

1.3 PICOD(t) is NP-Hard

For given side information sets, the length of the optimal pliable index code cannot be worse than the length of the optimal index code. This is because the index code encodes for a specific set of required messages, which is just one of the many configurations allowed in the pliable case. However, as we show in this section, computing the pliable index code of minimum length remains an NP-Hard problem. This will be accomplished by reducing an instance of the MONOTONE-1in3-SAT problem to an instance of PICOD(1).

Given a 3SAT instance ϕ with all variables in non-negated form, the MONOTONE-1in3-SAT problem asks whether there is a satisfying assignment such that exactly one variable is `True` in each clause of the formula. MONOTONE-1in3-SAT has been shown to be NP-Hard by Schaefer [27]. Suppose ϕ is made up of M variables $\alpha_i, \cdots, \alpha_M$ and N_0 clauses

$$\phi(\alpha_1, \cdots, \alpha_M) = \bigwedge_{i=1}^{N_0} (\alpha_{i,1} \vee \alpha_{i,2} \vee \alpha_{i,3}) \tag{1.1}$$

where clause i is a disjunction of the variables $\alpha_{i,1}, \alpha_{i,2}, \alpha_{i,3}$. The precise reduction is shown in the following lemma.

Lemma 1.3.1 *Given an instance ϕ of MONOTONE-1in3-SAT as defined above, there is an instance I_{ϕ,M,N_0} of PICOD(1) such that ϕ has a satisfying assignment if and only if there is a code of length 1 for I_{ϕ,M,N_0}.*

Proof: Given the MONOTONE-1in3-SAT instance ϕ, consider an instance I_{ϕ,M,N_0} of PICOD(1) defined as follows:

1. There are N_0 clients c_i, $i \in [N_0]$ corresponding to the clauses where c_i corresponds to clause i.
2. There are M messages b_j, $j \in [M]$ corresponding to the variables where message b_j corresponds to variable α_j. Here the choice of the field does not matter and can be chosen to be $GF(2)$.
3. The side information set for c_i consists of all the messages that *do not* correspond to the variables in clause i. That is

$$N_c[i] = \{j : \text{variable } \alpha_j \text{ is not in clause } i\} \tag{1.2}$$

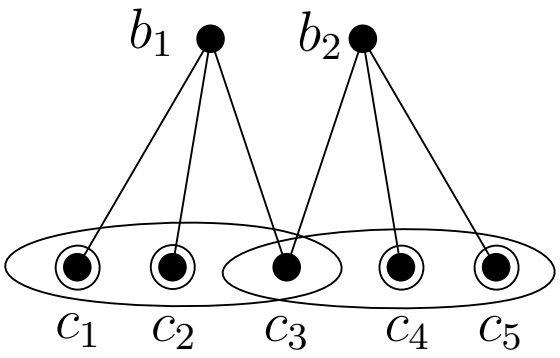


Figure 1.2 – Covering of the client vertices by neighboring message vertices. Here $B = \{b_1, b_2\}$ and $W_1(B) = \{c_1, c_2, c_4, c_5\}$.

Therefore, $|N_c[i]| = M - 3$ and $|\overline{N_c[i]}| = 3$ for all $i \in [N_0]$.

Suppose there exists a linear code of length 1 that is a solution to I_{ϕ,M,N_0}. It is necessarily of the following form

$$S = b_{j_1} + b_{j_2} \cdots + b_{j_s} \tag{1.3}$$

for some $j_1, \cdots, j_s \in [M]$. Let $J_s = \{j_1, \cdots, j_s\}$. Since every client c_i must be able to decode at least one message not in $b_{N_c[i]}$ and there is only one coded message, $\forall i \in [N_0]$, there exists $j_{k_i} \in J_s$ such that $j_{k_i} \in \overline{N_c[i]}$. Since the corresponding message has to be decodable by c_i, there can be at most one such index. Thus, the set J_s has the property that exactly one of its members is present in each $\overline{N_c[i]}$. Clearly, if we set the corresponding variables $\{\alpha_{j_k}, 1 \leq k \leq s\}$ to `True` and others to `False`, we make sure that all the clauses (which correspond to the clients) are satisfied and exactly one variable in each clause is `True`, which therefore satisfies ϕ. Thus, a code of length 1 for I_{ϕ,M,N_0} can be used to generate a satisfying assignment for ϕ that has exactly one `True` variable in each clause. Exactly the same argument can be reproduced backwards to prove the converse, which completes the reduction. Finally, it is easy to see that the reduction can be accomplished in polynomial time. ■

Since MONOTONE-1in3-SAT is NP-Hard, Lemma 1.3.1 implies that computing the minimum length code in PICOD(1) is also NP-Hard. This also implies that computing the minimum length code for PICOD(t) in general is NP-Hard.

1.4 Upper Bounds for PICOD(t)

We know that for the standard index coding problem there are instances which require $\Omega(n)$ coded messages. Is this also the case with PICOD(t)? We show in this section that for PICOD(t) we can do exponentially better.

In proving the results claimed in this and subsequent sections, we will be using the following result from linear algebra.

Lemma 1.4.1 *Given n_0 symbols in a large enough field $\mathcal{F}$ there exist $k_0 \leq n_0$ linear combinations of the symbols in $\mathcal{F}$ such that any subset of $k_1 \leq k_0$ symbols can be recovered if the remaining $n_0 - k_1$ symbols are known.*

In fact if the field size is greater than n_0, k_0 random linear combinations will have the above property with high probability [28]. This will be a key ingredient in our proofs, so we will assume that the field $\mathcal{F}$ in which the messages reside is of size greater than m (the number of messages). Unless otherwise stated, the addition operation in $\mathcal{F}$ will be denoted by $+$.

We will first consider PICOD(1). We can visualize an instance of PICOD(1) using a bipartite graph G with m vertices on one side representing the messages (called "message vertices") and n vertices on the other side representing the clients (called "client vertices"). We will identify the vertices by the messages or clients they represent. There is an edge from b_j to c_i if $j \in \overline{N_c[i]}$, i.e., when message b_j is not in the side information set of client c_i. In Figure 1.2 shown above, message b_1 is *not in* the side information sets of clients c_1, c_2, c_3 and hence is connected to them in G. In what follows, we will denote the neighborhood of c_i in G by $N[c_i]$ and its degree by $d(c_i) = |N[c_i]|$. Similarly, $N[b_j]$ is the neighborhood of b_j in G and $d(b_j) = |N[b_j]|$ is its degree.

Consider two messages b_1 and b_2 and their neighborhoods $N[b_1]$ and $N[b_2]$ in G. We distinguish the client vertices in $N[b_1] \cup N[b_2]$ into two types, depending on the number of message vertices they are adjacent to. The set of client vertices that are adjacent to exactly one of the message vertices in B, is denoted by $W_1(B)$. In Figure 1.2, for $B = \{b_1, b_2\}$, $W_1(B) = \{c_1, c_2, c_4, c_5\}$ and is depicted by the double circles. Note that if $b_1 + b_2$ is sent to these $|W_1(B)| = 4$ vertices, each of them can decode a message she does not have: c_1 and c_2 can decode b_1 as they know b_2; similarly, c_4 and c_5 can decode b_2 as they know b_1. On the other hand, the set of clients which are adjacent to more than one message vertex, as is $\{c_3\}$, can decode neither b_1 nor b_2. The same logic can be extended if B contains more than two message vertices: if we transmit the sum of the messages in B, all the $|W_1(B)|$ client vertices will be able to decode a message they do not have; in other words, it is sufficient to broadcast the sum message to "satisfy" all the $|W_1(B)|$ clients. We use this intuition to prove the following lemma.

Lemma 1.4.2 *Without loss of generality, let $C = \{c_1, c_2, \ldots, c_k\}$ be any group of k client vertices and $d_{\max} = \max\{d(c_i) | \, i \in [k]\}$ and $d_{\min} = \min\{d(c_i) | \, i \in [k]\}$. For some fixed constant $r \geq 1$, let $d_{\max} \leq r d_{\min}$. Then there is a code of length $O(\log k)$ that "satisfies" all the clients in C.*

Proof: We will use a probabilistic argument. Consider the neighborhood B_0 of C in G, i.e., $B_0 = \cup_{i=1}^{k} N[c_i]$. Randomly select a subset B_1 of message vertices by selecting each vertex of B_0 with probability p (which will be determined later). Then the probability P_i of c_i being adjacent to exactly one vertex in B_1 is

$$P_i = d(c_i)p(1-p)^{d(c_i)-1} \tag{1.4}$$

The expected number of vertices in $W(B_1)$ is

$$E_p[|W(B_1)|] = \sum_{i=1}^{k} d(c_i)p(1-p)^{d(c_i)-1} \geq k d_{\min} p(1-p)^{d_{\max}-1} \tag{1.5}$$

The expression $p(1-p)^{x-1}$ is maximized for $p = \frac{1}{x}$. Therefore by selecting $p = \frac{1}{d_{\max}}$ we get

$$E_p[|W(B_1)|] \geq k\frac{d_{\min}}{d_{\max}}(1-d_{\max})^{d_{\max}-1} \geq k\frac{d_{\min}}{ed_{\max}} \geq \frac{k}{er} \tag{1.6}$$

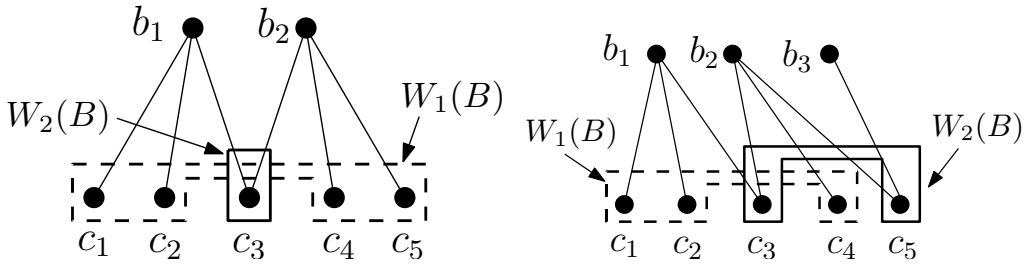
By the probabilistic method, there is at least one subset of B_1 for which $|W(B_1)| \geq \frac{k}{er}$ which means the sum of the bits in B_1 can satisfy a constant fraction of the k client vertices. We are then left with at most $k(1-\frac{1}{er})$ client vertices. The ratio of the maximum and minimum degrees in this set is also bounded by r and hence the argument can be repeated until only a constant number of them are left. Since the number of client vertices reduces by a constant fraction in each iteration, at most $O(\log k)$ coded messages are required to satisfy all the k client vertices. ■

In particular, if the cardinalities of the side information sets of all the clients are equal, a code of length $O(\log n)$ is sufficient to satisfy all the clients. For a general instance of PICOD(1) where the cardinalities of the side information sets are arbitrary, we use a suitable partition of the client vertices along with the above lemma to prove the following result.

Theorem 1.4.3 *For any PICOD*(1) *instance with m messages and n client vertices, all the client vertices can be satisfied with a code of length $O(\min(\log m \log(\frac{n}{\log m})), m, n)$.*

Proof: The degrees of the client vertices can range from 1 to n. Partition the vertices into g subsets $S_1, \ldots, S_g$ such that $S_i = \{c_l |\, 2^{i-1} \leq d(c_l) \leq 2^i\}$. For the non-empty ones, clearly the ratio of the maximum and minimum degrees in each of the sets S_i is at most 2 and $g \leq \lceil \log_2(m) \rceil$. Therefore, by the previous lemma, we need at most $K_1 \log(|S_i|)$ messages to satisfy the clients in S_i, for some absolute constant K_1. The total number of coded messages required is at most

$$K_1 t \sum_{i=1}^{g} \log(|S_i|) \leq K_1 t g \log\left(\frac{\sum_{i=1}^{g}|S_i|}{g}\right) = O(t \log m \log\left(\frac{n}{\log m}\right)) \tag{1.7}$$



(a) Here $B = \{b_1, b_2\}$ and $W_1(B) = \{c_1, c_2, c_4, c_5\}$ and $W_2(B) = \{c_3\}$

(b) Here $B = \{b_1, b_2, b_3\}$ and $W_1(B) = \{c_1, c_2, c_4\}$ and $W_2(B) = \{c_3, c_5\}$

Figure 1.3 – Covering of the client vertices by neighboring message vertices.

where the inequality follows from Jensen's inequality applied to the log(.) function. In addition, it is easy to see that broadcasting all the messages or broadcasting one message not in the side information of each client also satisfies all the clients. Combining all the three, we conclude that $O(\min(\log m \log(\frac{n}{\log m}), m, n))$ messages are sufficient for satisfying all the clients. ■

In particular, if the number of messages is polynomially related to the number of clients, i.e., $m = O(n^\delta)$ for some constant δ, then a code of length $O(\log^2 n)$ is sufficient for any instance of PICOD(1). This is exponentially better than the $\Omega(n)$ messages required for index coding in the worst case.

We will now generalize the bounds for PICOD(1) to PICOD(t). Given a subset of message vertices B, we can categorize the client vertices in the neighbourhood of B according to the number of message vertices they are adjacent to. The set of client vertices that are adjacent to exactly i message vertices in B, is denoted by $W_i(B)$. In Figure 1.3a, for $B = \{b_1, b_2\}$, $W_1(B) = \{c_1, c_2, c_4, c_5\}$ and $W_2(B) = \{c_3\}$. Note that if $b_1 + b_2$ is sent to the 4 vertices in $W_1(B)$, each of them can decode a message it does not have: c_1 and c_2 can decode b_1 as they know b_2; similarly, c_4 and c_5 can decode b_2 as they know b_1. Thus a single linear combination of the messages in B can satisfy all the client vertices in $W_1(B)$. Similarly, two independent linear combinations of b_1, b_2 can be used by client vertices in $W_2(B)$ to decode two messages.

In another example, consider the graph in Figure 1.3b. In this case $B = \{b_1, b_2, b_3\}$, $W_1(B) = \{c_1, c_2, c_4\}$ and $W_2(B) = \{c_3, c_5\}$. Similar to above, if $b_1 + b_2 + b_3$ is sent to the 3 vertices in $W_1(B)$, each of them can decode a message it does not have. For the two vertices in $W_2(B)$, notice that they are adjacent to two different subsets of message vertices. In this case, we can apply Lemma 1.4.1 to get two linear combinations of $\{b_1, b_2, b_3\}$ such that each of c_3, c_5 can decode two messages it does not have.

For any subset of message vertices B, define $t(B) = \max\{i : \text{ s.t. } W_i(B) \neq \emptyset\}$. In general, using the same reasoning as above, we have:

Lemma 1.4.4 *For any set of message vertices B, there exists a set of $t(B)$ linear combinations of the messages in B such that each client vertex in $W_i(B)$ can decode i messages it does not have, for $i \in [t(B)]$.*

Our approach then will be to select a suitable subset of message vertices such that a *large* number of client vertices have approximately t (or a constant fraction of t) message vertices adjacent to them. To this end, we will use a probabilistic argument. Similar to Lemma 1.4.2, the following main lemma proves an upper bound to the length of codes required to solve PICOD(t) for the case when the side information sets have low variation in their cardinalities.

Lemma 1.4.5 *Without loss of generality, let $C = \{c_1, c_2, \ldots, c_k\}$ be any group of k client vertices and $d_{\max} = \max\{d(c_i) | \, i \in [k]\}$ and $d_{\min} = \min\{d(c_i) | \, i \in [k]\}$, where $d_{\min} \geq t$. For some fixed constant $r \geq 1$, let $d_{\max} \leq r d_{\min}$. Then there is a code of length $O(\min(t \log k, t + \log^2 k))$ such that each client vertex can decode t messages it does not have.*

Proof: Randomly select a subset B_1 of message vertices by selecting each vertex with probability $p = \frac{t}{d_{\max}}$. Fix a particular client vertex c_1 (without loss of generality) with degree d in the graph. Let X_i denote the indicator variable which is 1 if the i-the neighbour of c_1 is present in the sample B_1, for $i \in [d]$. Clearly, the X_i's are i.i.d Bernoulli random variable with $P(X_i = 1) = p$. Let $X = X_1 + \ldots + X_d$. Then we have

$$E[X] = E[\sum_{i=1}^{d} X_i] = \sum_{i=1}^{d} E[X_i] = dp = \frac{dt}{d_{\max}} \tag{1.8}$$

Therefore, $t \geq E[X] \geq \frac{t}{r}$. We will also need concentration bounds on $E[X]$. Since X is a sum of i.i.d Bernoulli random variables, we can use the following Chernoff bounds which are valid for any $\epsilon \in (0, 1)$ [29].

$$P(X < (1 - \epsilon)E[X]) \leq e^{-\frac{\epsilon^2}{2} E[X]} \tag{1.9}$$

and

$$P(X > (1 + \epsilon)E[X]) \leq e^{-\frac{\epsilon^2}{3} E[X]} \tag{1.10}$$

Assume that $t \geq 24 r \log k$. If we choose $\epsilon = \sqrt{\frac{6r \log k}{t}}$, then clearly $\epsilon \leq \frac{1}{2}$. Then, using the fact that $E[X] \geq t/r$, we have

$$\begin{aligned} P(X > 3E[X]/2) \leq P(X > (1 + \epsilon)E[X]) &\leq e^{-\frac{\epsilon^2}{3} E[X]} \\ &\leq e^{-\frac{6r \log n}{3t} \cdot \frac{t}{r}} = e^{-2 \log n} = n^{-2} \end{aligned} \tag{1.11}$$

Thus we can conclude that

$$P(X > 3E[X]/2) \leq n^{-2} \tag{1.12}$$

Similarly,

$$\begin{aligned} P(X < E[X]/2) \leq P(X < (1-\epsilon)E[X]) &\leq e^{-\frac{\epsilon^2}{2}E[X]} \\ &= e^{-\frac{6r\log n}{2t}\cdot\frac{t}{r}} = e^{-3\log n} = n^{-3} \end{aligned} \tag{1.13}$$

Thus, we can conclude that

$$P(X < E[X]/2) \leq n^{-3} \tag{1.14}$$

Combining the above two results, we have

$$P(\frac{E[X]}{2} \leq X \leq \frac{3E[X]}{2}) \geq 1 - \frac{1}{n^2} - \frac{1}{n^3} \tag{1.15}$$

which implies that with high probability a particular client vertex has between $E[X]/2$ and $3E[X]/2$ adjacent message vertices in B_1. Therefore, the expected number of client vertices having the same property is at least $n - 1/n - 1/n^2$. By the probabilistic method there is at least one subset B_1 for which the expected value is reached or surpassed. This implies that there is a subset B_1 such that all client vertices have between $E[X]/2$ and $3E[X]/2$ message vertices adjacent to them (here there is slight abuse of notation and $E[X]$ for each client may be different). By an application of Lemma 1.4.4, there is a set of at most $3E[X]/2 \leq 3t/2$ linear combinations of the corresponding messages such that each client vertex can decode at least $E[X]/2 \geq t/2r$ messages it does not have. Note that if $3t/2 > d_{\max}$, then the number of messages can be cut off at $d_{\max}$.

What this shows is that if $t \geq 24r\log k$, then by using $O(t)$ broadcast messages we can make sure all the client vertices learn at least $t/2r$ new messages. We can now recursively use the same argument for a situation where each client now needs (at most) $t' = t(1 - 1/2r)$ new messages, stopping when t' becomes less than $24r\log k$. In the case when $t' < 24r\log k$, we can use t' rounds of the argument given in Lemma 1.4.2 for PICOD(1).. Thus if $f(k,t)$ is the number of messages required for sending t unknown messages to each of the k clients, we get the following recurrence

$$f(k,t) \leq \begin{cases} f(k, t(1-\frac{1}{2r})) + O(t) \text{ if } t \geq 24r\log k \\ O(tr\log k) \text{ otherwise} \end{cases} \tag{1.16}$$

This recurrence can be solved to get the required bound of $f(k,t) = O(\min(t\log k, t + \log^2 k))$. ■

In particular, if the cardinality of the side information sets of all the clients is the same,

a code of length $O(\min(t\log n, t+\log^2 n))$ is sufficient to satisfy all the clients. In this case, when $t \gg \log n$, the number of messages required grows linearly with t.

For the general case of client vertices having arbitrary degrees, we can partition them into at most $\log m$ groups such that the minimum and maximum degrees of the client vertices in each group are within a factor of 2. For each group, we can use the above result and derive the following bound on the length of codes.

Theorem 1.4.6 *For any PICOD(t) instance with m messages and n client vertices, all the client vertices can be satisfied with a code of length that is*

$$O(\min(t\log m\log\left(\frac{n}{\log m}\right), t\log m+\log m\log^2 n, m, tn)) \tag{1.17}$$

Proof: The degrees of the client vertices can range from 1 to m. Partition the vertices into g subsets $S_1,\ldots,S_g$ such that $S_i=\{c_l|\, 2^{i-1}\le d(c_l)\le 2^i\}$. For the non-empty ones, clearly the ratio of the maximum and minimum degrees in each of the sets S_i is at most 2 and $g\le\lceil\log_2(m)\rceil$. Therefore, by Lemma 1.4.5, we need at most the minimum of $K_1 t\log(|S_i|)$ and $K_2(t+\log^2(|S_i|))$ messages to satisfy the clients in S_i, for some absolute constants K_1, K_2. Using the first term, the total number of coded messages required can be upper bounded by

$$K_1 t\sum_{i=1}^{g}\log(|S_i|)\le K_1 tg\log\left(\frac{\sum_{i=1}^{g}|S_i|}{g}\right)=O(t\log m\log\left(\frac{n}{\log m}\right)) \tag{1.18}$$

Using the second term, the total number of coded messages required can be upper bounded by

$$\begin{aligned} &K_2\sum_{i=1}^{g}(t+\log^2(|S_i|))=K_2(tg+\sum_{i=1}^{g}\log^2(|S_i|))\le K_2(tg+g\log^2 n) \\ &=O(t\log m+\log m\log^2 n) \end{aligned} \tag{1.19}$$

Here we have used the fact that $|S_i|\le n$. In addition, it is easy to see that broadcasting all the messages or broadcasting t messages not in the side information set of each client also satisfies all the clients. Combining all the four bounds, we get our required result. ■

In particular, if the number of messages is polynomially related to the number of clients i.e. $m=O(n^\delta)$ for some constant δ, then a code of length $O(\min(t\log^2 n, t\log n+\log^3 n))$ is sufficient for any instance of PICOD(t). For random instances of PICOD(t) we have have the following result.

Theorem 1.4.7 *If each message appears in the side information set of a particular*

client with a fixed and constant probability q, then for a large enough n, almost surely any instance of PICOD(t) can be satisfied with a code of length $O(\min(t \log n, t + \log^2 n))$.

Proof: By the law of large numbers, the degree of each client vertex in the graph representation of a random instance of PICOD(t) is concentrated near the mean $n(1-q)$. For a fixed $\epsilon > 0$, for a large enough n almost surely $d(c_i) \in [n(1-q-\epsilon), n(1-q+\epsilon)]$. If we select an $\epsilon < q/3$, almost surely the ratio of the maximum and minimum degrees is ≤ 2. Then the claim follows from Lemma 1.4.5. ■

1.5 Bounds for OB-PICOD(t)

We now consider the OB-PICOD(t) problem. As defined in Section 1.2, in this problem the server only knows the size of the side information sets of the clients and has to broadcast messages such that each client can decode t new messages it does not have. We denote the maximum size of the side information sets by $s_{\max}$ and the minimum size by $s_{\min}$. We assume $s_{\max} \leq m - t$. Also, let Σ denote the set of distinct cardinalities of the side information sets in an instant. Our main result in this section is summarized in the following theorem.

Theorem 1.5.1 *For the OB-PICOD(t) problem, if each client knows at least $s_{\min}$ and at most $s_{\max}$ messages, then the server needs to broadcast at most $\min(s_{\max}+t, m-s_{\min})$ messages to satisfy all the clients. When $s_{\max} = s_{\min} = s$, this bound is tight for linear encoding and decoding.*

We prove this result by combining two lemmas, Lemma 1.5.2 and Lemma 1.5.3, that we next state and prove. The first lemma, Lemma 1.5.2, gives a simple upper bound on the length of codes for OB-PICOD(t).

Lemma 1.5.2 *To solve any instance of OB-PICOD(t), $\min(s_{\max}+t, m-s_{\min})$ coded messages are sufficient.*

Proof: There are two ways to make sure that all the clients are able to decode at least one message that they do not have. Select any $s_{\max}+t$ messages and send them uncoded, one at a time. Since the side information sets have size at most $s_{\max}$, there will always exist at least t messages that a client will not have. As another strategy, consider $m - s_{\min}$ linear combinations of all the messages obtained by the application of Lemma 1.4.1. Since each client knows at least $s_{\min}$ messages, it can recover at least t (in fact all) messages it does not know. Taking the minimum of the two strategies, we get an upper bound of $\min(s_{\max}+t, m-s_{\min})$ messages. ■

In particular when $s_{\min} = s_{\max} = s$, $\min(s+t, m-s)$ coded messages are sufficient to solve an instance of OB-PICOD(t). Further, if $s_{\max}$ is a constant, the server can satisfy all the clients using only t plus a constant number of broadcasts, without even knowing the exact side information sets.

We now derive lower bounds for OB-PICOD(t) for linear encoding and decoding, which match the upper bounds derived above for $s_{\max} = s_{\min} = s$. For this we will need some notation. Let $e_1, \ldots, e_m$ be the unit vectors in $\mathcal{F}^m$, i.e., e_i has a 1 (the identity element in $\mathcal{F}$) in the i-the position and 0 (the zero element in $\mathcal{F}$) elsewhere. Thus, they form an orthogonal basis for $\mathcal{F}^m$. Since the server uses linear encodings, the j-th encoded message can be represented as a dot product of an encoding vector $\mathbf{A}_j$ and the message vector $\mathbf{b} = \{b_1, \ldots, b_m\}$, with operations done over the field $\mathcal{F}$. For l encoded messages, we will have the corresponding l encoding vectors $\mathbf{A} = \{\mathbf{A}_1, \ldots, \mathbf{A}_l\}$. We will denote the vector space spanned by the vectors in $\mathbf{A}$ by $Span(\mathbf{A})$. For a subset of indices $B \subseteq [m]$, e_B will denote the corresponding set of unit vectors $e_i, i \in B$.

Lemma 1.5.3 *For linear encoding and decoding, any scheme for OB-PICOD(t) will need at least* $\max_{s \in \Sigma} \min(s+t, m-s)$ *messages. This simplifies to* $\min(s+t, n-s)$ *messages when* $s_{\max} = s_{\min} = s$.

Proof: Consider a particular s in Σ, the set of side information set cardinalities. The knowledge of the side information set can be expressed equivalently by the fact that the client can compute any vector in the span of $\mathbf{S} = \{e_{\alpha_1}, \ldots, e_{\alpha_s}\}$ where $\alpha_1, \ldots, \alpha_s$ are the indices of the messages that is in its side information set. For linear codes, for a client to be able to decode the i-th message using its side information sets and the encoded messages it is easy to see that e_i should belong to the span of $\mathbf{A} \cup \mathbf{S}$ (for a proof see [11]).

$$e_i \in Span(\mathbf{A} \cup \mathbf{S}) \tag{1.20}$$

Consider the set of all unique message indices that are decoded by *all* clients and call it B. We claim that $|B| \geq s + t$. Indeed, if $|B| < s + t$ then a client may have as side information, messages corresponding to a subset of size s of B. But then, it cannot decode t new messages that it does not have as B was assumed to contain *all* the decoded message indices.

For simplicity, assume that $|B| = s + t$, $s + t \leq m - s$ and without loss of generality $B = \{1, 2, \ldots, s + t\}$. Let $\mathbf{A}_\perp$ be the projection of $\mathbf{A}$ onto the first $s + t$ coordinates, i.e., the ones corresponding to B. Consider the following possible side information set:

$$\mathbf{S}_1 = \{e_{\alpha_{1,1}}, \ldots, e_{\alpha_{1,s}}\} \tag{1.21}$$

such that $\alpha_{1,1}, \ldots, \alpha_{1,s} \notin B$ (such a set exists as $s+t \leq m-s$). Clearly, by the condition of decodability of a new message (Eq. 1.20), at least one vector in e_B should be in $Span(\mathbf{A} \cup \mathbf{S}_1)$. Let it be e_{γ_1}. Since the $\alpha_{1,i}$ indices are disjoint from B, $e_{\gamma_1} \in Span(\mathbf{A}_\perp)$. Now consider the side information set

$$\mathbf{S}_2 = \{e_{\gamma_1}, e_{\alpha_{2,1}}, \ldots, e_{\alpha_{2,s-1}}\} \tag{1.22}$$

where $\alpha_{2,1}, \ldots, \alpha_{2,s-1} \notin B$. In this case an application of Eq. 1.20 implies that there exists $e_{\gamma_2} \in Span(\mathbf{A} \cup \mathbf{S}_2)$. Since e_{γ_2} is orthogonal to all the vectors in $\mathbf{S}_2$, $e_{\gamma_2} \in Span(\mathbf{A}_\perp)$. We can continue this argument for a total of $s+1$ steps where the last side information set is

$$\mathbf{S}_{s+1} = \{e_{\gamma_1}, \ldots, e_{\gamma_s}\} \tag{1.23}$$

In this case, since t new messages need to be decoded by the client, we can conclude that there are vectors $e_{\gamma_{s+1}}, e_{\gamma_{s+2}}, \ldots, e_{\gamma_{s+t}}$ that are not in $\mathbf{S}_{s+1}$ but are in $Span(\mathbf{A} \cup \mathbf{S}_{s+1})$, which implies they lie in $Span(\mathbf{A}_\perp)$. Since the vectors $e_{\gamma_1}, \ldots, e_{\gamma_{s+t}}$ are orthogonal and all of them lie in $Span(\mathbf{A}_\perp)$, $\mathbf{A}_\perp$ and by implication $\mathbf{A}$ must contain at least $s+t$ linearly independent vectors.

The other case where $s+t > m-s$ can be handled in a similar manner, where instead of $s+t$ sets, we will have $n-s$ side information sets for which the new messages will correspond to orthogonal vectors. Combining the two, we conclude that $\mathbf{A}$ must contain at least $\min(s+t, m-s)$ linearly independent vectors, which implies at least the same number of broadcasts need to be made. Finally, the assumption of $|B| = s+t$ can be removed by simple choosing the first $s+t$ elements of B.

The above argument can be repeated for each $s \in \Sigma$ and hence we get the best lower bound by taking the maximum of the individual bounds. Clearly, when $s_{\max} = s_{\min} = s$, there is only one element in Σ and the lower bound becomes $\min(s+t, m-s)$, which matches the upper bound in Lemma 1.5.2. ■

1.6 Heuristic Approximation Algorithms

In this section we propose polynomial time heuristic approximation algorithms for solving the PICOD(t) problem. Our main algorithm uses a greedy approach to iteratively find large subsets of clients that can be satisfied with a single coded message. In addition, we present two other algorithms specifically for PICOD(1) – one based on the upper bound proof in Section 1.4 and the other based on a reduction to the index coding problem.

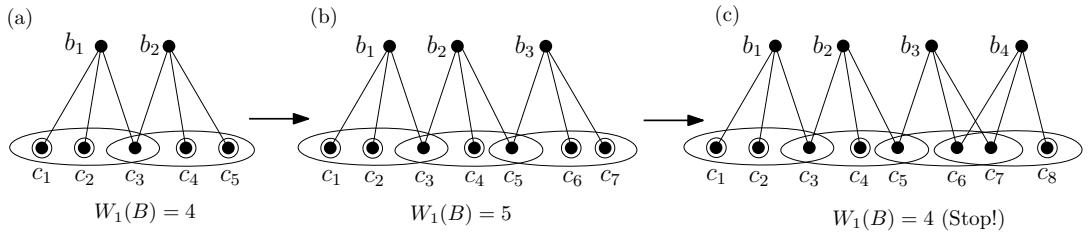


Figure 1.4 – Greedy construction of B with maximal $|W_1(B)|$.

1.6.1 Algorithm `GRCOV`

In this algorithm, using the graphical representation defined in Section 1.4, we try to find a set of message vertices B such that $|W_1(B)|$ is maximized. Rather than trying to obtain the *maximum* such set, we greedily find a *maximal* such set. Let $B = \{b_{v_1}, \cdots, b_{v_t}\}$ be a set of message vertices. B is a maximal set if for any message vertex $b_{v_{t+1}} \notin B$, $|W_1(B \cup \{b_{v_{t+1}}\})| < |W_1(B)|$. To find a maximal set, we start with the empty set and keep on adding message vertices that greedily maximizes $|W_1(B)|$ in each step; we stop when no further additions are possible without decreasing $|W_1(B)|$. For example, Figure 1.4 represents a possible sequence of operations where $B = \{b_1, b_2, b_3\}$ is a maximal set. When b_3 is added, the cardinality of $W(B)$ increases but further addition of b_4 decreases it, in which case we stop.

We maintain a counter $CNT[i]$ for each client vertex i to keep track of the remaining number of messages they want. For an instance of PICOD(t), they are all initialized to t. After finding a particular B for which $|W_1(B)|$ is large we can use the sum of the messages in B to satisfy the message vertices in $W_1(B)$. The edges connecting the vertices in $W_1(B)$ to the message vertices it can decode are removed and the value of $CNT[i]$ for each vertex in $W_1(B)$ is also reduced by one. The algorithm is resumed for the remaining graph, until all the $CNT[i]$ values go to zero. We call this algorithm `GRCOV` (for greedy cover). It is shown in pseudo-code format below and a simple implementation of the algorithm has a running time of $O(mn^2t)$.

1.6.2 Algorithm `RANDCOV`

The `RANDCOV` (for randomized cover) algorithm for PICOD(1) follows the procedure in the proof of Theorem 1.4.3 (in Section 1.4) to find an encoding. The client vertices are partitioned into at most $g = O(\log m)$ groups $S_1, \ldots, S_g$ such that the ratio of the maximum and minimum degrees in each group is at most r (a fixed constant). Let the maximum degree of client vertices in S_i be $d_{\max,i}$. In the neighborhood $N[S_i]$, we select each vertex with probability $p_i = \frac{1}{d_{\max,i}}$. If B_i is the set of selected vertices, the clients in $W(B_i)$ are satisfied. This process is continued until all the vertices in S_i are satisfied. The number of randomly sampled sets required is also the number of coded messages required. This is done for each of the sets S_i to find a code for all the client vertices.

Algorithm 1 `GRCOV`(G, m, n, t)

Init: G is an instance of PICOD(t) with n client vertices and m message vertices.
Init: $\mathcal{C} = \{\}$, $CNT[i] = t$ for $i \in [n]$.
  while $\exists i$ s.t. $CNT[i] \neq 0$ **do**
    $B \leftarrow \varnothing$.
    while B is not a *maximal set* **do**
      Find message vertex $b_v \notin B$ such that $|W_1(B \cup \{b_v\})|$ is maximized.
      $B \leftarrow B \cup \{b_v\}$.
    end while
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{\sum_{u=1}^{|B|} b_{v_u}, b_{v_u} \in B\}$.
    for $c_i \in W_1(B)$ **do**
      If c_i is able to decode b_j using the above encoding, then delete the corresponding edge in G.
      $CNT[i] \leftarrow CNT[i] - 1$.
    end for
  end while
  Output $\mathcal{C}$.

The expected running time of `RANDCOV` is $O(mn \log n)$.

Algorithm `RANDCOV-PP` Although the expected length of the code produced by `RANDCOV` is upper bounded by the terms derived in Section 1.4, as we shall see in the next section, a simple implementation does not perform very well as compared to `GRCOV` on random instances of PICOD(1). To make it more efficient we propose the following post processing phase. Let B_1 and B_2 be two sets of message vertices and let the corresponding client vertex sets that they satisfy be $W_1(B_1)$ and $W_1(B_2)$ respectively. If B_1 has no edges to $W_1(B_2)$ and B_2 has no edges to $W_1(B_1)$, we can send the sum of all the messages in $B_1 \cup B_2$ to satisfy all the client vertices in $W_1(B_1) \cup W_1(B_2)$. This can be extended to include more than two sets by selecting the sets greedily. When this post-processing step is added to `RANDCOV`, we call the algorithm `RANDCOV-PP`. The expected running time of `RANDCOV-PP` remains $O(mn \log n)$.

1.6.3 Algorithm `ICOD-SETCOV`

Finally, we propose another algorithm for PICOD(1) that is based on a reduction to the index coding problem. In an instance of PICOD(1), it is sufficient that c_i is able to decode any one message in $N[c_i]$. We split client c_i into $|N[c_i]|$ "pseudo-clients" each with a *distinct* message from $N[c_i]$ as a requirement and with the same *common* side information sets. Therefore, in total we get $\sum_{i=1}^{n} |N[c_i]|$ pseudo-clients. This is an instance of the index coding problem and can be solved using one of the algorithms proposed in [6]. We

use the simplest one based on greedy clique cover.

Let the set of encoded messages be E. For the greedy clique cover algorithm, each encoded messages allows for decoding in one step. In other words, each client can decode its required message using just one encoded message and each encoded message can satisfy the requirements of a certain number of pseudo-clients. This naturally defines a "covering" relationship where an encoded message covers a set of pseudo-clients. Also note that each pseudo-client corresponds to an original client, the one from which it was created. Therefore, for each encoded message $e_k \in E$ we can define the set of original clients that it "covers" as

$$Cov(e_k) = \{c_{k,1}, \cdots, c_{k,s_k}\} \subseteq \{c_1, \cdots, c_n\} \tag{1.24}$$

In fact, the same client can occur in several of these covering sets. Since we only need a client to be able to decode a single message that it does not know, it is sufficient to find a collection of e_k such that the corresponding $Cov(e_k)$-s cover all the clients. Further we want to minimize the size of this collection for the optimal encoding. This is precisely an instance of the minimum set cover problem with clients being the elements and the $Cov(e_k)$ being the sets. In our implementation, we use the standard greedy approximation algorithm to solve it. We call this algorithm `ICOD-SETCOV` (for index coding set-cover) and the running time for a non-optimized implementation is $O(m^2n^2)$.

1.7 Simulation Results

In this section, we present results of extensive simulations on random instances of PICOD(t) to evaluate the performance of the algorithms presented in the previous section. We will first analyze the performance of the all the three algorithms for PICOD(1). The number of clients and messages is chosen to be both $m = n = 512$ and the field in which the messages lie is $\mathcal{F} = GF(2)$. For `RANDCOV` and `RANDCOV-PP` we choose $r = 3$. The random instances of PICOD(1) are generated as a function of the probability p_{msg} of a client having a particular message in its side information set. Thus, for each client and each message we choose to have the message in the side information set of the client independently with probability p_{msg}. Equivalently, in the graph representation of the instance each edge is present with a probability of $1 - p_{\text{msg}}$. Such random instances can model block-fading in wireless channels, i.e., when the channel SNR is low, the client higher layers experience erasures with probability p_{msg}, while at a next block of high SNR, we want to perform "lossless" transmissions as efficiently as possible. Figure 1.5 shows the average performance of the algorithms over several runs (more than $10,000$ for each value of p_{msg}). For comparison, we also plot the performance of the clique cover algorithm presented in [6] for an instance of the index coding problem on the same randomly generated instances.

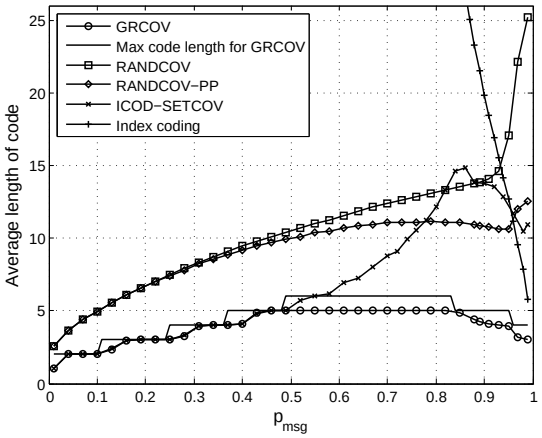
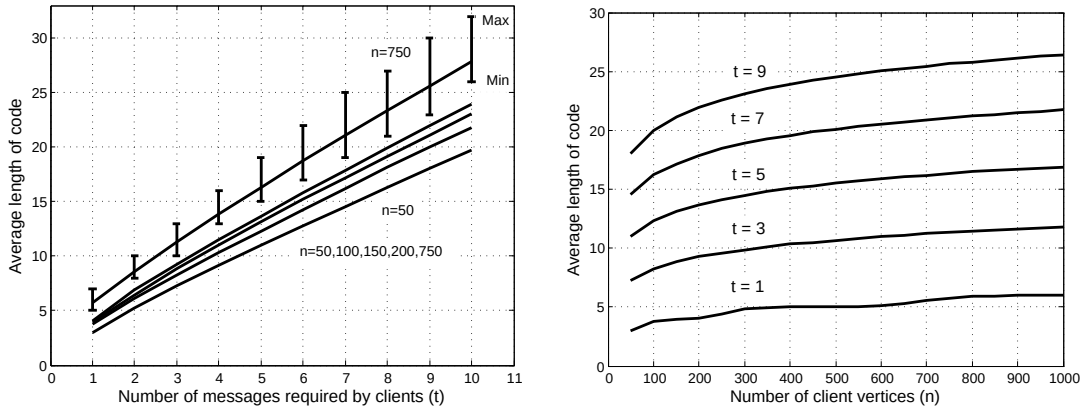


Figure 1.5 – Performance of PICOD(1) algorithms for varying p_{msg}

As expected, we observe a significant difference between the performance of the PICOD(1) algorithms and the index coding algorithm for the same p_{msg}. While all the three PICOD(1) algorithms proposed above take less than 26 bits on average, the index coding solution hovers in this range only for $p_{\text{msg}} \geq 0.87$ which is the case when the side information sets are dense. In the remaining range of p_{msg} values, all the PICOD(1) algorithms use fewer bits and the difference only becomes larger when the side information sets are sparser.

Among the four algorithms for PICOD(1) presented in the chapter, `GRCOV` performs the best. For the random graphs on which the simulations were run, arguments similar to the ones used in Section 1.4 can be used to show that it produces an encoding with the same asymptotic performance as `RANDCOV`, but the practical performance is much better. In fact, the maximum number of coded bits required by `GRCOV` (this is among the random instances in the simulation, not globally), which is also plotted in the figure, is not substantially different from the average number. The performance of `RANDCOV-PP` is substantially better than `RANDCOV`, especially when the side information sets are denser and hence the G is sparser. Also, the performance of `RANDCOV` takes a hit in this regime. Both of these are due to the fact that the number of partitions in the client vertices increases, although most of them are "disjoint" which allows `RANDCOV-PP` to improve the performance significantly. Finally, `ICOD-SETCOV` performs as good as `GRCOV` when $p_{\text{msg}} \leq 0.5$ but becomes worse as G becomes sparser. This can be partly explained by the suboptimal nature of the greedy set-cover algorithm that we are using inside `ICOD-SETCOV`.

To show the performance of the `GRCOV` algorithm for general instances of PICOD(t), we first generate random instances for a fixed $p_{\text{msg}} = 0.5$ and varying values of n and t. The number of messages m is taken to be equal to the number of clients n. Then, for a given value of n (number of clients/messages) and t (number of messages each client wants to know) we run `GRCOV` several times and take an average of the length of the code the



(a) Average code length for varying t as contours over n

(b) Average code length for varying n as contours over t

Figure 1.6 – Performance of `GRCOV` for different values of n and t on random instances of PICOD(t) with $p_{\text{msg}} = 0.5$.

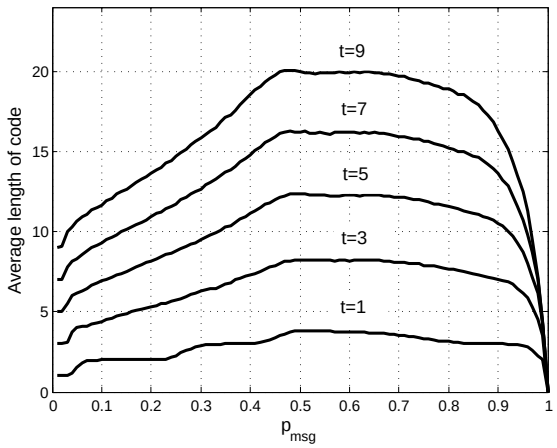


Figure 1.7 – Performance of `GRCOV` for varying p_{msg} and t.

algorithm produces. If in an instance, the size of the side information set for a particular client vertex is s and $t > n - s$, then $CNT[i]$ is initialized to $n - s$. We then plot the average length of the codes obtained versus t, for several values of n. The plot is shown in Figure 1.6a.

The figure has several trends that shows the efficacy of our algorithm. Notice that for each n, after a short initial phase, the average number of bits required grows almost linearly with t. This matches the trend expected from Theorem 1.4.7. Further, the contours representing different values of n are approximately parallel for $t > 5$ i.e. the slopes are independent of n. This also matches the trend suggested by an $O(t + \log^2 n)$ bound for large enough t. The error bars at the top, that represents the maximum (and minimum) length of the code encountered for random instances corresponding to $n = 750$,

also shows an approximately linear trend.

In Figure 1.6b, the parameter on the x-axis is changed to n and the lines correspond to particular values of t. Finally, in Figure 1.7, which is similar to Figure 1.5, we plot the dependence of the average code length with respect to p_{msg}, for different values of t. In both figures, we see that the behavior is not substantially different for different values of t, with curves essentially getting shifted upwards for higher values of t.

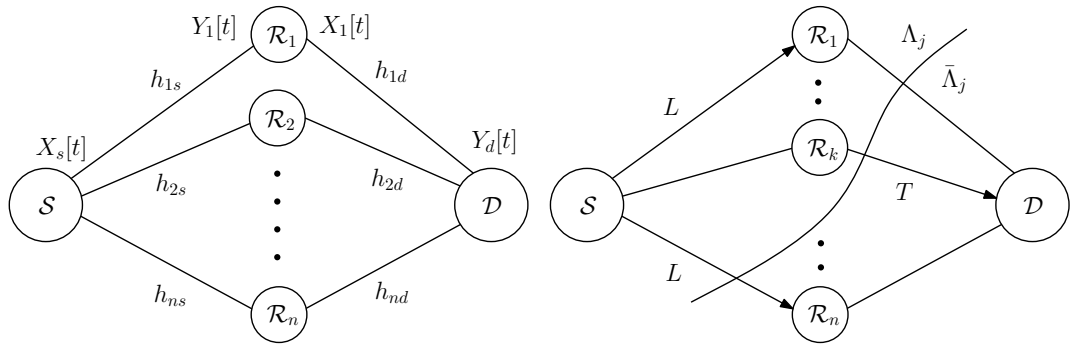
---

2 Optimal Schedules in Half-Duplex Diamond Networks

Calculating the capacity of half-duplex wireless relay networks is a hard problem. For such networks, since each relay can either be in a listening (L) or transmitting (T) state at any point of time, an additional dimension of optimization comes into play. For the n-relay half-duplex diamond network, shown in Fig. 2.1a, there exist 2^n possible combinations of L and T states and any capacity achieving strategy would need to optimize for how long each of these occurs. Even if the optimal schedule is actually computed, it is a non-trivial task to operate the network with many cooperating relays and many scheduling states.

In this chapter, we show that there might be no need for such an exponential size optimization and high operational complexity. Using a computational proof strategy, we show that for $n \leq 6$, the number of active states in the approximately optimal schedule is atmost $n + 1$. This is an exponential improvement over the possible 2^n scheduling states. We use simple approximations to the capacity of half-duplex diamond networks [8, 9, 10], which can be represented as linear programs, that are a constant gap (independent of channel SNRs) away from the capacity. The proof strategy then crucially uses *submodularity properties* of information flow across cuts in the network and *linear programming duality* to derive contradictions for optimal schedules having more than $n + 1$ states.

In fact, we conjecture that the above result is true for any n. The fact that a linear number of active states is sufficient is interesting both from a practical and theoretical perspective. Switching between $n + 1$ as opposed to 2^n states significantly reduces the operational complexity and could lead to practical schemes. From a theoretical point of view, this result reveals that the linear programs we are optimizing have a special structure, which is intimately linked to the flow of information across cuts in wireless networks. Indeed, our approach essentially identifies this structure and leverages it to prove the optimal bounds on the number of active states.



(a) The Gaussian n-relay half-duplex diamond network

(b) Relaying states and cuts in the network. For a particular relaying state, an arrow on a link denotes that it is active

Figure 2.1 – Network model with channel coefficients of the individual links, relaying states and cuts.

2.1 Related Work

The conjecture that $n + 1$ states are approximately optimal for n-relay half duplex diamond networks was first made in [30]. The result for $n = 2$ follows from the work of Bagheri et. al. [31]. The conjecture has since been generalized to arbitrary half-duplex relay networks in the work of Cardone et. al. [32], where the relevant term to be optimized is the generalized degrees of freedom (gDOF). In this case, the authors show that it holds for 2-relay networks.

2.2 Problem Formulation and Main Results

2.2.1 Network Model

We consider the Gaussian n-relay diamond network where a source $\mathcal{S}$ transmits information to a destination $\mathcal{D}$ with the help of half-duplex relays. At any given time t, each relay $\mathcal{R}_k$ can either listen (L) or transmit (T), but not both; we denote by $M_k[t] \in \{L, T\}$ its state. The source and the destination are assumed to be always in the T and L states, respectively.

Let $X_s[t]$ be the signal transmitted by $\mathcal{S}$ at time t, $X_k[t]$ be the signal transmitted by relay $\mathcal{R}_k$, $Y_d[t]$ and $Y_k[t]$ the signals received by $\mathcal{D}$ and $\mathcal{R}_k$, respectively. Then

$$\begin{aligned} X_k[t] &= 0 \text{ when } M_k[t] = L \\ Y_k[t] &= h_{is}X_s[t] + Z_k[t] \text{ when } M_k[t] = L \\ &= 0 \text{ when } M_k[t] = T \end{aligned}$$

$$Y_d[t] = \sum_{k=1}^{n} h_{kd} X_k[t] + Z[t] \text{ when } M_d[t] = L$$

where h_{ks}, h_{kd} are the complex channel coefficients between $\mathcal{S}$ and $\mathcal{R}_k$ and $\mathcal{R}_k$ and $\mathcal{D}$, respectively. $Z_k[t]$ and $Z[t]$ are i.i.d white Gaussian noise with unit variance. The power constraints for the source and all the relays are fixed to P. A representative figure is shown in Fig. 2.1a.

We can then calculate the individual link capacities from $\mathcal{S}$ to $\mathcal{R}_k$ (l_k) and from $\mathcal{R}_k$ to $\mathcal{D}$ (r_k) as

$$l_k = \log(1 + |h_{ks}|^2 P), \quad r_k = \log(1 + |h_{kd}|^2 P) \tag{2.1}$$

Let $[n]$ represent the set $\{1, 2, \cdots, n\}$. For $i \in [2^n]$, let $m_i \in M = \{L, T\}^n$ be a distinct relaying state, i.e., a particular configuration of listening and transmitting states for all the relays. The fraction of time the relays spend in state m_i will be denoted by p_i, where $\sum_{i \in [2^n]} p_i = 1$. We will use $L(m_i), T(m_i) \subseteq [n]$ to denote the set of indices of the relays in listening and transmitting state in m_i, respectively. Also, for $j \in [2^n]$, $\Lambda_j \subseteq [n]$ denotes the cut separating $\mathcal{S} \cup (\cup_{k \in \Lambda_j} \mathcal{R}_k)$ from $\mathcal{D} \cup (\cup_{k \in \bar{\Lambda}_j} \mathcal{R}_k)$. A representative cut is shown in Fig. 2.1b

To keep the exposition simple, we will assume that the l_k's and r_k's are all distinct. The term l-value(s) and r-value(s) will refer to the l_k's and r_k's, respectively. Finally, unless otherwise stated, the term "constant" will mean a quantity that depends only on the number of relays and is independent of the channel SNRs.

2.2.2 An Approximation to the Capacity

Let C_{hd}^n denote the capacity of the n-relay half-duplex diamond network; to achieve it, we need to optimize over p_i, the fraction of time that the relays are in state m_i. From the work of [8, 9, 10], C_{hd}^n can be approximated up to constant additive terms by a quantity C_{lp}^n that is a function only of the individual link capacities $\{l_i, r_i\}$ as defined in (2.1).

Theorem 2.2.1 *For an n relay half-duplex diamond network, there exist constants $G(n)$ and $G'(n)$ such that*

$$C_{lp}^n - G'(n) \leq C_{hd}^n \leq C_{lp}^n + G(n) \tag{2.2}$$

where

$$C_{lp}^n = \max_{p_i} \min_{j \in [2^n]} \sum_{i=1}^{2^n} p_i \left(\max_{k \in \bar{\Lambda}_j \cap L(m_i)} l_k + \max_{k \in \Lambda_j \cap T(m_i)} r_k \right) \tag{2.3}$$

$G(n)$ and $G'(n)$ are both $O(n)$ terms, independent of SNRs.

Let $\mathbf{p}$ denote the vector $(p_1, p_2, \ldots, p_{2^n})$. C_{lp}^n can also be viewed as the optimum solution of the following linear program, which we denote by **LP**.

LP : Maximize C

$$C_j(\mathbf{p}) \geq C \text{ for each } j \in [2^n] \tag{2.4}$$

$$\sum_{i=1}^{2^n} p_i = 1;\ \forall i, p_i \geq 0, C \geq 0 \tag{2.5}$$

where

$$C_j = C_{\Lambda_j} = C_j(\mathbf{p}) = \sum_{i=1}^{2^n} p_i \left(\max_{k \in \bar{\Lambda}_j \cap L(m_i)} l_k + \max_{k \in \Lambda_j \cap T(m_i)} r_k \right) \tag{2.6}$$

Each primal variable p_i denotes the fraction of time spent by the relays in state m_i and each constraint (except the last one) corresponds to a distinct cut Λ_j. $C_j(\mathbf{p})$ thus represents the information flow across the cut Λ_j over all the relaying states for the schedule $\mathbf{p}$. Thus, C_{lp}^n is the maximum value of the minimum information flow over all the cuts Λ_j, the maximization being done over all schedules $\mathbf{p}$ such that $\sum_{i \in [2^n]} p_i = 1$. We will also use the alternate notation of $C_{\Lambda_j}(\mathbf{p})$ or simply C_j to represent $C_j(\mathbf{p})$, when suitable.

LP is a maximization problem, and hence its dual, which is also a linear program, is a minimization problem. Let $\mathbf{p}^d$ denote the vector $(p_1^d, \ldots, p_{2^n}^d)$. The dual can be written as follows.

DLP : Minimize C^d

$$C_i^d(\mathbf{p}^d) \leq C^d \text{ for each } i \in [2^n] \tag{2.7}$$

$$\sum_{j=1}^{2^n} p_j^d = 1;\ \forall j, p_j^d \geq 0, C^d \geq 0 \tag{2.8}$$

where

$$C_i^d = C_{m_i}^d(\mathbf{p}^d) = C_i^d(\mathbf{p}^d) = \sum_{j=1}^{2^n} p_j^d \left(\max_{k \in \bar{\Lambda}_j \cap L(m_i)} l_k + \max_{k \in \Lambda_j \cap T(m_i)} r_k \right) \tag{2.9}$$

Each dual variable p_j^d corresponds to the cut Λ_j and each dual constraint (except the last one) corresponds to the relaying state m_i. p_j^d can be thought of as non-negative weights given to each cut with their sum normalized to one. $C_i^d(\mathbf{p}^d)$ thus represents the information flow when the relays are in state m_i over all the cuts for the dual vector $\mathbf{p}^d$. Thus, C_{lp}^n is the minimum value of the maximum information flow for all the relaying states m_i, the minimization being done over all dual vectors $\mathbf{p}^d$ such that $\sum_{j \in [2^n]} p_j^d = 1$. We will also use the alternate notation of $C_{m_i}^d(\mathbf{p}^d)$ or simply C_i^d to represent $C_i^d(\mathbf{p}^d)$ quantity, when suitable.

$l_k (k \in [n])$	Capacity of point to point channel from $\mathcal{S}$ to $\mathcal{R}_k$
$r_k (k \in [n])$	Capacity of point to point channel from $\mathcal{R}_k$ to $\mathcal{D}$
C^n_{hd}	Capacity of n-relay half-duplex diamond network
C^n_{lp}	Approximation to capacity through linear program
$m_i (i \in [2^n])$	Relaying state
$\Lambda_j (j \in [2^n])$	Cut in the network
p_i, p^d_j	The primal variable for state m_i and dual variable for cut Λ_j
$\mathbf{p}, \mathbf{p}^d$	The vector of primal variables p_i and dual variables p^d_j, both of length 2^n
$C_j(\mathbf{p}), C_j, C_{\Lambda_j}(\mathbf{p})$	Information flow through cut Λ_j in **LP** for primal vector $\mathbf{p}$
$C^d_i(\mathbf{p}^d), C^d_i, C^d_{m_i}(\mathbf{p}^d)$	Information flow for state m_i in **DLP** for dual vector $\mathbf{p}^d$

Table 2.1 – Summary of terms defined in this section.

As an example, consider a 2-relay network with $l_1 = 3, l_2 = 1, r_1 = 2$ and $r_2 = 4$. Here the primal vector $\mathbf{p} = (p_1, p_2, p_3, p_4)$ corresponds to the states $m_1 = \{L, L\}$, $m_2 = \{L, T\}$, $m_3 = \{T, L\}$, $m_4 = \{T, T\}$. The dual vector $\mathbf{p} = (p^d_1, p^d_2, p^d_3, p^d_4)$ corresponds to the cuts $\Lambda_1 = \emptyset$, $\Lambda_2 = \{2\}$, $\Lambda_3 = \{1\}$ and $\Lambda_4 = \{1, 2\}$. Then, C^2_{lp} is the objective value of the following primal and dual linear programs.

Maximize C	Minimize C^d
$\max(3,1).p_1 + 3.p_2 + 1.p_3 + 0.p_4 \geq C$	$\max(3,1).p^d_1 + 3.p^d_2 + 1.p^d_3 + 0.p^d_4 \leq C^d$
$3.p_1 + (3+4).p_2 + 0.p_3 + 4.p_4 \geq C$	$3.p^d_1 + (3+4).p^d_2 + 0.p^d_3 + 4.p^d_4 \leq C^d$
$1.p_1 + 0.p_2 + (1+2).p_3 + 2.p_4 \geq C$	$1.p^d_1 + 0.p^d_2 + (1+2).p^d_3 + 2.p^d_4 \leq C^d$
$0.p_1 + 4.p_2 + 2.p_3 + \max(2,4).p_4 \geq C$	$0.p^d_1 + 4.p^d_2 + 2.p^d_3 + \max(2,4).p^d_4 \leq C^d$
$p_1 + p_2 + p_3 + p_4 = 1$	$p^d_1 + p^d_2 + p^d_3 + p^d_4 = 1$
$p_1, p_2, p_3, p_4 \geq 0$	$p^d_1, p^d_2, p^d_3, p^d_4 \geq 0$

For this network, $C^2_{lp} = \frac{19}{10}$ with optimal primal vector $p_1 = \frac{1}{4}$, $p_2 = \frac{1}{5}$,$p_3 = \frac{11}{20}$, and $p_4 = 0$ and optimal dual vector $p^d_1 = \frac{1}{2}$, $p^d_2 = 0$, $p^d_3 = \frac{2}{5}$, $p^d_4 = \frac{1}{10}$.

There is a natural correspondence between cuts and the states making **LP** and **DLP** quite symmetrical. To take advantage of this fact, we will assume that the states and the cuts are numbered in such a way that Λ_j and $T(m_i)$ are equal as subsets of $[n]$ when $i = j$. For example, in the above program, $\Lambda_2 = \{2\}$ and $T(m_2) = \{2\}$. In the sequel, we will assume that the normalization constraints ($\sum_{i \in [2^n]} p_i = 1$ and $\sum_{j \in [2^n]} p^d_j = 1$) and non-negativity constraints ($C, C^d, \mathbf{p}, \mathbf{p}^d \geq 0$) are always present. The other constraints ($C_j(\mathbf{p}) \geq C$ and $C^d_i(\mathbf{p}^d) \leq C^d$) will be called primal and dual *flow constraints*, respectively.

The various terms defined in this section are summarized in Table 2.1.

2.2.3 Main Results

The number of active states in a schedule, i.e., the number of non-zero components in $\mathbf{p}$ is an indicator of schedule complexity. Let $(\tilde{\mathbf{p}}, C_{lp}^n)$ be the optimal solution to **LP**. In [30], the following conjecture was made on the number of active states in in $\tilde{\mathbf{p}}$.
Conjecture: *There exists an optimal solution $(\tilde{\mathbf{p}}, C_{lp}^n)$ to* **LP** *that has at most $n+1$ active states.*

If true, this will mean that schedules of only *linear* complexity, instead of the exponential, are enough to approximately achieve the capacity of diamond networks. In this chapter, we describe a computational proof strategy to resolve the conjecture and implement it for $n \leq 6$ to prove the following theorem.

Theorem 2.2.2 *There exists an optimal solution $(\tilde{\mathbf{p}}, C_{lp}^n)$ to* ***LP*** *that has at most $n+1$ active states for networks of size $n \leq 6$ relays.*

2.3 Proof Strategy and Key Techniques

In the present and the next two sections, we develop a computational proof strategy for proving Thm. 2.2.2. We start off by assuming that the optimal solution to **LP** has exactly $t > n+1$ active states. If **LP** is non-degenerate, there will both be a primal and dual optimum with *exactly* t active states. For each pair of such t-tuples (τ, τ_d) of active states in the primal and dual optimum, we will try to discover a contradiction. The contradiction can come from one of the following two sources that constrains these pairs: (i) *submodularity* relationships between the expressions $C_j(\mathbf{p})$ and $C_i^d(\mathbf{p}^d)$, or (ii) complementary slackness conditions of **LP** and **DLP**. Through several steps, we arrive at a contradiction by either showing that such a set of active state pairs violates the submodular inequalities or by showing that it implies that there are more than $2^n - t$ inactive states. If this can be shown for all possible pairs of t-tuples (τ, τ_d), then it will imply that the optimal solution to **LP** cannot have exactly t active states.

These two key building blocks of the proof strategy are described next. In the sequel, we will always assume that the relays are arranged such that the l_i values are in descending order, i.e., $l_i > l_j$ for $i < j$. The r_i values will then be relatively ordered in one of the $n!$ possible ways. We will denote the ordering by π. We will refer to the indices of a set of primal variables as state indices and those of a set of dual variables as cut indices.

For any cut $\Lambda_j \subseteq [n]$, each term $C_{\Lambda_j}(\mathbf{p})$ is a convex combination of 2^n terms representing the interaction of the cut with the 2^n states m_i. We will denote the interaction term by

$$C_{\Lambda_j, m_i} = \left(\max_{k \in \bar{\Lambda}_j \cap L(m_i)} l_k + \max_{k \in \Lambda_j \cap T(m_i)} r_k \right) \tag{2.10}$$

$m_i \downarrow, \Lambda_j \rightarrow$	$\{\}$	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$\{L,L,L\}$	l_1	l_2	l_1	l_1	l_3	l_2	l_1	0
$\{T,L,L\}$	l_2	l_2+r_1	l_3	l_2	l_3+r_1	l_2+r_1	0	r_1
$\{L,T,L\}$	l_1	l_3	l_1+r_2	l_1	l_3+r_2	0	l_1+r_2	r_2
$\{L,L,T\}$	l_1	l_2	l_1	l_1+r_3	0	l_2+r_3	l_1+r_3	r_3
$\{T,T,L\}$	l_3	l_3+r_1	l_3+r_2	0	l_3+r_2	r_1	r_2	r_2
$\{T,L,T\}$	l_2	l_2+r_1	0	l_2+r_3	r_1	l_2+r_3	r_3	r_3
$\{L,T,T\}$	l_1	0	l_1+r_2	l_1+r_3	r_2	r_3	l_1+r_3	r_3
$\{T,T,T\}$	0	r_1	r_2	r_3	r_2	r_3	r_3	r_3

Table 2.2 – The example network has three relays with $l_1 > l_2 > l_3$ and $r_1 < r_2 < r_3$. The columns denote the relaying states $m_1, \ldots, m_8$ and the rows denote the cuts $\Lambda_1, \ldots, \Lambda_8$. The number in a particular cell is the interaction term between the state and cut corresponding to it.

As a running example, we will use a 3 relay network with $l_1 > l_2 > l_3$ and $r_1 < r_2 < r_3$. The relaying states, cuts and their interactions are shown in Table 2.2. The fact that $r_1 < r_2 < r_3$, denotes a specific ordering, which we will call π_0.

2.3.1 Submodularity Properties of C_j and C_i^d

A function f defined on the subsets of some finite universe U to $\mathbb{R}$ is called *submodular* if for all subsets $A, B \subseteq U$,

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B) \tag{2.11}$$

For a fixed $\mathbf{p}$, the quantity $C_{\Lambda_j}(\mathbf{p})$ is defined for each of the 2^n cuts Λ_j, which are themselves subsets of $[n]$. They naturally define a function on the subsets Λ_j of $[n]$. Crucially, these functions are not arbitrary and are in fact submodular. This is shown in the following lemma.

Lemma 2.3.1 *For any* $\Lambda_{j_1}, \Lambda_{j_2} \subseteq [n]$,

$$C_{\Lambda_{j_1}}(\mathbf{p}) + C_{\Lambda_{j_2}}(\mathbf{p}) \geq C_{\Lambda_{j_1} \cup \Lambda_{j_2}}(\mathbf{p}) + C_{\Lambda_{j_1} \cap \Lambda_{j_2}}(\mathbf{p})$$

Proof: For a fixed i, C_{Λ_j, m_i} is also a function defined on the subsets Λ_j. If we prove that submodularity holds for C_{Λ_j, m_i} for each $i \in [2^n]$, then our claim follows. We claim that the following holds for any two sets Λ_1' and Λ_2'

$$\max_{k \in \Lambda_1'} r_k + \max_{k \in \Lambda_2'} r_k \geq \max_{k \in \Lambda_1' \cup \Lambda_2'} r_k + \max_{k \in \Lambda_1' \cap \Lambda_2'} r_k \tag{2.12}$$

Let $x = \max_{k\in\Lambda'_1} r_k$ and $y = \max_{k\in\Lambda'_2} r_k$, then clearly $\max_{k\in\Lambda'_1\cup\Lambda'_2} r_k = \max(x, y)$. If $x \geq y$, then the claim is reduced to $\max_{k\in\Lambda'_1\cap\Lambda'_2} r_k \leq \max_{k\in\Lambda'_2} r_k$ which is also true as the r.h.s is a maximum over a superset. The case of $x < y$ is symmetrical. Now substituting $\Lambda'_1 = \Lambda_{j_1} \cap T(m_i)$ and $\Lambda'_2 = \Lambda_{j_2} \cap T(m_i)$, we get

$$\max_{k\in\Lambda_{j_1}\cap T(m_i)} r_k + \max_{k\in\Lambda_{j_2}\cap T(m_i)} r_k \geq \max_{k\in(\Lambda_{j_1}\cup\Lambda_{j_2})\cap T(m_i)} r_k + \max_{k\in(\Lambda_{j_1}\cap\Lambda_{j_2})\cap T(m_i)} r_k \tag{2.13}$$

Similarly, substituting $\Lambda'_1 = \overline{\Lambda}_{j_1} \cap L(m_i)$ and $\Lambda'_2 = \overline{\Lambda}_{j_2} \cap L(m_i)$ and replacing the r_k's with l_k's, we get

$$\max_{k\in\overline{\Lambda}_{j_1}\cap L(m_i)} l_k + \max_{k\in\overline{\Lambda}_{j_2}\cap L(m_i)} l_k \geq \max_{k\in\overline{(\Lambda_{j_1}\cup\Lambda_{j_2})}\cap L(m_i)} l_k + \max_{k\in\overline{(\Lambda_{j_1}\cap\Lambda_{j_2})}\cap L(m_i)} l_k \tag{2.14}$$

where we use the fact that $\overline{\Lambda_{j_1}\cap\Lambda_{j_2}} = \overline{\Lambda_{j_1}}\cup\overline{\Lambda_{j_2}}$ and $\overline{\Lambda_{j_1}\cup\Lambda_{j_2}} = \overline{\Lambda_{j_1}}\cap\overline{\Lambda_{j_2}}$. Adding the two inequalities (2.13) and (2.14), we get

$$C_{\Lambda_{j_1},m_i} + C_{\Lambda_{j_2},m_i} \geq C_{\Lambda_{j_1}\cup\Lambda_{j_2},m_i} + C_{\Lambda_{j_1}\cap\Lambda_{j_2},m_i} \tag{2.15}$$

from which the claim of the lemma follows. ■

Similar properties were also shown in [33]. Thus for each $j_1, j_2 \in [2^n]$ ($j_1 \neq j_2$), and $j_3 \in [2^n]$ such that $\Lambda_{j_3} = \Lambda_{j_1} \cup \Lambda_{j_2}$ and $j_4 \in [2^n]$ such that $\Lambda_{j_4} = \Lambda_{j_1} \cap \Lambda_{j_2}$, we have

$$C_{j_1} + C_{j_2} \geq C_{j_3} + C_{j_4} \tag{2.16}$$

Note that in our notation $C_{j_1} = C_{\Lambda_{j_1}}$. For a n-relay network, it can be shown that there are $2^{n-1}(2^n+1) - 3^n$ such submodular inequalities. We will put all distinct 4-tuples j_1, j_2, j_3, j_4 that represent a submodular inequality into the following set.

$$SM = \{(j_1, j_2, j_3, j_4) \text{ such that } C_{j_1} + C_{j_2} \geq C_{j_3} + C_{j_4}\} \tag{2.17}$$

The relative ordering of j_1, j_2 and j_3, j_4 is immaterial. Note that SM is a function of n only and does not depend on the l_i, r_i values or $\mathbf{p}$ and $\mathbf{p}^d$. Thus for our example 3-relay network, SM is the following

$$\begin{aligned} SM = \{&\{2,3,1,5\},\{2,4,1,6\},\{2,7,1,8\},\{3,4,1,7\},\{3,6,1,8\},\\ &\{4,5,1,8\},\{5,6,2,8\},\{5,7,3,8\},\{6,7,4,8\}\} \end{aligned} \tag{2.18}$$

Further, for the s-th submodular inequality involving the tuple $(j_1^s, j_2^s, j_3^s, j_4^s)$

$$C_{j_1^s} + C_{j_2^s} - C_{j_3^s} - C_{j_4^s} = \sum_{k=1}^{z_s} c_{\alpha_k^s} p_{\alpha_k^s} \tag{2.19}$$

for some strictly positive quantities $c_{\alpha_k^s}$ that depend only on the relative ordering π. Here z_s is the number of such terms. Thus, if $C_{j_1^s} = C_{j_2^s} = C_{j_3^s} = C_{j_4^s}$ for some value of $\mathbf{p}$, then each of the $p_{\alpha_k^s}$'s have to be identically equal to zero. We will denote this set of α_k^s for the s-th submodular inequality by $Z_s(\pi)$, i.e.,

$$Z_s(\pi) = \{\alpha_k^s \text{ such that } p_{\alpha_k^s} \equiv 0 \text{ when } C_{j_1^s} = C_{j_2^s} = C_{j_3^s} = C_{j_4^s}\} \tag{2.20}$$

In our 3-relay example, for the first submodular tuple $\{2, 3, 1, 5\}$, we have

$$C_2 + C_3 - C_1 - C_5 = (l_2 - l_3)p_1 + l_2 p_4 + r_1 p_5 + r_1 p_8 \tag{2.21}$$

Therefore, $Z_i(\pi_0) = \{1, 4, 5, 8\}$ as $l_2 > l_3$. Note that $Z_s(\pi)$ is defined only in terms of the equality of the terms in the s-th submodular inequality and depends only on π. It does not depend on the exact value of $\mathbf{p}$ but only requires the fact that $\mathbf{p} \geq 0$. The set of $Z_s(\pi)$ over all the submodular inequality tuples in SM is denoted by $Z_{SM}(\pi)$. For our 3-relay network, we have the following

$$\begin{aligned} Z_{SM}(\pi) = \{&\{1,4,5,8\},\{3,6,8\},\{1,3,4,5,6,8\},\{2,7,8\},\{1,2,4,5,7,8\} \\ &\{1,2,3,6,7,8\},\{1,2,7,8\},\{1,3,6\},\{1,4,5\}\} \end{aligned} \tag{2.22}$$

Due to the symmetry of **LP** and **DLP**, the above argument also holds for C_i^d, with an equivalence between Λ_j and $T(m_i)$ when $i = j$. The set of tuples in SM is the same and can equivalently be defined as

$$SM = \{(i_1, i_2, i_3, i_4) \text{ such that } C_{i_1}^d + C_{i_2}^d \geq C_{i_3}^d + C_{i_4}^d\} \tag{2.23}$$

We can also define $Z_s(\pi)$ for the s-th submodular inequality $(i_1^s, i_2^s, i_3^s, i_4^s)$ from SM as

$$Z_s(\pi) = \{\alpha_k^s \text{ such that } p_{\alpha_k^s}^d = 0 \text{ when } C_{i_1^s}^d = C_{i_2^s}^d = C_{i_3^s}^d = C_{i_4^s}^d\} \tag{2.24}$$

where

$$C_{i_1^s}^d + C_{i_2^s}^d - C_{i_3^s}^d - C_{i_4^s}^d = \sum_{k=1}^{z_s} c_{\alpha_k^s}^d p_{\alpha_k^s}^d \tag{2.25}$$

for some strictly positive quantities $c_{\alpha_k^s}^d$.

2.3.2 Linear Programming Duality

Corresponding to the optimal solution $\tilde{\mathbf{p}}$ of **LP**, there is also an optimal dual solution $\tilde{\mathbf{p}}^d$ of **DLP**. Assume that this is also basic feasible. This means, at least $2^n + 1$ of the constraints must be satisfied with equality for $\tilde{\mathbf{p}}$. Here, we will make a mild assumption of *non-degeneracy* in the linear programs we are considering. This means that every basic

feasible solution is defined by *exactly* $2^n + 1$ *tight* constraints. If $\tilde{\mathbf{p}}$ has t active states then exactly t of the flow constraints are satisfied with equality and the remaining $2^n - t$ flow constraints are *strict* inequalities. By the condition of complementary slackness, this implies that $\tilde{\mathbf{p}}^d$ has exactly t active dual states (which correspond to the tight flow constraints in **LP**), **DLP** has exactly t tight dual flow constraints (corresponding to the active states in **LP**) and the remaining $2^n - t$ dual flow constraints are strict inequalities.

This basic fact from LP duality will be used extensively in our proof strategy. To summarize, for each pair (τ, τ_d) of t-tuples representing the active states in the primal and dual optimum, the above discussion implies that

$$C_j(\tilde{\mathbf{p}}) = C_{lp}^n \text{ for } j \in \tau_d \text{ and } C_i^d(\tilde{\mathbf{p}}^d) = C_{lp}^n \text{ for } i \in \tau \tag{2.26}$$

$$C_j(\tilde{\mathbf{p}}) > C_{lp}^n \text{ for } j \in [2^n] \setminus \tau_d \text{ and } C_i^d(\tilde{\mathbf{p}}^d) < C_{lp}^n \text{ for } i \in [2^n] \setminus \tau \tag{2.27}$$

In our proof strategy, we will start from a pair (τ, τ_d) of t-tuples and try to derive a contradiction through several methods as described in the next two sections.

2.4 Contradictions from Submodular Inequalities and LP Duality

The contradictions derived from the submodular inequalities can be of 3 types.

2.4.1 Type I Contradiction

From (2.26) and (2.27), each of $\{C_j, j \in \tau_d\}$ is equal to C_{lp}^n and each of $\{C_j, j \in [2^n] \setminus \tau_d\}$ is *strictly greater* than C_{lp}^n. However, the submodular relationships among the C_j's may lead to contradictions. Let the k-th submodular 4-tuple SM_k from the set SM be $\{k_1, k_2, k_3, k_4\}$. Then, we have a contradiction if there exists a $k \in [|SM|]$ and the following holds

$$|\{k_1, k_2\} \cap \tau_d| = 2 \text{ and } |\{k_3, k_4\} \cap ([2^n] \setminus \tau_d)| \geq 1 \tag{2.28}$$

This is because if $|\{k_3, k_4\} \cap ([2^n] \setminus \tau_d)| \geq 1$, then $C_{k_3} + C_{k_4} > 2C_{lp}^n$ and $C_{k_1} + C_{k_2} = 2C_{lp}^n$, while the submodular inequality implies $C_{k_1} + C_{k_2} \geq C_{k_3} + C_{k_4}$. In our 3-relay example, if $\tau_d = \{1, 2, 3, 4, 6\}$, we have $C_2 = C_3 = C_1 = C_{lp}^3$ and $C_5 > C_{lp}^3$. We get a contradiction for the first submodular tuple in SM which is $\{2, 3, 1, 5\}$. On the other hand, for $\tau_d = \{1, 2, 3, 5, 8\}$, we cannot derive any contradictions of Type I.

Similarly, it is also true that each of $\{C_i^d, i \in \tau\}$ are equal to C_{lp}^n and each of $\{C_i^d, i \in [2^n] \setminus \tau\}$ are *strictly smaller* than C_{lp}^n. Then, we have a contradiction if there exists a

$k \in [|SM|]$ and the following holds

$$|\{k_3, k_4\} \cap \tau| = 2 \text{ and } |\{k_1, k_2\} \cap ([2^n] \setminus \tau)| \geq 1 \tag{2.29}$$

In this case, $C_{k_1} + C_{k_2} < 2C_{lp}^n$ and $C_{k_3} + C_{k_4} = 2C_{lp}^n$, while the submodular inequality implies $C_{k_1} + C_{k_2} \geq C_{k_3} + C_{k_4}$. In our 3-relay example, if $\tau = \{1, 2, 4, 5, 6\}$, we have $C_1^d = C_2^d = C_5^d = C_{lp}^3$ and $C_3^d < C_{lp}^3$. However, we get a contradiction for the first submodular tuple in SM which is $\{2, 3, 1, 5\}$. On the other hand, for $\tau = \{1, 2, 3, 4, 5\}$, we cannot derive any contradictions of Type I.

2.4.2 Type II Contradiction

The fact that $\{C_j, j \in \tau_d\}$ are all equal, forces some of the primal variables to be zero via the submodular inequalities. The state indices that are forced to be zero, denoted by $FZ(\tau_d, \pi)$ (FZ stands for *forced zeros*), is precisely the following

$$FZ(\tau_d, \pi) = \cup_{k \in [|SM|]} \{Z_k(\pi) \text{ such that } |SM_k \cap \tau_d| = 4\} \tag{2.30}$$

Similarly, the equality of $\{C_i^d, i \in \tau\}$ forces some dual variables to be zero. These cut indices are denoted by $FZ_d(\tau, \pi)$ and is given by

$$FZ_d(\tau, \pi) = \cup_{k \in [|SM|]} \{Z_k(\pi) \text{ such that } |SM_k \cap \tau| = 4\} \tag{2.31}$$

If $|FZ(\tau_d, \pi)| > 2^n - t$ or $|FZ_d(\tau, \pi)| > 2^n - t$, we have a contradiction because we started off with t active states.

For our 3-relay example, for $\tau_d = \{1, 2, 3, 5, 8\}$, we get $FZ(\tau_d, \pi_0) = \{1, 4, 5, 8\}$, in which case we have a contradiction. For $\tau_d = \{1, 2, 4, 6, 8\}$, we get $FZ(\tau_d, \pi_0) = \{3, 6, 8\}$, in which case we do not have a contradiction. Similarly, for $\tau = \{1, 2, 3, 4, 5\}$, we get $FZ_d(\tau, \pi_0) = \{1, 4, 5, 8\}$ in which case we also have a contradiction. For $\tau = \{1, 2, 3, 4, 6\}$, we get $FZ_d(\tau, \pi_0) = \{3, 6, 8\}$, in which case we do not have a contradiction.

2.4.3 Type III Contradiction

The equality of $\{C_j, j \in \tau_d\}$ combined with the submodular inequalities can sometimes *imply* all the remaining flow constraints *automatically*, irrespective of the value of $\mathbf{p}$ (of course assuming $\mathbf{p}$ is non-negative). Let $I_d(\tau_d, \pi)$ denote the set of cut indices j such that the condition $C_j \geq C_{lp}^n$ is implied if we assume $C_j = C_{lp}^n$ for $j \in \tau_d$. Trivially, all the indices in τ_d can be included in $I_d(\tau_d, \pi)$. For the k-th tuple $\{k_1, k_2, k_3, k_4\}$ of SM, the following is true

$$|\{k_3, k_4\} \cap I_d(\tau_d, \pi)| = 2 \text{ and } (k_1 \in \tau_d) \Rightarrow k_2 \in I_d(\tau_d, \pi) \tag{2.32}$$

$$|\{k_3, k_4\} \cap I_d(\tau_d, \pi)| = 2 \text{ and } (k_2 \in \tau_d) \Rightarrow k_1 \in I_d(\tau_d, \pi) \tag{2.33}$$

This is because the first condition implies that $C_{k_3} + C_{k_4} \geq 2C_{lp}^n$ and if $C_{k_1} = C_{lp}^n$, then $C_{k_2} \geq C_{k_3} + C_{k_4} - C_{lp}^n \geq C_{lp}^n$. Similarly, if $C_{k_2} = C_{lp}^n$, then $C_{k_1} \geq C_{k_3} + C_{k_4} - C_{lp}^n \geq C_{lp}^n$. The same is true for all the tuples in SM. This process can be repeated over all the tuples in SM until the size of $I_d(\tau_d, \pi)$ does not increase. At the end of this process, if $|I_d(\tau_d, \pi)| = 2^n$, then this implies that if all the $\{C_j, j \in \tau_d\}$ are equal to C_{lp}^n, then the remaining flow constraints are satisfied automatically.

The discussion from the previous subsection on Type II contradictions also implies that the states corresponding to $FZ(\tau_d, \pi)$ are zero. Putting these together, **LP** can equivalently be replaced by the following linear program.

Maximize C

$$C_j(\mathbf{p}) = C \text{ for each } j \in \tau_d \tag{2.34}$$

$$\sum_{i \in [2^n] \setminus FZ(\tau_d, \pi)} p_i = 1; \sum_{i \in FZ(\tau_d, \pi)} p_i = 0 \tag{2.35}$$

$$\forall i, p_i \geq 0, C \geq 0 \tag{2.36}$$

In this case, the constraints of the linear program are simply a set of equations. Let the rank of this system of equations be $R(\tau_d, \pi)$. If $R(\tau_d, \pi)$ turns out to be less than $t + 1$, then the constraints can be further reduced to a set of $R(\tau_d, \pi)$ equations. Again by the theory of linear programming, there is a basic feasible solution that is optimal. Since there are $R(\tau_d, \pi)$ equations and $t + 1$ variables, at least one state variable (C is always non-zero in the optimal solution) is zero in the basic feasible solution. But this again contradicts our assumption that there are exactly t active states in the optimal solution to **LP**.

In our 3-relay example, for $\tau_d = \{1, 2, 4, 6, 8\}$, $FZ(\tau_d, \pi_0) = \{3, 6, 8\}$ and in fact $|I_d(\tau_d, \pi_0)| = 2^3 = 8$. The reduced form of **LP** is the following

Maximize C

$$l_1 p_1 + l_2 p_2 + l_1 p_4 + l_3 p_5 + l_1 p_7 = C$$
$$l_2 p_1 + (l_2 + r_1) p_2 + l_2 p_4 + (l_3 + r_1) p_5 = C$$
$$l_1 p_1 + l_2 p_2 + (l_1 + r_3) p_4 + (l_1 + r_3) p_7 = C$$
$$l_2 p_1 + (l_2 + r_1) p_2 + (l_2 + r_3) p_4 + r_1 p_5 + r_3 p_7 = C$$
$$r_1 p_2 + r_3 p_4 + r_2 p_5 + r_3 p_7 = C$$
$$p_1 + p_2 + p_4 + p_5 + p_7 = 1$$
$$p_1, p_2, p_4, p_5, p_7, C \geq 0$$

As the sum of the second and third constraints is equal to the sum of the first and fourth

constraints, the rank of this system is 5, which is less than $t + 1 = 6$. We thus have a contradiction of Type III.

Similarly, the equality of $\{C_i^d, i \in \tau\}$ combined with the submodular inequalities may imply all the remaining flow constraints automatically. Let $I(\tau, \pi)$ denote the set of state indices i such that the condition $C_i^d \leq C_{lp}^n$ is implied if we assume $C_i^d = C_{lp}^n$ for $i \in \tau$. Trivially, all the indices in τ can be included in $I(\tau, \pi)$. For the k-the tuple of SM, the following is true

$$|\{k_1, k_2\} \cap I(\tau, \pi)| = 2 \text{ and } (k_3 \in \tau) \implies k_4 \in I(\tau, \pi) \tag{2.37}$$

$$|\{k_1, k_2\} \cap I(\tau, \pi)| = 2 \text{ and } (k_4 \in \tau) \implies k_3 \in I(\tau, \pi) \tag{2.38}$$

The same is true for all the tuples in SM. This process can be repeated over all the tuples in SM until the size of $I(\tau, \pi)$ does not increase. At the end, if $|I(\tau, \pi)| = 2^n$, then this implies that if all the $\{C_i^d, i \in \tau\}$ are equal to C_{lp}^n, then the remaining dual flow constraints are satisfied automatically. The remaining steps for deriving a contradiction are similar to the ones described above, with a contradiction arising when $R_d(\tau, \pi)$, the rank of the reduced **DLP** system, is less than $t + 1$.

2.4.4 Remarks

The three ways to derive contradictions developed so far can be implemented in a parallel manner for a set of tight primal constraints τ_d and a set of tight dual constraints τ as they do not *interact* in any way. This is crucial for an efficient implementation of the proof strategy, as discussed in Section 2.5. However, these techniques by themselves do not lead to the optimal bounds on the number of active states in **LP**. Two additional techniques for deriving contradictions using LP duality are required, as discussed next.

We again start from a pair (τ, τ_d) of indices, each of size t that correspond to the active states in the primal and dual optimal solutions, respectively. We derive two more types of contradictions from the interaction of τ and τ_d.

2.4.5 Type IV Contradiction

From the complementary slackness conditions of linear programming, it follows that a tight primal constraint corresponds to a non-zero dual variable and a tight dual constraint corresponds to a non-zero primal variable. From Section 2.4.2, the submodular inequalities imply that if $\{C_j, j \in \tau_d\}$ are all equal then all $\{p_i, i \in FZ(\tau_d, \pi)\}$ must be zero and if $\{C_i^d, i \in \tau\}$ are all equal then all $\{p_j^d, j \in FZ_d(\tau, \pi)\}$ must be zero. Thus, we have a contradiction if the following condition holds.

$$|FZ(\tau_d, \pi) \cap \tau| > 0 \text{ or } |FZ_d(\tau, \pi) \cap \tau_d| > 0 \tag{2.39}$$

2.4.6 Type V Contradiction

It is possible that all the above methods fail to discover a contradiction. In that case, we go back to the original assumptions about τ and τ_d. First, for two linear combinations $D_1(\mathbf{p}), D_2(\mathbf{p})$ of $\mathbf{p} = (p_1, \ldots, p_{2^n})$, the relational operator $\succeq$ is defined as follows. $D_1(\mathbf{p}) \succeq D_2(\mathbf{p})$ iff

$$D_1(\mathbf{p}) - D_2(\mathbf{p}) = \sum_{i=1}^{2^n} \alpha_i p_i \text{ where } \alpha_i \geq 0 \text{ for } i \in [2^n] \tag{2.40}$$

Let $C_{j|\tau}(\mathbf{p})$ denote the value of $C_j(\mathbf{p})$ with only the active states of $\mathbf{p}$ implied by τ retained. That is, for $\mathbf{p}' = (p'_1, \ldots, p'_{2^n})$,

$$C_{j|\tau}(\mathbf{p}) = C_j(\mathbf{p}') \text{ where } p'_i = p_i \text{ for } i \in \tau \text{ and 0 otherwise} \tag{2.41}$$

Similarly, we define $C^d_{i|\tau_d}(\mathbf{p}^d)$ for $\mathbf{p}'^d = (p'^d_1, \ldots, p'^d_{2^n})$ as follows

$$C^d_{i|\tau_d}(\mathbf{p}^d) = C^d_i(\mathbf{p}'^d) \text{ where } p'^d_j = p^d_j \text{ for } j \in \tau_d \text{ and 0 otherwise} \tag{2.42}$$

Note that both $C_{j|\tau}(\mathbf{p})$ and $C^d_{i|\tau_d}(\mathbf{p}^d)$ should be seen as linear functions of t variables. By Eq. (2.27), we have a contradiction if the following is true

$$\exists(j_1 \in \tau_d, j_2 \in [2^n] \setminus \tau_d) \text{ s.t. } C_{j_1|\tau}(\mathbf{p}) \succeq C_{j_2|\tau}(\mathbf{p}) \tag{2.43}$$

Similarly, we have a contradiction if the following is true

$$\exists(i_1 \in \tau, i_2 \in [2^n] \setminus \tau) \text{ s.t. } C^d_{i_2|\tau_d}(\mathbf{p}^d) \succeq C^d_{i_1|\tau_d}(\mathbf{p}^d) \tag{2.44}$$

Finally, we also have a contradiction if the following is true for the primal constraints

$$\exists(j_1, j_2 \in \tau_d, j_1 \neq j_2) \text{ s.t. } C_{j_1|\tau}(\mathbf{p}) \succeq C_{j_2|\tau}(\mathbf{p}) \tag{2.45}$$

In this case, if $C_{j_1|\tau}(\mathbf{p}) - C_{j_2|\tau}(\mathbf{p})$ is not identically zero, each $p_i, i \in \tau$ appearing in the expression must be zero, as both the terms are equal to C^n_{lp}. On the other hand, if the difference is identically zero, then the rank of the constraint matrix is less than $t+1$ and there is a contradiction using reasoning similar to the one outlined in Section 2.4.3. Similarly, we have a contradiction if the following is true for the dual constraints.

$$\exists(i_1, i_2 \in \tau, i_1 \neq i_2) \text{ s.t. } C_{i_1|\tau_d}(\mathbf{p}^d) \succeq C_{i_2|\tau_d}(\mathbf{p}^d) \tag{2.46}$$

Note that to check whether an expression of the type $C_{j_1|\tau}(\mathbf{p}) \succeq C_{j_2|\tau}(\mathbf{p})$ holds, it is enough to specify the ordering π. As before, only the fact that $\mathbf{p}$ is non-negative is required, not its exact value.

The various terms defined in this section are summarized in Table 2.3.

τ, τ_d	The set of state and cut indices that are active in the optimal solution
SM	Set of 4-tuples defining the submodular inequalities among C_j or C_i^d
$Z_s(\pi)$	Set of state or cut indices that are inactive if all terms in s-th submodular tuple are equal
$FZ(\tau_d, \pi)$	Set of state indices that are inactive if $C_j = C_{lp}^n$ for $j \in \tau_d$
$FZ_d(\tau, \pi)$	Set of cut indices that are inactive if $C_i^d = C_{lp}^n$ for $i \in \tau$
$I_d(\tau_d, \pi)$	Set of cut indices such that the corresponding flow constraints in **LP** are implied if $C_j = C_{lp}^n$ for $j \in \tau_d$
$R(\tau_d, \pi)$	Rank of the reduced form of **LP** if $\lvert I_d(\tau_d, \pi)\rvert = 2^n$
$I(\tau, \pi)$	Set of state indices such that the corresponding flow constraints in **LP** are implied if $C_i^d = C_{lp}^n$ for $i \in \tau$
$R_d(\tau, \pi)$	Rank of the reduced form of **DLP** if $\lvert I(\tau_d, \pi)\rvert = 2^n$
$C_{j\lvert\tau}(\mathbf{p})$	The expression $C_j(\mathbf{p})$ with $p_i, i \notin \tau$ set to zero
$C_{i\lvert\tau_d}^d(\mathbf{p}^d)$	The expression $C_i(\mathbf{p}^d)$ with $p_j^d, j \notin \tau_d$ set to zero

Table 2.3 – Summary of terms defined in this section related to different types of contradictions.

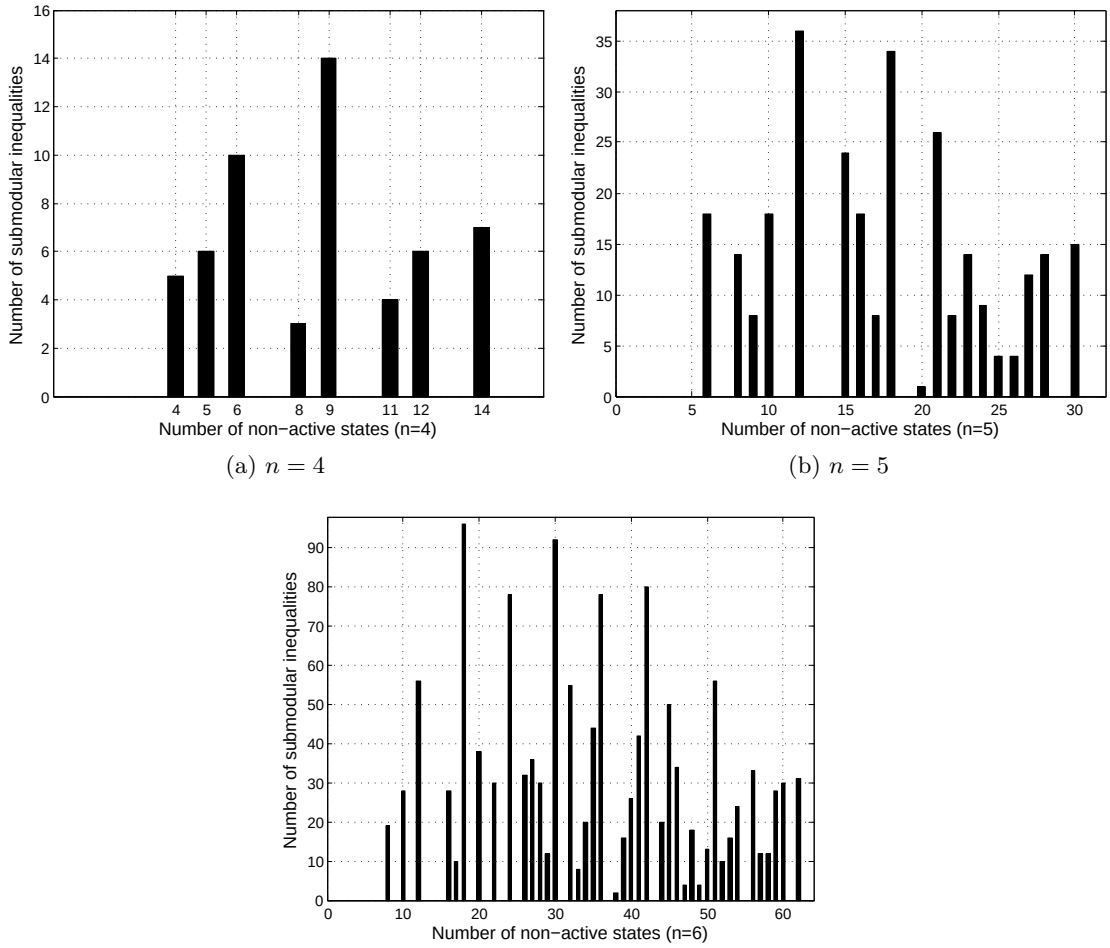
2.5 Proof of Theorem 2.2.2

The proof strategy described in the previous two sections was implemented using symbolic algebra in `Mathematica` for $n = 3, 4, 5, 6$. To minimize computation time, we implement the proof strategy in two steps.

2.5.1 Proof Implementation: First Stage

In the first stage we will use contradictions of Type I, II and III only for tuples τ_d of active dual states in the optimum to derive upper bounds on the number of active states in **LP**. The relays are ordered such that $l_i > l_j$ for $i < j$. We fix the ordering of the r_i's, where the specific ordering is denoted by π. Suppose **LP** has exactly $t \in [n+2, 2^n]$ active states. This means there is a t-tuple τ_d of active dual states or equivalently tight primal constraints. For a given n, we first compute the set of submodular inequalities SM. At this stage, for each of the possible $\binom{2^n}{t}$ t-tuples, we check whether there is a Type I contradiction. Rather than first generating all the $\binom{2^n}{t}$ tuples and then checking for contradictions, we only generate the t-tuples that do not have a Type I contradiction using a recursive procedure. This results in greater efficiency. The set of τ_d that do not have a Type I contradiction is denoted by $S_{d,1}(n, t)$.

Since the terms in **LP** are of the form $\max_{k \in \Lambda_j \cap L(m_i)} l_k + \max_{k \in \Lambda_j \cap T(m_i)} r_k$, fixing π fixes these interaction terms as a function of the r_i's (in the sense that the max is resolved). This



(a) $n = 4$

(b) $n = 5$

(c) $n = 6$

Figure 2.2 – Histogram of the cardinalities of the sets in $Z_{SM}(\pi)$ for $n = 4, 5, 6$.

also implies we can compute the sets in $Z_{SM}(\pi)$. For each $\tau_d \in S_{d,1}(n,t)$, we can then compute the sets $FZ(\tau_d, \pi), I_d(\tau_d, \pi), R(\tau_d, \pi)$. This enables us to look for Type II and Type III contradictions for each tuple. The set of τ_d that do not have a Type II and Type III contradiction is denoted by $S_{d,2}(n,t,\pi)$ and $S_{d,3}(n,t,\pi)$ respectively. This computation is done for all the possible $n!$ orderings π.

A histogram of the cardinalities of the sets in $Z_{SM}(\pi)$ is shown in Fig. 2.2 for the configuration where $r_i < r_j$ for $i < j$. We observe in our results that the histograms are the same for all orderings π. The cardinalities of the sets $S_{d,1}(n,t)$, $\bar{S}_{d,2}(n,t)$ and $\bar{S}_{d,3}(n,t)$ for $n = 3, 4, 5, 6$ and for different values of $t \in [n+2, 2^n]$ are shown below. $\bar{S}_{d,2}(n,t)$ and $\bar{S}_{d,3}(n,t)$ denotes the average value of $S_{d,2}(n,t,\pi)$ and $S_{d,3}(n,t,\pi)$ over all the $n!$ orderings π (rounded to the nearest integer), respectively. Numbers are only

shown for those t where $|S_{d,1}(n,t)|$ is not zero.

$t \to$	5	6	8
$\lvert S_{d,1}(3,t)\rvert$	6	6	1
$\lvert \bar{S}_{d,2}(3,t)\rvert$	4	0	0
$\lvert \bar{S}_{d,3}(3,t)\rvert$	0	0	0

$t \to$	6	7	8	9	10	12	16
$\lvert S_{d,1}(4,t)\rvert$	120	54	62	20	24	12	1
$\lvert \bar{S}_{d,2}(4,t)\rvert$	64	32	0	0	0	0	0
$\lvert \bar{S}_{d,3}(4,t)\rvert$	28	0	0	0	0	0	0

$t \to$	9	10	11	12	13	14	15	16	17	18	20	24	32
$\lvert S_{d,1}(5,t)\rvert$	1000	1140	500	780	240	390	120	200	10	100	60	20	1
$\lvert \bar{S}_{d,2}(5,t)\rvert$	469	30	0	0	0	0	0	0	0	0	0	0	0
$\lvert \bar{S}_{d,3}(5,t)\rvert$	106	0	0	0	0	0	0	0	0	0	0	0	0

$t \to$	17	18	19	20	21	22	23	24	25	26
$\lvert S_{d,1}(6,t)\rvert$	8550	12990	4110	8700	3420	4800	720	4140	600	1680
$\lvert \bar{S}_{d,2}(6,t)\rvert$	19	8	0	0	0	0	0	0	0	0
$\lvert \bar{S}_{d,3}(6,t)\rvert$	19	0	0	0	0	0	0	0	0	0

$t \to$	27	28	30	32	33	34	36	40	48	64
$\lvert S_{d,1}(6,t)\rvert$	360	1530	720	507	12	60	300	12	30	1
$\lvert \bar{S}_{d,2}(6,t)\rvert$	0	0	0	0	0	0	0	0	0	0
$\lvert \bar{S}_{d,3}(6,t)\rvert$	0	0	0	0	0	0	0	0	0	0

The minimum t^* such that for all $t > t^*$, $S_{d,3}(n,t,\pi)$ is an empty set for all the $n!$ orderings π is an upper bound to the number of active states in **LP**. Thus, we obtain the following lemma.

Lemma 2.5.1 *There exist optimal solutions to* **LP** *for* $n = 3, 4, 5$ *and* 6 *relays such that it has at most* $u_1(n)$ *active states, where*

n	3	4	5	6
$u_1(n)$	4	6	9	17

This already proves the optimal upper bound for $n = 3$ and upper bounds that are much better than the trivial 2^n for $n = 4, 5, 6$.

2.5.2 Proof Implementation: Second Stage

Our starting point will be the upper bounds shown above. Note that this stage is only implemented for $n = 4, 5, 6$. Suppose **LP** has exactly $t \in [n+2, u_1(n)]$ active states. This means there is a t-tuple τ_d of active dual states and a t-tuple τ of active primal states.

For a given n, we first compute the set of submodular inequalities SM. At this stage, for each of the possible $\binom{2^n}{t}$ τ_d and τ, we check whether there is a Type I contradiction. As above, only the tuples that do not have a contradiction are generated using a recursive procedure. The set of τ_d and τ that do not have a Type I contradiction is denoted by $S_{d,1}(n,t)$ and $S_{p,1}(n,t)$ respectively.

For a given ordering π, for each $\tau_d \in S_{d,1}(n,t)$, we can then compute the sets $FZ(\tau_d,\pi)$, $I_d(\tau_d,\pi)$, $R(\tau_d,\pi)$ and for each $\tau \in S_{p,1}(n,t)$, we can compute the sets $FZ_d(\tau,\pi)$, $I(\tau,\pi)$, $R_d(\tau,\pi)$. This enables us to look for Type II and Type III contradictions for each tuple. The set of τ_d that do not have a Type II and Type III contradiction is denoted by $S_{d,2}(n,t,\pi)$ and $S_{d,3}(n,t,\pi)$ respectively. The set of τ that do not have a Type II and Type III contradiction is denoted by $S_{p,2}(n,t,\pi)$ and $S_{p,3}(n,t,\pi)$ respectively

The computations for the primal and dual tuples can be carried out in parallel till this stage. Next, we check each pair of tuples from the product set $S_{p,3}(n,t,\pi) \times S_{d,3}(n,t,\pi)$ for Type IV contradictions. The ones that survive are checked for Type V contradictions. This requires the computation of $C_{j|\tau}(\mathbf{p})$ and $C^d_{i|\tau_d}(\mathbf{p}^d)$ as described in Section 2.4. Let the set of t-tuple pairs that survive all the stages be $S_{p,d}(n,t,\pi)$. We repeat this computation for all the orderings π. We now state the proof of Thm. 2.2.2.

Proof of Theorem 2.2.2: The proof strategy was implemented for $n \in \{4,5,6\}$, for each $t \in [n+2, u_1(n)]$ and for each of the $n!$ relative orderings of the r_i's. Below we report the cardinalities of the sets $S_{p,1}(n,t), S_{d,1}(n,t)$. We also report $\bar{S}_{d,2}(n,t)$, $\bar{S}_{p,2}(n,t)$, $\bar{S}_{d,3}(n,t)$ and $\bar{S}_{p,3}(n,t)$ which are the average of the cardinalities of the sets $S_{d,2}(n,t)$, $S_{p,2}(n,t)$, $S_{d,3}(n,t)$ and $S_{p,3}(n,t)$ over the $n!$ orderings π, respectively (rounded to the nearest integer). Finally, the quantity $\bar{S}_{p,d}(n,t)$, which is $S_{p,d}(n,t,\pi)$ averaged over all π, is reported.

For $n = 4$ and $n = 5$, we have

$t \rightarrow$	6	7
$\lvert S_{d,1}(4,t)\rvert$	120	54
$\lvert S_{p,1}(4,t)\rvert$	707	598
$\lvert \bar{S}_{d,2}(4,t)\rvert$	64	32
$\lvert \bar{S}_{p,2}(4,t)\rvert$	692	535
$\lvert \bar{S}_{d,3}(4,t)\rvert$	28	0
$\lvert \bar{S}_{p,3}(4,t)\rvert$	686	524
$\lvert \bar{S}_{p,d,5}(4,t)\rvert$	0	0

$t \rightarrow$	7	8	9
$\lvert S_{d,1}(5,t)\rvert$	1320	1535	1000
$\lvert S_{p,1}(5,t)\rvert$	143900	232570	320920
$\lvert \bar{S}_{d,2}(5,t)\rvert$	1164	803	469
$\lvert \bar{S}_{p,2}(5,t)\rvert$	143881	232285	316400
$\lvert \bar{S}_{d,3}(5,t)\rvert$	924	353	106
$\lvert \bar{S}_{p,3}(5,t)\rvert$	143871	232240	316310
$\lvert \bar{S}_{p,d,5}(5,t)\rvert$	0	0	0

For $n = 6$ we have.

$t \rightarrow$	8	9	10	11	12	13	14	15
$\lvert S_{d,1}(6,t)\rvert$	26620	25060	30270	22980	29040	18240	23490	14060
$\lvert S_{p,1}(6,t)\rvert(\times 10^8)$	1.1	3.4	9.1	21	44	8.0	13	20
$\lvert \bar{S}_{d,2}(6,t)\rvert$	21745	20392	18700	13377	8288	4275	1125	498
$\lvert \bar{S}_{p,2}(6,t)\rvert(\times 10^8)$	1.1	3.4	9.1	21	44	8.0	13	20
$\lvert \bar{S}_{d,3}(6,t)\rvert$	19945	16432	12274	5765	3608	1071	498	0
$\lvert \bar{S}_{p,3}(6,t)\rvert(\times 10^8)$	1.1	3.4	9.1	21	44	8.0	13	20
$\lvert S'_{p,d,5}(6,t)\rvert$	0	0	0	0	0	0	0	0

$t \rightarrow$	16	17
$\lvert S_{d,1}(6,t)\rvert$	17345	8550
$\lvert S_{p,1}(6,t)\rvert(\times 10^8)$	34	51
$\lvert \bar{S}_{d,2}(6,t)\rvert$	11	19
$\lvert \bar{S}_{p,2}(6,t)\rvert(\times 10^8)$	34	51
$\lvert \bar{S}_{d,3}(6,t)\rvert$	11	19
$\lvert \bar{S}_{p,3}(6,t)\rvert(\times 10^8)$	34	51
$\lvert S'_{p,d,5}(6,t)\rvert$	0	0

The cardinalities of the sets $S_{p,\tau_d,5}(n,t,\pi)$ for $n = 4,5,6$ and $t \in \{n+2,\ldots,u_1(n)\}$ are all 0, which means that any pair of basic feasible optimum solutions of **LP** and **DLP** with more than $n+1$ active states leads to a contradiction. Therefore, for $n = 4,5,6$, the optimal solution to **LP** must have atmost $n+1$ states. For $n = 3$, the optimal bound was obtained in Lemma 2.5.1 above and the result for $n = 2$ follows from the work of Bagheri et. al. [31]. ■

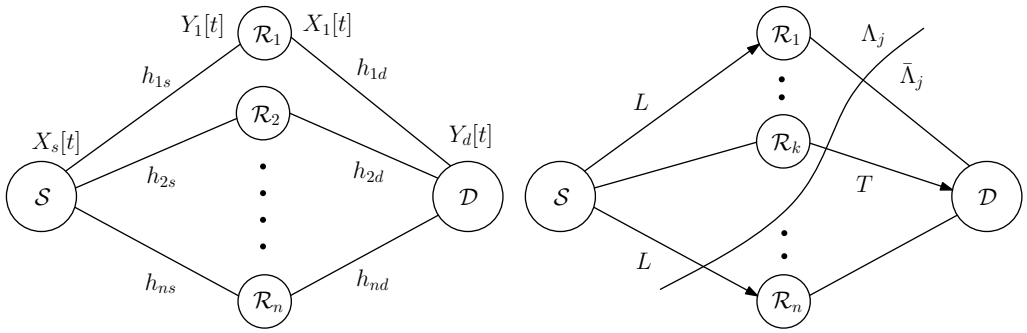
3 Routing Strategies for Half-Duplex Diamond Networks

In the previous chapter, we showed that it is possible to achieve rates close to the capacity of a n-relay half-duplex diamond network by using only $n+1$ active states in the schedule (for $n \leq 6$), instead of the possible 2^n possible states. Even with the exponentially reduced complexity of scheduling, it may still be difficult to operate the network using these states because it may require the cooperation of all the n relays. In addition, computation of these states may itself be a hard problem.

In this chapter, we show that if we are willing to sacrifice a bit on the rates, we can achieve significant reductions in operational and computation complexity. Following the network simplification approach of Nazaroglu et. al. [10], we show that very simple *routing strategies* that use only two relays and two relaying states can achieve rates at least half of the capacity of the whole network (approximately). For 2-relay networks, we show that routing strategies achieve at least 8/9 of the capacity (approximately). The relaying states employ only point-to-point connections and do not use any broadcasting or multiple access.

We use the same linear programming approximations to capacity used in Chapter 2. We use feasible solutions to the dual program to obtain upper bounds to the approximate capacity expressions, and prove that there exist two relays which, when operated using the simple 2-state schedule, can achieve a rate at least equal to half the capacity upper bound. We also show that such a pair can be found in $O(n \log n)$ time.

If we are willing to sacrifice on the optimality of schedules, rather than designing a relaying strategy by solving an optimization problem to find the optimal schedule and operating the network by switching between these states, through our routing strategies we can achieve approximately at least half of the capacity of the network by using only a pair of relays and two scheduling states. This leads to substantial savings in complexity, especially in time-varying networks where we may often need to redesign and readapt the relay operations. An additional benefit is that we only need to consume power for two relays, instead of n.



(a) The Gaussian n-relay half-duplex diamond network.

(b) Relaying states and cuts in the network.

Figure 3.1 – Network model with channel coefficients of the individual links, relaying states and cuts. For a particular relaying state, an arrow on a link denotes that it is active.

3.1 Related Work

The results on routing strategies using only a few relays and states adds to the recent literature of network simplification for diamond relay networks. Nazaroglu et. al. [10] showed that in full-duplex diamond networks using two relays enables us to achieve approximately 2/3 of the network capacity; this result, translated to half-duplex networks, implies that using two relays and 2-state schedules enables us to achieve 1/3 of the capacity. In this work, we improve this fraction to 1/2, still using two states and notably avoiding the broadcast and multiple access links. The results in [10] were extended to a special class of diamond networks with 2 receive and transmit antennas at the source and destination in [34].

In terms of literature on relay selection in wireless networks, previous work has mainly addressed full duplex relays. The work of Tannious et. al. [35] and Bletsas et. al. [36] propose algorithms to select the *single best* relay (in terms of cooperative diversity) in diamond networks. The work of Cai et. al. [37] and Zhao et. al. [38] analyse the performance of heuristics for selecting a subset of relays for Amplify-and-Forward (AF) based protocols in diamond networks. More recent work by Agnihotri at. al. [39] proves upper bounds on multiplicative and additive gaps for AF-based relay selection.

3.2 Problem Formulation and Main Results

3.2.1 Network Model

We consider the Gaussian n-relay diamond network where a source $\mathcal{S}$ transmits information to a destination $\mathcal{D}$ with the help of half-duplex relays. The notations and equations

describing the signals are the same as Chapter 2. A representative figure is shown in Fig. 3.1a.

We can then calculate the individual link capacities from $\mathcal{S}$ to $\mathcal{R}_k$ (l_k) and from $\mathcal{R}_k$ to $\mathcal{D}$ (r_k) as

$$l_k = \log(1 + |h_{ks}|^2 P), \quad r_k = \log(1 + |h_{kd}|^2 P) \tag{3.1}$$

Let $[n]$ represent the set $\{1, 2, \cdots, n\}$. For $i \in [2^n]$, let $m_i \in M = \{L, T\}^n$ be a distinct relaying state, i.e., a particular configuration of listening and transmitting states for all the relays. The fraction of time the relays spend in state m_i will be denoted by p_i, where $\sum_{i\in[2^n]} p_i = 1$. We will use $L(m_i), T(m_i) \subseteq [n]$ to denote the set of indices of the relays in listening and transmitting state in m_i, respectively. Also, for $j \in [2^n]$, $\Lambda_j \subseteq [n]$ denotes the cut separating $\mathcal{S} \cup (\cup_{k\in\Lambda_j} \mathcal{R}_k)$ from $\mathcal{D} \cup (\cup_{k\in\bar{\Lambda}_j} \mathcal{R}_k)$. A representative cut is shown in Fig. 3.1b

To keep the exposition simple, we will assume that the l_k's and r_k's are all distinct. The term l-value(s) and r-value(s) will refer to the l_k's and r_k's, respectively. Finally, unless otherwise stated, the term "constant" will mean a quantity that depends only on the number of relays and is independent of the channel SNRs.

3.2.2 An Approximation to the Capacity

Let C^n_{hd} denote the capacity of the n-relay half-duplex diamond network; to achieve it, we need to optimize over p_i, the fraction of time that the relays are in state m_i. As described in Chapter 2, C^n_{hd} can be approximated, upto constant additive terms, by C^n_{lp} which only depends on the individual link capacities $\{l_k, r_k\}$ as defined in (3.1). C^n_{lp} can also be viewed as the optimum solution of a linear program. For completeness, we state the minimization version of it (which is the dual program described in Chapter 2).

Let $\mathbf{p}^d$ denote the vector $(p^d_1, \ldots, p^d_{2^n})$. The dual can be written as follows.

DLP : Minimize C^d

$$\sum_{j=1}^{2^n} p^d_j \left(\max_{k\in\bar{\Lambda}_j\cap L(m_i)} l_k + \max_{k\in\Lambda_j\cap T(m_i)} r_k \right) \leq C^d \text{ for each } i \in [2^n] \tag{3.2}$$

$$\sum_{j=1}^{2^n} p^d_j = 1;\ \forall j, p^d_j \geq 0, C^d \geq 0 \tag{3.3}$$

Each dual variable p^d_j corresponds to the cut Λ_j and each dual constraint (except the last one) corresponds to the relaying state m_i. p^d_j can be thought of as non-negative weights given to each cut with their sum normalized to one.

$l_k (k \in [n])$	Capacity of point to point channel from $\mathcal{S}$ to $\mathcal{R}_k$
$r_k (k \in [n])$	Capacity of point to point channel from $\mathcal{R}_k$ to $\mathcal{D}$
C_{hd}^n	Capacity of n-relay half-duplex diamond network
C_{lp}^n	Approximation to capacity through linear program
$m_i (i \in [2^n])$	Relaying state
$\Lambda_j (j \in [2^n])$	Cut in the network
p_j^d	The dual variable corresponding to cut Λ_j
S_{ij}, S_{RS}	The rate achieved by routing strategy using $\mathcal{R}_i, \mathcal{R}_j$ and the maximum over all pairs

Table 3.1 – Summary of terms defined in this section.

3.2.3 Routing Strategies

We define a *routing strategy* as one that employs exactly two relays $\mathcal{R}_i$ and $\mathcal{R}_j$ $(i < j)$, and operates them using only two states $\{L, T\}$ and $\{T, L\}$, where each relay in T performs a decode-and-forward operation. Let p_1, p_2 be the fraction of time ($\mathcal{R}_i$, $\mathcal{R}_j$) are in the states (L, T) and (T, L) respectively. We define the maximum rate achieved by such a strategy as S_{ij}, which is given by

$$S_{ij} = \max_{\substack{p_1, p_2 \\ p_1 + p_2 = 1}} \min(p_1 l_i, p_2 r_i) + \min(p_2 l_j, p_1 r_j) \tag{3.4}$$

where the first term is the rate carried by the first and the second term is the rate carried by the second relay. This maximization can be solved to obtain ([31, 30])

$$S_{ij} = \frac{l_j(r_j + l_i)}{l_j + r_j} \text{ if } l_i l_j \leq r_i r_j, l_i \leq l_j \tag{3.5}$$

$$= \frac{l_i(l_j + r_i)}{l_i + r_i} \text{ if } l_i l_j \leq r_i r_j, l_i \geq l_j \tag{3.6}$$

$$= \frac{r_i(l_i + r_j)}{l_i + r_i} \text{ if } l_i l_j \geq r_i r_j, r_i \geq r_j \tag{3.7}$$

$$= \frac{r_j(l_j + r_i)}{l_j + r_j} \text{ if } l_i l_j \geq r_i r_j, r_i \leq r_j \tag{3.8}$$

The maximum rate achievable by our *routing strategy* is denoted by S_{RS}. It can be calculated by maximizing S_{ij} over all possible pairs of relays, i.e.,

$$S_{\text{RS}} = \max_{i,j \in [n], i<j} S_{ij} \tag{3.9}$$

The various terms defined in this section are summarized in Table 3.1.

3.2.4 Main Results

The main result in this chapter is the following.

Theorem 3.2.1 *For any n-relay half-duplex diamond network,*

$$S_{RS} \geq \begin{cases} \frac{8}{9}C_{lp}^n \text{ for } n = 2 \\ \frac{1}{2}C_{lp}^n \text{ for } n \geq 3 \end{cases} \tag{3.10}$$

Since $C_{lp}^n \geq C_{hd}^n - G(n)$ (from Section 2 in Chapter 2), if $S_{\text{RS}} \geq \gamma C_{lp}^n$ for some fraction $\gamma \in [0, 1]$, then it follows that $S_{\text{RS}} \geq \gamma C_{hd}^n - \gamma G(n)$. That is, routing strategies achieve at least γ fraction of the capacity of the network, up to constant additive factors. The proof of the theorem naturally implies the following proposition.

Proposition 3.2.2 *In any n-relay half-duplex diamond network, we can find a pair of relays i, j, which when operated using routing strategies achieves a rate at least equal to $\frac{1}{2}C_{lp}^n$ in $O(n \log n)$ time.*

3.3 Proof for 2-Relay Networks

In this section, we prove Thm. 3.2.1 for 2-relay networks. For brevity, assume $\{l_1, l_2, r_1, r_2\} = \{a, b, c, d\}$. The linear program for C_{lp}^2 can be solved exactly to obtain closed form expressions. Depending on the relative values of a, b, c, d, the expressions can take different forms. We will show the proofs for the case $ab \leq cd$ and $a \geq b, c \leq d$. For this case

$$C_{lp}^2 = \frac{ac(b+d) + bd(a-b)}{(b+d)(a+c-b)} \tag{3.11}$$

and

$$S_{\text{RS}} = \frac{a(b+c)}{a+c} \tag{3.12}$$

3.3.1 Proof of Thm. 3.2.1 for 2-relay networks

Using the expressions for S_{RS} and C_{lp}^2, we have

$$\frac{9S_{\text{RS}}}{8C_{lp}^2} - 1 = \frac{9ab^2(a-b) + abc(a+c) + df_1(a,b,c)}{8(a+c)(ac(b+d) + bd(a-b))} \tag{3.13}$$

where $f_1(a,b,c) = a^2b - ab^2 + a^2c - 8abc + 8b^2c + ac^2$. Writing f_1 as a quadratic expression in c, we have

$$f_1(a,b,c) = ac^2 + (a^2 - 8ab + 8b^2)c + ab(a-b)$$

Clearly, if $a^2 - 8ab + 8b^2 \geq 0$, then $f_1(a,b,c) \geq 0$. Since the equation $x^2 - 8x + 8 = 0$ has two roots approximately equal to 1.17 and 6.82, as long as $a/b \in [1, 1.17] \cup [6.82, +\infty]$, $a^2 - 8ab + 8b^2 \geq 0$ and hence $f_1(a,b,c) \geq 0$. On the other hand, we can also look at f_1 as a quadratic function in c and look at its discriminant $\Delta(a,b)$ as a function of a, b. We have

$$\begin{aligned}\Delta(a,b) &= (a^2 - 8ab + 8b^2)^2 - 4a(ab(a-b)) \\ &= (a-2b)^2(a^2 - 16ab + 16b^2)\end{aligned}$$

Since the roots of $x^2 - 16x + 16 = 0$ are approximately 1.07 and 14.92, the discriminant $\Delta(a,b) < 0$ if $1.07 \leq a/b \leq 14.92$, in which case f_1 as a function of c is non-negative. Since the interval $[1, 1.17] \cup [6.82, +\infty] \cup [1.07, 14.92]$ covers all possible values of a/b, we can conclude that $f_1(a,b,c) \geq 0$ in all cases. Hence

$$\frac{9S_{\text{RS}}}{8C_{lp}^2} - 1 \geq 0 \implies \frac{S_{\text{RS}}}{C_{lp}^2} \geq \frac{8}{9} \tag{3.14}$$

which proves the theorem. ■

The multiplicative ratio is essentially the best we can obtain. Consider the network with $a = 2e, b = e, c = e, d = ke$ for some $k > 2$ and $e > 0$. Then, plugging in the expressions for S_{RS} and C_{lp}^2, we have

$$\frac{S_{\text{RS}}}{C_{lp}^2} = \frac{4(2+2k)}{3(2+3k)} \to \frac{8}{9} \text{ as } k \to \infty$$

To summarize, we have shown that for the 2-relay half-duplex diamond network, routing strategies can achieve at least 8/9 of the capacity of the network, approximately.

3.4 Proof for Antisymmetric Networks

We now turn our attention to the proof of Thm. 3.2.1 for networks with $n > 2$ relays. To this end, we will show appropriate lower bounds on S_{RS} (the maximum rate achievable by routing strategies) and upper bounds on C_{lp}^n. The ratio between these two bounds will lead to a lower bound on S_{RS}/C_{lp}^n. While the lower bound on S_{RS} can be obtained by choosing a specific pair of relays, we derive fairly tight upper bounds on C_{lp}^n by computing suitable dual feasible solutions to **LP**. In this section, we first prove the theorem for a special class of diamond networks that we call *antisymmetric* and then proceed to apply

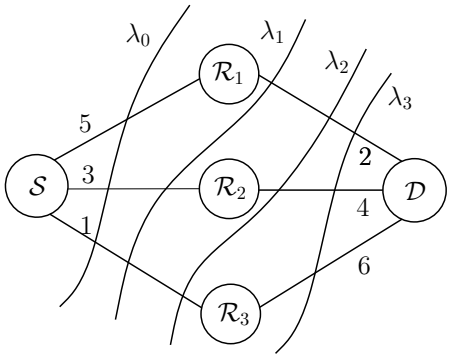


Figure 3.2 – Example of an antisymmetic network with $l_1 = 5, l_2 = 3, l_3 = 1$ and $r_1 = 2, r_2 = 4, r_3 = 6$. The tight cuts λ_0,λ_1,λ_2 and λ_3 are also shown.

the techniques developed therein to arbitrary diamond networks in the next section.

3.4.1 Antisymmetric Networks

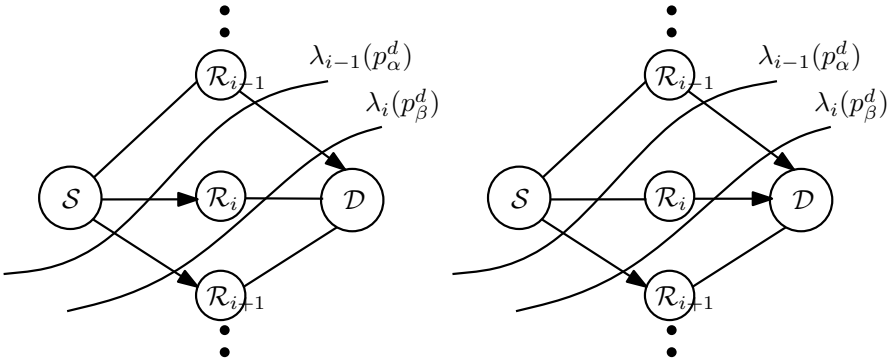
An *antisymmetric* network has $l_i > l_j$ and $r_i < r_j$ for $i < j$, that is, the l-values and r-values are arranged in descending and ascending order, respectively (hence the term antisymmetric). An example with 3 relays is shown in Fig. 3.2. The cross cuts with $\mathcal{R}_1, \ldots, \mathcal{R}_i$ on the side of the source and $\mathcal{R}_{i+1}, \ldots, \mathcal{R}_n$ on the other side will be particularly useful for deriving the upper bounds. We denote them by λ_i. In the special cases of λ_0 and λ_n, all the relays are on source side or destination side of the cut. They are also displayed for the 3 relay example in Fig. 3.2.

3.4.2 Upper Bounds for Antisymmetric Networks

Since **LP** is a maximization problem, from weak-duality [40], any feasible solution of **DLP** is an upper bound to the value of the optimum in **LP**. To keep these bounds analytically tractable, we would like to have dual feasible solutions that only have a few non-zero states. Further, to get upper bounds that are close to C_{lp}^n, we need to assign values to dual variables that correspond to *tight* cuts in the network, i.e., the ones that are likely to be the tight constraints for the optimal solution to **LP**. In the specific case of antisymmetric networks, the cross-cuts λ_i constitute a natural set of candidates for such tight cuts. In the following lemma, we make this intuition concrete and derive n upper bounds $U_i, i \in [n]$ to C_{lp}^n based on them.

Lemma 3.4.1 *Define $r_0 = l_{n+1} = 0$. In an antisymmetric network, for each $i \in [n]$*

$$C_{lp}^n \leq U_i \triangleq \frac{(r_i - r_{i-1})(l_i + r_{i-1}) + (l_{i+1} + r_{i-1})(l_i - l_{i+1})}{l_i - l_{i+1} + r_i - r_{i-1}} \tag{3.15}$$



Relaying State m_α Relaying State m_β

Figure 3.3 – Dominating relaying states for chosen cuts λ_{i-1} and λ_i in an antisymmetric network.

Proof: First consider the case when $i \in [2, \ldots, n-1]$. The non-zero variables we pick in **DLP** will correspond to the two successive cross-cuts λ_{i-1} and λ_i. Let the two corresponding dual variables be p_α^d and p_β^d. Since the remaining dual variables are zero, the constraint of **DLP** corresponding to relaying state m reduces to

$$p_\alpha^d\Big(\max_{k\in\bar{\lambda}_{i-1}\cap L(m)} l_k + \max_{k\in\lambda_{i-1}\cap T(m)} r_k\Big) + p_\beta^d\Big(\max_{k\in\bar{\lambda}_i\cap L(m)} l_k + \max_{k\in\lambda_i\cap T(m)} r_k\Big) \le C^d \tag{3.16}$$

We will show that in this simplified dual program, exactly two constraints dominate all others, for any positive values of p_α^d and p_β^d. As shown in Fig. 3.3, these constraints correspond to the relaying states where $\mathcal{R}_1, \ldots, \mathcal{R}_{i-1}$ are in T (and the remaining in L) and where $\mathcal{R}_1, \ldots, \mathcal{R}_i$ are in T (and the remaining in L). Call these states m_α and m_β, respectively. The constraints for m_α and m_β are

$$p_\alpha^d(l_i + r_{i-1}) + p_\beta^d(l_{i+1} + r_{i-1}) \le C^d \tag{3.17}$$
$$p_\alpha^d(l_{i+1} + r_{i-1}) + p_\beta^d(l_{i+1} + r_i) \le C^d \tag{3.18}$$

Consider any other relaying state m that has $\mathcal{R}_i$ in L. In m, let i_1 be the highest index less than i such that $\mathcal{R}_{i_1}$ is in T and i_2 be the lowest index greater than i such that $\mathcal{R}_{i_2}$ is in L. Then the constraint for m is

$$p_\alpha^d(l_i + r_{i_1}) + p_\beta^d(l_{i_2} + r_{i_1}) \le C^d \tag{3.19}$$

Since $i_1 \le i-1$ (which implies $l_{i_1} \ge l_{i-1}$) and $i_2 \ge i+1$ (which implies $r_{i_2} \ge r_{i+1}$), this constraint is dominated by (3.17) above. Similarly, if we consider any relaying state that has $\mathcal{R}_i$ in T, the corresponding constraint will be dominated by (3.18).

Therefore, any value of (p_α, p_β) that satisfies (3.17) and (3.18) with equality will automatically satisfy the other $2^n - 2$ constraints. Thus, we get the following system of equations.

$$p_\alpha^d(l_i + r_{i-1}) + p_\beta^d(l_{i+1} + r_{i-1}) = C^d \tag{3.20}$$
$$p_\alpha^d(l_{i+1} + r_{i-1}) + p_\beta^d(l_{i+1} + r_i) = C^d \tag{3.21}$$
$$p_\alpha^d + p_\beta^d = 1 \tag{3.22}$$

By weak duality, solving this system gives us the claimed upper bound for C_{lp}^n. For $i = 1$ and $i = n$, a similar argument holds. For $i = 1$, the dominating constraints correspond to the relaying states where all the relays are in L (corresponding to m_α) and when only $\mathcal{R}_1$ is in T (corresponding to m_β). For $i = n$, the dominating constraints correspond to the relaying states when only $\mathcal{R}_n$ is in L (corresponding to m_α) and when all the relays are in T (corresponding to m_β). ■

To get the lower bounds to S_{RS}, we will choose one from the $n-1$ subnetworks consisting of the relays $\mathcal{R}_i, \mathcal{R}_{i+1}$ for $1 \leq i \leq n-1$. The precise choice will become clear in the context of the proof of Thm. 3.2.1 for antisymmetric networks, that we present below.

3.4.3 Proof of Thm. 3.2.1 for Antisymmetric Networks

Recall that we denote by U_i the upper bound proved in the previous lemma. For $i \in [n-1]$, consider the determinant $\Delta_i = l_i l_{i+1} - r_i r_{i+1}$. In an antisymmetric network, Δ_i is a decreasing function of i. Suppose the sign of the determinant changes at t, i.e., $\Delta_t \leq 0$ and $\Delta_{t-1} > 0$ for some $t \in [2, \ldots, n-1]$. We claim that the following holds

$$\frac{\max\left(S_{t-1,t}, S_{t,t+1}\right)}{U_t} \geq \frac{1}{2} \tag{3.23}$$

For brevity, let $b = l_{t-1}$, $c = l_t$, $d = l_{t+1}$, $e = r_{t-1}$, $f = r_t$, $g = r_{t+1}$. Indeed, we have

$$U_t = \frac{(c+e)(f-e) + (d+e)(c-d)}{c-d+f-e} \tag{3.24}$$

and

$$S_{t-1,t} = \frac{f(c+e)}{c+f} \text{ and } S_{t,t+1} = \frac{c(d+f)}{c+f} \tag{3.25}$$

Thus,

$$\frac{S_{t-1,t} + S_{t,t+1}}{U_t} - 1 = \frac{f(c-d)^2 + c(e-f)^2}{(c+f)((c+e)(f-e) + (d+e)(c-d))} \geq 0 \tag{3.26}$$

Since U_t is an upper bound to C_{lp}^n, the maximum of $S_{t-1,t}$ and $S_{t,t+1}$ will give us the required bound. When $\Delta_i \geq 0$ for all $i \in [n-1]$, for brevity, let $a = l_{n-1}$, $b = l_n$,

$c = r_{n-1}$, $d = r_n$. We have

$$U_n = \frac{d(b+c) - c^2}{d - c + b},\ S_{n-1,n} = \frac{d(b+c)}{b+d} \tag{3.27}$$

Then,

$$\frac{S_{n-1,n}}{U_n} - \frac{1}{2} = \frac{dc(d-c) + db(d-c) + c^2 b + db^2}{2(d+b)(d(b+c) - c^2)} \geq 0 \tag{3.28}$$

In the other extreme case when $\Delta_i \leq 0$ for all $i \in [n-1]$, let $a = l_1$, $b = l_2$, $c = r_1$, $d = r_2$. We have

$$U_1 = \frac{a(b+c) - b^2}{a - b + c},\ S_{1,2} = \frac{a(b+c)}{a+c} \tag{3.29}$$

and

$$\frac{S_{1,2}}{U_1} - \frac{1}{2} = \frac{ab(a-b) + ac(a-b) + b^2 c + ac^2}{2(a+c)(a(b+c) - b^2)} \geq 0 \tag{3.30}$$

Since in each case we have a pair i, j such that $S_{ij}/C^n_{lp} \geq 1/2$, this implies $S_{\mathrm{RS}}/C^n_{lp} \geq 1/2$. ■

3.5 Proof for General Networks

The proof for general networks builds on the proof for antisymmetric networks. The main idea is to extract a subnetwork, which we call the *skeleton*, that is in an antisymmetric configuration and derive lower bounds to S_{RS} and upper bounds to C^n_{lp} based on certain parameters of the skeleton.

3.5.1 Skeleton of General Networks

The *skeleton* of a general diamond network is derived as follows. If not already done, the relays are reordered such that the l-values are arranged in a decreasing order. After re-ordering, the first relay is always in the skeleton; we then sequentially go through the relays until we get a relay $\mathcal{R}_{k_1}$ such that $l_{k_1} < l_1$ and $r_{k_1} > r_1$. Next, we look for another relay $\mathcal{R}_{k_2}$ such that $l_{k_2} < l_{k_1}$ and $r_{k_2} > r_{k_1}$, and so on. In other words, we grow a subnetwork which is in an antisymmetric configuration.

We denote the indices of the relays in the skeleton by $a_1, \ldots, a_p$, where p is the size of the skeleton. For convenience, we also define $a_0 = 0$ and $r_0 = 0$. Similarly, we define $a_{p+1} = n+1$ and $l_{n+1} = 0$. The same skeleton can be derived by sorting the relays in decreasing order of r-values and repeating the procedure above starting from the relay with the highest r-value.

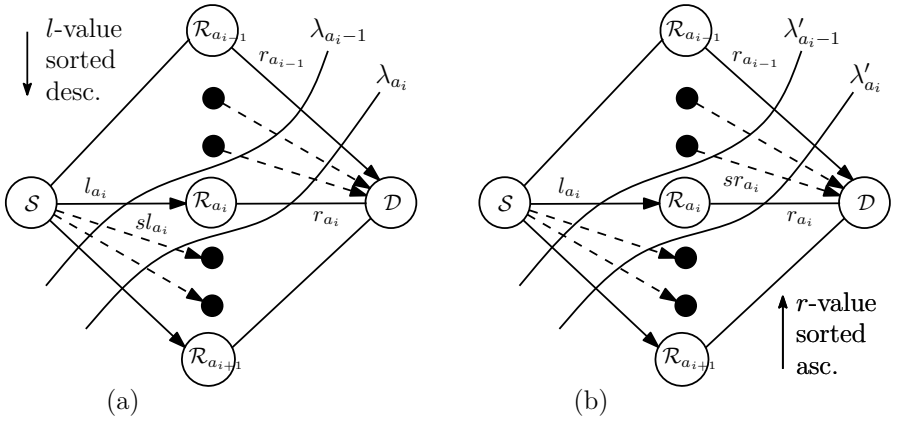


Figure 3.4 – (a) The cross cuts λ_{a_i-1} and λ_{a_i} shown with the relaying state m_α. The relays are ordered with decreasing l-values. (b) The cross cuts λ'_{a_i-1} and λ'_{a_i} shown with the relaying state m'_α. The relays are ordered with increasing r-values. The links for relays not in the skeleton are shown in dashed lines.

Assuming for the moment $p \geq 2$, it will be beneficial to visualize the whole network from the perspective of the skeleton. With the l-values of the whole network sorted in descending order, a set of three consecutive relays in the skeleton ($\mathcal{R}_{a_{i-1}}$, $\mathcal{R}_{a_i}$, $\mathcal{R}_{a_{i+1}}$) embedded in the network is shown in Fig. 3.4(a).

3.5.2 Upper Bounds for General Networks

For each relay $\mathcal{R}_{a_i}$ in the skeleton, we need to define a new variable sl_{a_i} (which stands for *successor left*) to denote the greatest l-value in the whole network (including the relays not in the skeleton) that is lower than the l_{a_i}. If such a link does not exist, then $sl_{a_i} = 0$.

We pick two cross cuts λ_{a_i-1} and λ_{a_i} defined as follows. λ_{a_i-1} is the cross cut where $\mathcal{R}_{a_i}$ and all relays below it belong to the side of the $\mathcal{D}$, whereas λ_{a_i} is the cross cut where $\mathcal{R}_{a_i}$ and all relays above it belong to the side of $\mathcal{S}$. Let the dual variables corresponding to these cuts be p^d_α and p^d_β respectively. With respect to these two cuts, the constraints that *dominate* correspond to the states where (i) $\mathcal{R}_{a_i}$ and every relay (from the whole network) below it are in state L and the others are in state T (call this relaying state m_α) and (ii) $\mathcal{R}_{a_i}$ and every relay (from the whole network) above it are in state T and the others are in state L (call this relaying state m_β).

The interaction between m_α and λ_{a_i-1} and λ_{a_i} is shown in Fig. 3.4(a). The dual constraint arising from this is

$$p^d_\alpha(l_{a_i} + r_{a_{i-1}}) + p^d_\beta(sl_{a_i} + r_{a_{i-1}}) \leq C^d \tag{3.31}$$

Since the r-values of relays between $\mathcal{R}_{a_{i-1}}$ and $\mathcal{R}_{a_i}$ are less than $r_{a_{i-1}}$, these values do

not play a role. The dual constraint from the interaction between m_β and the two cuts is

$$p_\alpha^d(sl_{a_i} + r_{a_{i-1}}) + p_\beta^d(sl_{a_i} + r_{a_i}) \leq C^d \tag{3.32}$$

Using the same arguments as in the proof of Lemma 3.4.1, we can show that the value of $(p_\alpha^d, p_\beta^d, C^d)$ obtained from the following set of equations is dual feasible for **LP**.

$$p_\alpha^d(l_{a_i} + r_{a_{i-1}}) + p_\beta^d(sl_{a_i} + r_{a_{i-1}}) = C^d \tag{3.33}$$
$$p_\alpha^d(sl_{a_i} + r_{a_{i-1}}) + p_\beta^d(sl_{a_i} + r_{a_i}) = C^d \tag{3.34}$$
$$p_\alpha^d + p_\beta^d = 1 \tag{3.35}$$

On solving, we get the following upper bound to C_{lp}^n

$$U_{i,1} = \frac{(l_{a_i} - sl_{a_i})(r_{a_i} + sl_{a_i}) + (r_{a_{i-1}} + sl_{a_i})(r_{a_i} - r_{a_{i-1}})}{r_{a_i} - r_{a_{i-1}} + l_{a_i} - sl_{a_i}} \tag{3.36}$$

Since $r_{a_0} = r_0 = 0$, this bound is valid for all $i \in [p]$.

We now reorient the network by sorting the relays in ascending order of r-values. The same skeleton $\mathcal{R}_{a_1}, \ldots, \mathcal{R}_{a_p}$ can then be defined by starting from the relay with the highest r-value. Define a new variable sr_{a_i} (which stands for *successor right*) to denote the greatest r-value in the whole network (including the relays not in the skeleton) that is lower than the r-value of $\mathcal{R}_{a_i}$. If there is no such link, then $sr_{a_i} = 0$. With the r-values of the whole network sorted in ascending order, a set of three consecutive relays in the skeleton ($\mathcal{R}_{a_{i-1}}$, $\mathcal{R}_{a_i}$, $\mathcal{R}_{a_{i+1}}$) embedded in the network is shown in Fig. 3.4(b).

We pick two cross cuts λ'_{a_i-1} and λ'_{a_i} defined as follows. λ'_{a_i-1} is the cross cut where $\mathcal{R}_{a_i}$ and all relays below it belong to the side of the $\mathcal{D}$, whereas λ'_{a_i} is the cross cut where $\mathcal{R}_{a_i}$ and all relays above it belong to the side of $\mathcal{S}$. Note that in general $\lambda'_{a_i} \neq \lambda_{a_i}$. Let the dual variables corresponding to these cuts be p_α^d and p_β^d respectively. With respect to these two cuts, the constraints that *dominate* correspond to the states where (i) $\mathcal{R}_{a_i}$ and every relay (from the whole network) below it are in state L and the others are in state T (call this relaying state m'_α) and (ii) $\mathcal{R}_{a_i}$ and every relay (from the whole network) above it are in state T and the others are in state L (call this relaying state m'_β).

The interaction between m'_α and λ'_{a_i-1} and λ'_{a_i} is shown in Fig. 3.4(b). The dual constraint arising from this interaction is

$$p_\alpha^d(l_{a_i} + sr_{a_i}) + p_\beta^d(l_{a_{i+1}} + sr_{a_i}) \leq C^d \tag{3.37}$$

Since the l-values of relays between $\mathcal{R}_{a_{i-1}}$ and $\mathcal{R}_{a_{i+1}}$ are less than $l_{a_{i+1}}$, these values do not play a role. The dual constraint from the interaction between m'_β and the two cuts is

$$p_\alpha^d(l_{a_{i+1}} + sr_{a_i}) + p_\beta^d(l_{a_{i+1}} + r_{a_i}) \leq C^d \tag{3.38}$$

Using the same arguments as in the proof of Lemma 3.4.1, we can show that the value of $(p^d_\alpha, p^d_\beta, C^d)$ obtained from the following set of equations is dual feasible for **LP**.

$$p^d_\alpha(l_{a_i} + sr_{a_i}) + p^d_\beta(l_{a_{i+1}} + sr_{a_i}) = C^d \tag{3.39}$$
$$p^d_\alpha(l_{a_{i+1}} + sr_{a_i}) + p^d_\beta(l_{a_{i+1}} + r_{a_i}) = C^d \tag{3.40}$$
$$p^d_\alpha + p^d_\beta = 1 \tag{3.41}$$

On solving, we get the following upper bound to C^n_{lp}

$$U_{i,2} = \frac{(r_{a_i} - sr_{a_i})(l_{a_i} + sr_{a_i}) + (l_{a_{i+1}} + sr_{a_i})(l_{a_i} - l_{a_{i+1}})}{l_{a_i} - l_{a_{i+1}} + r_{a_i} - sr_{a_i}} \tag{3.42}$$

Since $l_{a_{p+1}} = l_{n+1} = 0$, this bound is valid for all $i \in [p]$.

We have thus proven the following lemma, which gives upper bounds to C^n_{lp} for a general diamond network in terms of the l, r, sl and sr values of the links appearing in the skeleton.

Lemma 3.5.1 *Define $l_{a_{p+1}} = l_{n+1} = r_{a_0} = r_0 = 0$. In a general diamond network with $p \geq 2$ relays in the skeleton, for each $i \in [p]$*

$$C^n_{lp} \leq U_{i,1} = \frac{(l_{a_i} - sl_{a_i})(r_{a_i} + sl_{a_i}) + (r_{a_{i-1}} + sl_{a_i})(r_{a_i} - r_{a_{i-1}})}{r_{a_i} - r_{a_{i-1}} + l_{a_i} - sl_{a_i}} \tag{3.43}$$
$$C^n_{lp} \leq U_{i,2} = \frac{(r_{a_i} - sr_{a_i})(l_{a_i} + sr_{a_i}) + (l_{a_{i+1}} + sr_{a_i})(l_{a_i} - l_{a_{i+1}})}{l_{a_i} - l_{a_{i+1}} + r_{a_i} - sr_{a_i}} \tag{3.44}$$

Clearly, for antisymmetric networks $p = n$, $sl_{a_i} = l_{i+1}$ and $sr_{a_i} = r_{i-1}$ and we get back the bounds from the previous section with $U_{i,1} = U_{i,2}$. We are now in a position to prove our claim $S_{\mathrm{RS}}/C^n_{lp} \geq 1/2$ for general diamond networks.

3.5.3 Proof of Thm. 3.2.1 for General Networks

When the skeleton is of size one, i.e., $p = 1$, there is relay that dominates all others in terms of the l and r-values. In this case, we can show that

$$C^n_{lp} \leq U_1 = \frac{(l_1 - sl_1)(r_1 + sl_1) + sl_1 r_1}{r_1 + l_1 - sl_1} \tag{3.45}$$

Further, if we pick the relay in the skeleton and the relay whose l-value is sl_1, then the rate achieved by running the routing strategy on these two relays is at least $U_1/2$.

Now consider the general case when the skeleton has at least 2 relays (i.e. $p \geq 2$). To get a lower bound on S_{RS}, it will be sufficient to only look at the links in the skeleton. Consider

the $p-1$ 2-relay networks consisting of $\mathcal{R}_{a_i}$ and $\mathcal{R}_{a_{i+1}}$ and define the determinant to be $\Delta_i = l_{a_i} l_{a_{i+1}} - r_{a_i} r_{a_{i+1}}$. Since the skeleton is an antisymmetric network by itself, Δ_i is a decreasing function of i. Suppose the sign of the determinant changes at t, i.e., $\Delta_t \leq 0$ and $\Delta_{t-1} > 0$ for some $t \in [2, \ldots, p-1]$. We claim that the following holds

$$\frac{\max\left(S_{a_{t-1},a_t}, S_{a_t,a_{t+1}}\right)}{\min(U_{t,1}, U_{t,2})} \geq \frac{1}{2} \tag{3.46}$$

where $U_{t,1}, U_{t,2}$ are the upper bounds defined in the previous lemma. For brevity, let $c = l_{a_t}$, $d = l_{a_{t+1}}$, $x = sl_{a_t}$ and $e = r_{a_{t-1}}$, $f = r_{a_t}$, $y = sr_{a_t}$.

Then, writing $U_{t,2}$ as a function of $y = sr_{a_t}$, we get

$$U_{t,2}(y) = \frac{(f-y)(c+y) + (d+y)(c-d)}{c-d+f-y} \tag{3.47}$$

$$= d + \frac{f(c-d) + y(f-y)}{c-d+f-y} \tag{3.48}$$

Since $f > y$ and $c > d$,

$$\frac{\partial U_{t,2}(y)}{\partial y} = \frac{(f-y)(f-y+2(c-d))}{(f-y+c-d)^2} > 0 \tag{3.49}$$

Thus, $U_{t,2}(y)$ is an increasing function of y and the maximum is reached when $y = f$ (since y has to be less than f). Therefore, $U_{t,2}(y) < U_{t,2}(f) = d + f$. Similarly, by considering $U_{t,1}$ as a function of $x = sl_{a_t}$ we can conclude that $U_{t,1}(x) < U_{t,1}(c) = c + e$. Thus,

$$\frac{S_{a_{t-1},a_t} + S_{a_t,a_{t+1}}}{\min(U_{t,1}, U_{t,2})} > \frac{S_{a_{t-1},a_t} + S_{a_t,a_{t+1}}}{\min(d+f, c+e)} \tag{3.50}$$

Now,

$$S_{a_{t-1},a_t} = \frac{f(c+e)}{c+f} \text{ and } S_{a_t,a_{t+1}} = \frac{c(d+f)}{c+f} \tag{3.51}$$

Assume $d + f \leq c + e$. Then,

$$\frac{S_{a_{t-1},a_t} + S_{a_t,a_{t+1}}}{d+f} - 1 = \frac{f(c+e-d-f)}{(c+f)(d+f)} \geq 0 \tag{3.52}$$

The case of $d + f \geq c + e$ is analogous. To conclude, the maximum among S_{a_{t-1},a_t} and $S_{a_t,a_{t+1}}$ will be greater than $\min(U_{t,1}, U_{t,2})/2$ and hence greater than $C^n_{lp}/2$. The extreme cases when $\Delta_i \geq 0$ or $\Delta_i \leq 0$ for all $i \in [p]$ can be handled in a manner similar to the proof for antisymmetric networks. ■

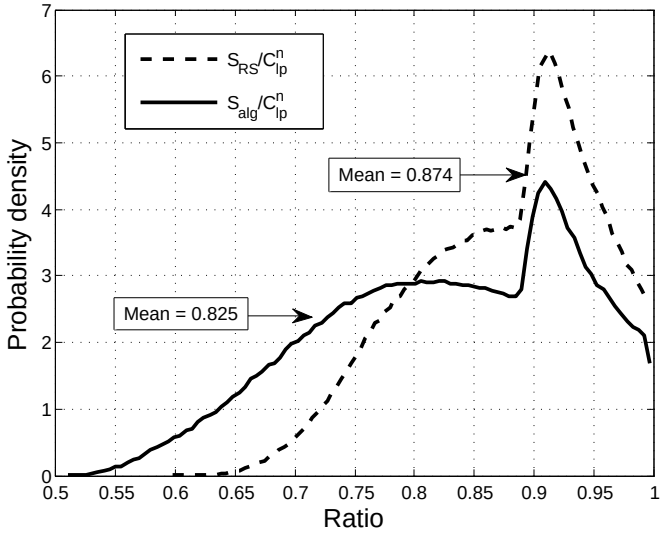


Figure 3.5 – Probability density of the ratio of rate achieved by routing strategies and C^n_{lp} for randomly chosen 5-relay networks.

3.6 Algorithms and Simulation Results

The proof of Thm. 3.2.1 readily gives us an algorithm for determining a pair of relays that (approximately) achieves at least half the capacity of the network using only routing strategies.

3.6.1 Proof of Proposition 3.2.2

We first need to compute the skeleton, which can be accomplished in linear time once the relays have been sorted in descending order of l-values. The exact pair is then determined by computing the determinant Δ_i of the $p-1$ successive pairs of relays in the skeleton (when $p \geq 2$). If the determinant is $\Delta_i \geq 0$ for all $i \in [p]$, then the last pair is our output, else if $\Delta_i \leq 0$ for all $i \in [p]$, then the first pair is our output. Otherwise, the output is one of the pairs where the sign of the determinant changes. In case there is just one relay in the skeleton, the output of the algorithm follows from the first part of the proof of Thm. 3.2.1 in Section 3.5. Clearly, the time for sorting the network in terms of the l-values dominates other computations and hence our algorithm takes $O(n \log n)$ time.

3.6.2 Simulation Results

Let the rate achieved by the pair of relays output by the algorithm above be S_{alg}. We have in fact proved that $S_{alg}/C^n_{lp} \geq 1/2$. It is easy to construct examples where this bound is tight, but $1/2$ may be a pessimistic bound for other channel configurations. In Fig. 3.5, we present simulation results that plots the p.d.f for S_{alg}/C^n_{lp} and S_{RS}/C^n_{lp} for

10^6 tuples of l and r-values chosen uniformly at random from the range $[1, 200]$ for a 5-relay diamond network.

The mean value of S_{alg}/C_{lp}^n is about 0.825, which is less than 6% lower than the mean value of S_{RS}/C_{lp}^n, i.e., when the best pair of relays is chosen. If we look at the cumulative distribution, over the random choices of l and r-values, S_{alg}/C_{lp}^n is greater than 0.6 in more than 98.1% of the cases and is greater than 0.7 in more than 85.7% of the cases. On the other hand the time taken by the algorithm is $O(n \log n)$ in place of $O(n^2)$ that an exhaustive search for the best pair of relays takes.

---

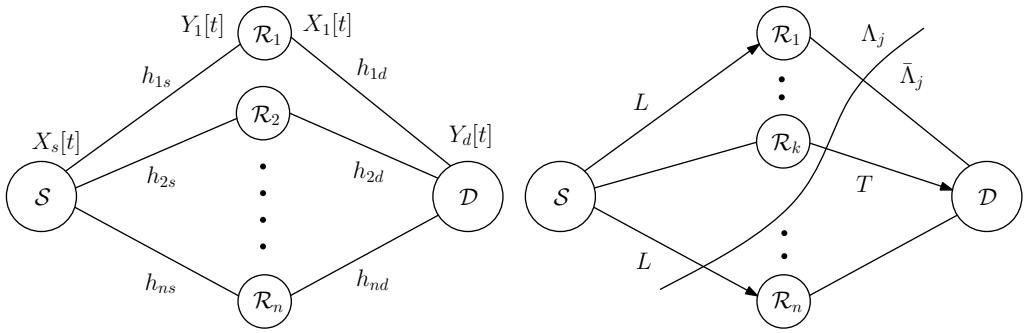
4 Local Strategies for Half-Duplex Diamond Networks

In the previous two chapters, we presented two different approaches to reduce the operational and design complexity of half-duplex diamond networks. A major assumption in both these approaches is that the global channel state information is known to a centralized authority (e.g. the source) which can then compute the low complexity schedule or select a pair of relays for routing. In many practical scenarios, especially when the channel strengths are changing rapidly, this assumption may be too prohibitive as communicating the CSI to a centralized node incurs a cost.

In this chapter, we investigate the performance we can achieve if we restrict ourselves to only *local* relay operations. By local, we mean that two conditions are satisfied: (i) each relay in the network only has access to its incoming and outgoing channel realizations, and (ii) there can be no communication between the relays to share CSI, and hence, no node in the network can solve a centralized optimization problem.

We propose the following approach for the network operation: every relay in the n-relay network uses its incoming and outgoing links to derive the half-duplex listen and transmit fractions that would be optimal in the absence of all other relays in the network. We then allow the relays to *switch* multiple times (say σ) between listen and transmit states *independently at random* in the duration of operation, while still respecting the overall listen-transmit fractions at each relay.

We show that, using this approach for the 2 relay diamond network, we can achieve at least 3/4 of the capacity of the network (approximately) as the number of switches $\sigma \to \infty$. We also show that for a *deterministic* local strategy with only *one* listen-transmit cycle the above fraction drops to 1/2. Interestingly, we see that incorporating only a few switches already enables to leverage most of the benefits that we prove in the limiting $\sigma \to \infty$ case. Numerical evaluation of our strategies over networks with larger number of relays show similar trends. To prove the results, we use the same linear programming approximations to the capacity of half-duplex diamond networks as in Chapter 2.



(a) The Gaussian n-relay half-duplex diamond network.

(b) Relaying states and cuts in the network.

Figure 4.1 – Network model with channel coefficients of the individual links, relaying states and cuts. For a particular relaying state, an arrow on a link denotes that it is active.

4.1 Related Work

In previous work, the authors in [41] show for a specific case that schedules using only local information and one switch at each relay can achieve rates close to capacity. The authors in [30], [42] approach low complexity relaying in half-duplex diamond networks by reducing the number of relaying states or using a subset of relays to approximately achieve a constant fraction of the capacity . We would like to note that the use of randomization in our work is fundamentally different from the use of random switching in [43] and [32]. In their work, the randomness in the switching sequence is used to convey additional information from the source to the destination while we use randomness to generate schedules that achieve higher rates.

4.2 Problem Formulation

4.2.1 Network Model

We consider the Gaussian n-relay diamond network where a source $\mathcal{S}$ transmits information to a destination $\mathcal{D}$ with the help of half-duplex relays. The notations and equations for signals are the same as Chapter 2. A representative figure is shown in Fig. 4.1a.

We can then calculate the individual link capacities from $\mathcal{S}$ to $\mathcal{R}_k$ (l_k) and from $\mathcal{R}_k$ to $\mathcal{D}$ (r_k) as

$$l_k = \log(1 + |h_{ks}|^2 P), \quad r_k = \log(1 + |h_{kd}|^2 P) \tag{4.1}$$

Let $[n]$ represent the set $\{1, 2, \cdots, n\}$. For $i \in [2^n]$, let $m_i \in M = \{L, T\}^n$ be a distinct relaying state, i.e., a particular configuration of listening and transmitting states for

all the relays. The fraction of time the relays spend in state m_i will be denoted by p_i, where $\sum_{i\in[2^n]} p_i = 1$. We will use $L(m_i), T(m_i) \subseteq [n]$ to denote the set of indices of the relays in listening and transmitting state in m_i, respectively. Also, for $j \in [2^n]$, $\Lambda_j \subseteq [n]$ denotes the cut separating $\mathcal{S} \cup (\cup_{k\in\Lambda_j} \mathcal{R}_k)$ from $\mathcal{D} \cup (\cup_{k\in\bar{\Lambda}_j} \mathcal{R}_k)$. A representative cut is shown in Fig. 4.1b

The term l-value(s) and r-value(s) will refer to the l_k's and r_k's, respectively. Finally, unless otherwise stated, the term "constant" will mean a quantity that depends only on the number of relays and is independent of the channel SNRs.

4.2.2 An Approximation to the Capacity

Let C^n_{hd} denote the capacity of the n-relay half-duplex diamond network. As described in Chapter 2, C^n_{hd} can be approximated, upto constant additive terms, by C^n_{lp} which is the optimum solution of the following linear program.

$$\begin{aligned} &\textbf{LP}: \text{Maximize } C \\ &\sum_{i=1}^{2^n} p_i \left(\max_{k\in\Lambda_j\cap L(m_i)} l_k + \max_{k\in\Lambda_j\cap T(m_i)} r_k \right) \geq C \text{ for each } j \in [2^n] \\ &\sum_{i=1}^{2^n} p_i = 1;\ \forall i, p_i \geq 0, C \geq 0 \end{aligned} \tag{4.2}$$

Each primal variable p_i denotes the fraction of time spent by the relays in state m_i and each constraint (except the last one) corresponds to a distinct cut Λ_j.

4.2.3 Independent Switching at Relays

At a local level, each relay can control when it switches from an L state to a T state and vice-versa, and it can do so multiple times within the duration of operation. We normalize the duration to unity and hence, all the switches are made at points in the interval $[0, 1]$. For purposes of counting, a switch will always denote a *transition from L to T*. We will also assume that each relay always starts in a L state and ends in a T state. Thus, if a relay makes σ switches, there will be σ transitions from L to T and $\sigma - 1$ transitions from T to L. Choosing a local switching strategy then amounts to choosing $2\sigma - 1$ points on the unit interval for each relay independently (see Fig. 4.2). For $\mathcal{R}_k$, let the points (in ascending order) at which the transitions from L to T happen be denoted by $p^{L\to T}_{k,1}, \ldots, p^{L\to T}_{k,\sigma}$ and let the points (in asc. order) at which the transitions from T to L happen be denoted by $p^{T\to L}_{k,1}, \ldots, p^{T\to L}_{k,\sigma-1}$. Together, they define the *switching sequence* for $\mathcal{R}_k$, which is denoted by $S_k(\sigma)$.

$$S_k(\sigma) = \{p^{L\to T}_{k,1}, p^{T\to L}_{k,1}, \ldots, p^{T\to L}_{k,\sigma-1}, p^{L\to T}_{k,\sigma}\} \tag{4.3}$$

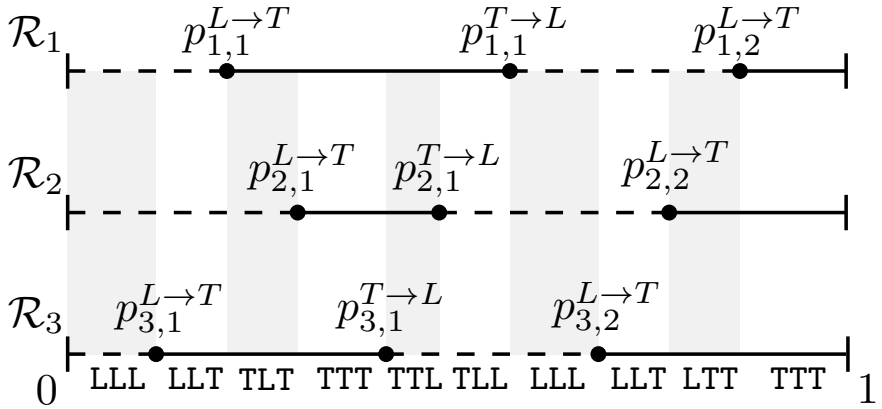


Figure 4.2 – Example illustration of randomized switching and induced states in a 3-relay network with each relay using 2 switches. The dashed portions denote the L state.

For ease of notation, we set $p_{k,\sigma}^{T\to L} = 1$ and $p_{k,0}^{T\to L} = 0$ for all $k \in [n]$. We use $S(\sigma)$ to denote the union of all the switching sequences, i.e., $S(\sigma) = \cup_{k\in[n]}\{S_k(\sigma)\}$. For a given $S(\sigma)$, the (approximate) rate achieved by the network is denoted by $C^n(S(\sigma))$. We focus on *randomized switching*, i.e. each relay $\mathcal{R}_k$ chooses the positions in $S_k(\sigma)$ randomly. In practice, this can be thought of as a pseudorandom pattern that is shared with the destination. Note that the above strategy is linear (in n) in terms of the *state complexity*, i.e., the number of relaying states with non-zero probabilities. More precisely, the number of active states for σ switches is at most $\min\{2^n, n(2\sigma-1)+1\}$.

For a given switching sequence $S(\sigma)$, the total time spent by $\mathcal{R}_k$ in state L (denoted by $P_{k,L}(S(\sigma))$) and in state T (denoted by $P_{k,T}(S(\sigma))$), can then be computed as

$$P_{k,L}(S(\sigma)) = \sum_{s=1}^{\sigma}(p_{k,s}^{L\to T} - p_{k,s-1}^{T\to L}) \tag{4.4}$$

$$P_{k,T}(S(\sigma)) = \sum_{s=1}^{\sigma}(p_{k,s}^{T\to L} - p_{k,s}^{L\to T}) \tag{4.5}$$

4.3 Local Scheduling Strategy

Achieving C_{lp}^n requires global knowledge of the link strengths in order to solve the optimization problem in (4.2) and determine the fraction of time spent in each scheduling state m_i. In practice, this may be expensive and will require inter-relay communication as well as a central node (eg. the source) that performs the optimization. The optimization itself consists of $2^n + 1$ variables and $2^n + 1$ constraints; solving it explicitly becomes prohibitive even for moderately large values of n. Instead, as a more practical approach, we look at what can be achieved by using only local information at the relays; we assume that each relay only knows the channel strengths of its incoming and outgoing links, i.e.,

$\mathcal{R}_k$ knows l_k and r_k.

4.3.1 Local Optimality Criterion

We have each $\mathcal{R}_k$ listen and transmit for an (overall) fraction of time that is *optimal* for an isolated single half-duplex relay $\mathcal{R}_k$ (essentially a one-hop line network). In the absence of any other information about the strengths of links connecting the other relays, this is a reasonable strategy to follow. It is easy to see that for $\mathcal{R}_k$, the optimal listening and transmitting fractions that our strategy should choose are as follows

$$P_{k,L}(S(\sigma)) = \frac{r_k}{l_k + r_k} \text{ and } P_{k,T}(S(\sigma)) = \frac{l_k}{l_k + r_k} \tag{4.6}$$

The quantity we will be interested in is the *expected rate* achieved by the network for σ switches $C^n_{rnd}(\sigma) = \mathbb{E}[C^n(S(\sigma))]$, where the expectation is taken over all the random choices of $S_k(\sigma)$'s that satisfy the criterion in (4.6).

4.3.2 Varying the Number of Switches

We will also analyze the performance of our strategy as we progressively increase the number of switches that each relay employs. In that regard, we have the following extremes cases:

Deterministic Switching When $\sigma = 1$, the set $S(\sigma)$ is uniquely determined. Each $\mathcal{R}_k$ makes only one switch from L to T at the point $\frac{r_k}{l_k+r_k}$ (see Fig. 4.3 for illustration). Notice that this corresponds to a *deterministic* switching strategy and hence we denote the achieved rate by $C^n_{det} \equiv C^n_{rnd}(1)$.

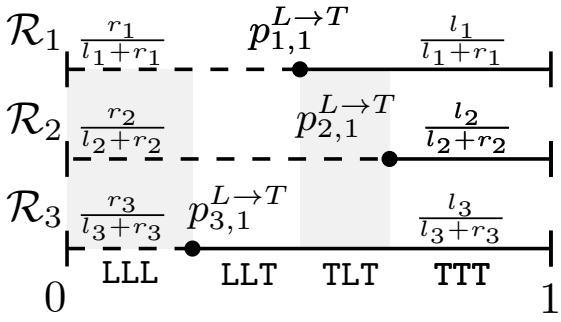


Figure 4.3 – Deterministic switching and induced states in a 3 relay network. The local L and T fractions are also shown.

Continuous Switching In the limit of σ becoming very large, the relays switch in a manner such that at each instant, $\mathcal{R}_k$ is in state L with probability $\frac{r_k}{l_k+r_k}$ and in state T with probability $\frac{l_k}{l_k+r_k}$. We denote the limiting rate by $C^n_{lim} \equiv \lim_{\sigma\to\infty} C^n_{rnd}(\sigma)$. Table 4.1 summarizes the quantities defined in this section.

σ	Number of switches from L to T in each relay.
C^n_{lp}	Approximate capacity of the network.
$C^n_{rnd}(\sigma)$	Expected rate achieved for σ switches.
C^n_{det}	Rate achieved for $\sigma = 1$.
C^n_{lim}	Rate achieved in the limit of large σ.
C^n_{lploc}	Upper bound to rates for strategies satisfying (4.6).

Table 4.1 – Summary of quantities considered.

4.3.3 Upper Bound to Local Strategies

In order to further understand the performance limits of our local randomized switching strategy, we also look at an *upper bound* to the rates achieved by any strategy that follows the local optimality criterion (4.6). The optimal rate so obtained is denoted by C^n_{lploc}. This can be computed by adding the following constraints to the **LP** (4.2) for each relay $\mathcal{R}_k$.

$$\sum_{i:k\in L(m_i)} p_i = \frac{r_k}{l_k+r_k} \forall k \in [n] \tag{4.7}$$

The quantity on the left of (4.7) represents the total fraction of time $\mathcal{R}_k$ is in state L for a schedule $\{p_i\}_{i\in[2^n]}$. The new linear program is as follows.

$$\begin{aligned}
&\textbf{LPLOC}: \text{Maximize } C \qquad (4.8)\\
&\sum_{i=1}^{2^n} p_i \left(\max_{k\in\Lambda_j\cap L(m_i)} l_k + \max_{k\in\Lambda_j\cap T(m_i)} r_k \right) \geq C \text{ for each } j \in [2^n]\\
&\sum_{i:k\in L(m_i)} p_i = \frac{r_k}{l_k+r_k} \forall k \in [n]\\
&\sum_{i=1}^{2^n} p_i = 1;\ \forall i, p_i \geq 0, C \geq 0
\end{aligned}$$

Thus, the optimum of C^n_{lploc} of **LPLOC** represents the maximum rate achievable by strategies (not necessarily without coordination) that follow (4.6).

4.3.4 Computation of Rates

A particular switching sequence $S(\sigma)$ uniquely induces a *global schedule* $\{p_i(S(\sigma))\}_{i\in[2^n]}$. The fraction of time the network is in state $m_i \in \{L,T\}^n$ is the union of (possibly disjoint) intervals where each $\mathcal{R}_k$ is in the state implied by m_i. More formally,

$$p_i(S(\sigma)) = \left| \bigcup_{r_1,\ldots,r_n\in[\sigma]^n} \left\{ \bigcap_{k\in L(m_i)} [p_{k,r_k-1}^{T\to L}, p_{k,r_k}^{L\to T}] \bigcap_{k\in T(m_i)} [p_{k,r_k}^{L\to T}, p_{k,r_k}^{T\to L}] \right\} \right| \tag{4.9}$$

where the union is over all possible σ^n n-tuples in $[\sigma]^n$. In practice, each term $p_i(S(\sigma))$ can be computed efficiently by first sorting all the points (irrespective of the relay index) in $S(\sigma)$ and then traversing the sorted sequence, keeping track of the relay states. Once this is accomplished, $C^n(S(\sigma))$ can be computed by performing a min-cut computation as follows:

$$C^n(S(\sigma)) = \min_{j\in[2^n]} \sum_{i=1}^{2^n} p_i(S(\sigma)) \left(\max_{k\in\bar{\Lambda}_j\cap L(m_i)} l_k + \max_{k\in\Lambda_j\cap R(m_i)} r_k \right) \tag{4.10}$$

The expected rate $C_{rnd}^n(\sigma)$ can then be computed numerically by taking a large enough sample of random $S(\sigma)$'s.

The above discussion trivially holds for C_{det}^n. For C_{lim}^n, as the random switches at each relay occur independently of each other, in the limit of large σ as discussed in Section 4.3.2, the fraction of time spent in a state m_s is given by:

$$p_i^{lim} = \prod_{k\in L(m_i)} \frac{r_k}{l_k + r_k} \prod_{k\in T(m_i)} \frac{l_k}{l_k + r_k} \tag{4.11}$$

C_{lim}^n can then be computed by setting $p_i(S(\sigma)) = f_s^{lim}$ in (4.10).

In the following sections, we present results that illustrate the performance of our local random switching strategy.

4.4 Performance over the 2-relay network

For brevity, in this section we use the following substitution for the 2-relay network: $a \leftarrow l_1, b \leftarrow l_2, c \leftarrow r_1, d \leftarrow r_2$.

4.4.1 Comparison with C_{lp}^n - Lower Bounds

For $n = 2$ relays, the linear program for C_{lp}^n can be solved to obtain a closed form expression [30]. Four cases arise depending on whether $a \geq b$, $c \geq d$ and the value of $\delta = ab - cd$. We will only show the proofs for the case $a \geq b$, $c \leq d$ and $\delta \leq 0$; the other

cases being similar. For this case, we have (from [30]):

$$C_{lp}^2 = \frac{ac(b+d) + bd(a-b)}{(b+d)(a+c-b)} \tag{4.12}$$

Using this expression, we can show the following lower bound on the performance of local continuous switching.

Theorem 4.4.1 *For a 2-relay half-duplex diamond network,*

$$\frac{C_{lim}^2}{C_{lp}^2} \geq \frac{3}{4} \tag{4.13}$$

Proof: For deriving the expression for C_{lim}^2, we order the relaying states as $\{LL, LT, TL, TT\}$ and let the corresponding time fractions be p_1, p_2, p_3, p_4. From the previous section, for the limiting (continuous) random local schedule, we have

$$\begin{aligned} p_1 &= \frac{cd}{(a+c)(b+d)}, p_2 = \frac{cb}{(a+c)(b+d)} \\ p_3 &= \frac{ad}{(a+c)(b+d)}, p_4 = \frac{ab}{(a+c)(b+d)} \end{aligned} \tag{4.14}$$

By definition

$$\begin{aligned} C_{lim}^2 = \min\{&ap_1 + ap_2 + bp_3, ap_1 + (a+d)p_2 + dp_4, bp_1 + (b+c)p_3 + cp_4, \\ &dp_2 + cp_3 + dp_4\} \end{aligned} \tag{4.15}$$

which, for our case, simplifies to

$$C_{lim}^2 = \frac{a(bc+bd+cd)}{(a+c)(b+d)} \qquad \text{if } a \leq d \tag{4.16}$$

$$= \frac{d(ab+ac+bc)}{(a+c)(b+d)} \qquad \text{if } a > d \tag{4.17}$$

In the case $a \leq d$, showing $C_{lim}^2 \geq \frac{3}{4} C_{lp}^2$ is equivalent to

$$4a(a-b+c)(cd+b(c+d)) - 3(a+c)\left(-b^2 d + a(cd+b(c+d))\right) \geq 0 \tag{4.18}$$

Denoting the l.h.s by ϕ, the following inequalities hold.

$$\begin{aligned} \phi/c &\overset{(i)}{\geq} a^2 b - 4ab^2 + abc + a^2 d - 3abd + 3b^2 d + acd \\ &\overset{(ii)}{\geq} -3ab^2 + abc + a^2 d - 3abd + 3b^2 d + acd \end{aligned}$$

$$\overset{(iii)}{\geq} -3ab^2 + a^2b^2/d + a^2d - 3abd + 3b^2d + a^2b$$
$$= \frac{1}{d}(a^2(b^2 + d^2 + bd) + a(-3b^2d - 3bd^2) + 3b^2d^2)$$

(i) and (ii) hold as $a \geq b$; (iii) holds because $c \geq ab/d$. The numerator in the last quantity is a quadratic expression in the variable a and its discriminant is

$$\Delta = 9(bd^2 + b^2d)^2 - 12b^2d^2(b^2 + d^2 + bd) = -3b^2(b-d)^2d^2 \leq 0 \tag{4.19}$$

Therefore, $\phi \geq 0$, which establishes our claim. Clearly, equality is attained when a, b, c, d are equal. The proof for $a > d$ follows similarly. ■

For deterministic switching, the worst case ratio drops to $1/2$, as shown below.

Theorem 4.4.2 *For a 2-relay half-duplex diamond network,*

$$\frac{C^2_{det}}{C^2_{lp}} \geq \frac{1}{2} \tag{4.20}$$

Proof: To derive the expression for C^2_{det}, notice that relay 1 listens for $\frac{c}{a+c}$ of time and relay 2 listens for $\frac{d}{b+d}$ fraction. Depending on which one is larger, we will have the states $\{LL, TL, TT\}$ or $\{LL, LT, TT\}$. For the first case, we have

$$p_1 = \frac{c}{a+c}, p_2 = 0, p_3 = \frac{ad - bc}{(a+c)(b+d)}, p_4 = \frac{b}{b+d} \tag{4.21}$$

Using these, we can derive

$$C^2_{det} = \frac{-b^2c + a(cd + b(c+d))}{(a+c)(b+d)} \qquad \text{if } a + c \leq b + d \tag{4.22}$$
$$= \frac{acd + b\left(-c^2 + ad + cd\right)}{(a+c)(b+d)} \qquad \text{if } a + c \geq b + d \tag{4.23}$$

For the first case, proving our claim is equivalent to showing

$$2\left(-b^2c + a(cd + b(c+d))\right)(a + c - b) \tag{4.24}$$
$$-(a+c)(ac(b+d) + bd(a-b)) \geq 0 \tag{4.25}$$

Denoting the l.h.s by ϕ, the following inequalities hold.

$$\phi/c = -3ab^2 + 2b^3 - b^2c + a^2d - abd + b^2d + acd + (a-b)(ab + bc)$$
$$\overset{(i)}{\geq} -3ab^2 + 2b^3 + a^2d - abd + b^2(d-c) + acd$$

$$\overset{(ii)}{\geq} a^2(b+d) - a(3b^2+bd) + 2b^3 = (a-b)(ab+ad-2b^2)$$
$$\overset{(iii)}{\geq} (a-b)(b(a-b) + b^{\frac{1}{2}}(a^{\frac{3}{2}} - b^{\frac{3}{2}}))$$

Here, (i) is true because $a \geq b$, (ii) is true because $c \geq d$ and $cd \geq ab$ and finally (iii) is true because $d^2 \geq cd \geq ab \implies d \geq \sqrt{ab}$. Again, equality holds when a, b, c, d are equal. The proof for the other three cases is similar. ■

For a finite values of $\sigma > 1$, it is more difficult to get closed form expressions and prove analytical lower bounds. In the case of $\sigma = 2$, numerical evidence suggests that the lowest value of $C^2_{rnd}(2)/C^2_{lp}$ is 0.7 and it is attained when $\{a, b, c, d\}$ are all equal. In fact, we show the following lemma.

Lemma 4.4.3 *In a 2-relay half-duplex diamond network, when $a = b = c = d$, then the following holds.*

$$\frac{C^2_{rnd}(2)}{C^n_{lp}} = \frac{7}{10} \tag{4.26}$$

Proof: The result follows from a lengthy case by case analysis of different configurations of the relaying states and is given in the Appendix A.1. ■

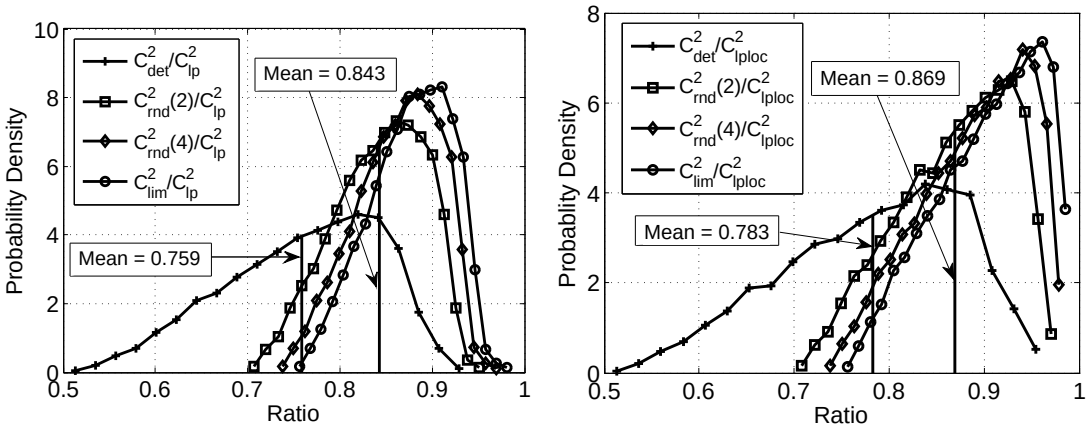
4.4.2 Comparison with C^n_{lploc} - Lower Bounds

C^2_{lploc} represents the upper bound of the rates achievable by strategies that adhere to the local optimality criterion (4.6). It is interesting to see how our switching strategies perform with respect to this bound. We reiterate that C^n_{lploc} also encompasses strategies that allow inter-relay communication, and hence is strictly an upper bound for the types of *distributed* scheduling strategies we propose, albeit a tighter one than C^n_{lp}.

For $n = 2$ relays, the linear program for C^n_{lploc} can be solved to obtain a closed form expression. It is as follows

$$C^2_{lploc} = \frac{b(d+a)}{b+d} \text{ if } ab \leq cd, a \leq b \tag{4.27}$$
$$= \frac{a(b+c)}{a+c} \text{ if } ab \leq cd, a \geq b \tag{4.28}$$
$$= \frac{c(a+d)}{a+c} \text{ if } ab \geq cd, c \geq d \tag{4.29}$$
$$= \frac{d(b+c)}{b+d} \text{ if } ab \geq cd, c \leq d \tag{4.30}$$

Since C^2_{lploc} is the optimum of **LPLOC**, which is a more constrained version of **LP**,



(a) Ratio with C^2_{lp} (b) Ratio with C^2_{lploc}

Figure 4.4 – Numerical evaluations for the 2 relay network with channel strengths sampled uniformly and independently from $[0, 30]$ dB

$C^2_{lploc} \leq C^2_{lp}$. The following lemma is a direct consequence of Thm. 4.4.1 and Thm. 4.4.2.

Lemma 4.4.4

$$\frac{C^2_{lim}}{C^2_{lploc}} \geq \frac{3}{4} \; and \; \frac{C^2_{det}}{C^2_{lploc}} \geq \frac{1}{2} \tag{4.31}$$

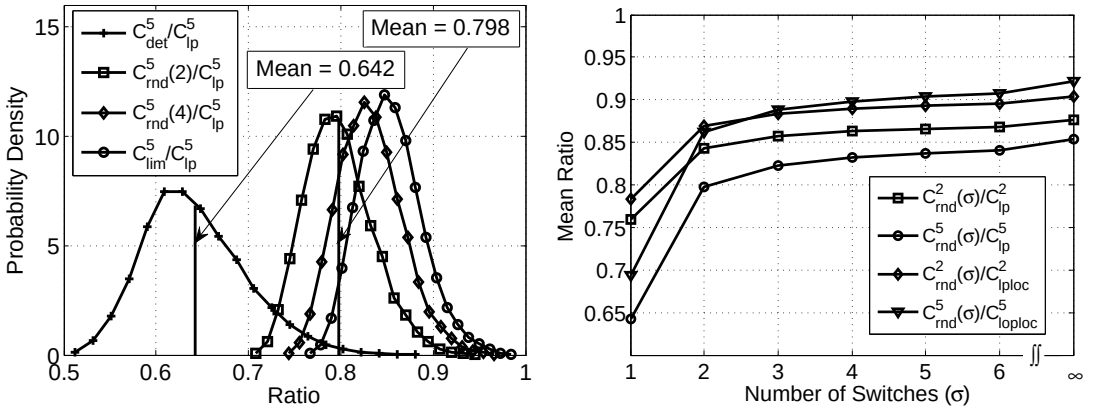
In fact, these are the best bounds that can be obtained because for the case $a = b = c = d$, we have

$$\frac{C^2_{lim}}{C^2_{lploc}} = \frac{3}{4}, \; \frac{C^2_{det}}{C^2_{lploc}} = \frac{1}{2} \tag{4.32}$$

4.4.3 Numerical Evaluation

The above two theorems show that the worst case performance of randomized switching (in the limit of large σ) is much better than that of deterministic switching. For finite values of $\sigma > 1$, the worst case value of $C^2_{rnd}(\sigma)/C^2_{lp}$ is likely to lie between the two extremes of 0.5 and 0.75. To investigate its performance, we perform the following simulation. We select the channel strengths for the four links in the network, independently and uniformly at random, in the range $[0, 30]$ dB. For each configuration, we compute the quantities C^2_{det}/C^2_{lp}, C^2_{lim}/C^2_{lp} and $C^2_{rnd}(\sigma)/C^2_{lp}$ for $\sigma = 2, 4$ and then plot the p.d.f of these quantities as shown in Fig. 4.4a.

The plot shows that there is a significant jump (about 11%) in the average performance



(a) Ratio with C^5_{lp}

(b) Mean of $C^n_{rnd}(\sigma)/C^n_{lp}$ and $C^n_{rnd}(\sigma)/C^n_{lploc}$ for $\sigma = 1, 2, 3, 4, 5, 6, \infty$ and $n = 2, 5$

Figure 4.5 – Numerical evaluations for 5-relay network and variation with σ

of $C^2_{rnd}(2)/C^2_{lp}$ (0.843) over C^2_{det}/C^2_{lp} (0.759). Thereafter, as we increase σ, the average performance slowly saturates to 0.877, which is the mean of C^2_{lim}/C^2_{lp}. Thus, randomization and increasing the number of switches increases the average performance by about 15.5%, a large chunk of which (approximately **71**% of the difference) is leveraged by using two random switches

For comparison with C^2_{lploc}, it is difficult to obtain closed form bounds for $C^2_{rnd}(\sigma)/C^2_{lploc}$ for finite $\sigma > 1$. When the simulations are repeated for the ratios C^2_{det}/C^2_{lploc}, C^2_{lim}/C^2_{lploc} and $C^2_{rnd}(\sigma)/C^2_{lploc}$ for $\sigma = 2, 4$, trends as shown in Fig. 4.4b is observed. Although the worst case ratios are the same as those with respect to C^2_{lp}, the average performance is better.

We observe a jump in mean performance from 0.783 to 0.869 (an increase of 11%) when we increase σ from 1 to 2. Again, this forms the major chunk of increase in performance due to randomization. Interestingly, this also shows that our randomized strategy with just two switches achieves, on an average, about 86.9% of the maximum rate achievable by any strategy with the local optimality condition, even ones that use global CSI.

4.5 Performance over larger networks

For diamond networks of larger size, the performance trends observed in the previous section are essentially similar with a few interesting caveats. Fig. 4.5a plots the p.d.f of C^5_{det}/C^5_{lp}, C^5_{lim}/C^5_{lp} and $C^5_{rnd}(\sigma)/C^5_{lp}$ for $\sigma = 2, 4$ for random instances of a 5-relay network. In this case, the gain of mean performance going from $\sigma = 1$ to $\sigma = 2$ is significantly more: $0.642 \to 0.798$–an increase of about **24.3**%. This also shows that deterministic switching performs worse for larger number of relays, but even $\sigma = 2$ greatly

boosts performance.

In Fig. 4.5b, we plot the mean performance ratios of different schemes as a function of the number of switches for $n = 2, 5$ relays. This plot essentially conveys three messages: (i) C^n_{lploc} is a more useful outer bound than C^n_{lp} when comparing local scheduling strategies; (ii) increasing the number of switches has highly diminishing returns for larger σ and very quick saturation towards the asymptotic value is observed, which is practically important as too many switches can have significant network overhead, and (iii) Local CSI helps over not using any CSI: the performance of $C^n_{nsi}(\sigma)$, which incorporates σ random switches *without* respecting (4.6) performs significantly worse, especially for small σ.

5 Relay Selection in Full-Duplex Layered Networks

In the previous three chapters we looked at three different ways to reduce the complexity of relaying in half-duplex diamond networks. In this chapter we turn our attention to full-duplex layered relay networks. In such networks, a source communicates with the destination using relays that are arranged in L layers and a relay in layer l forwards information to relays in layer $l+1$. Using all the relays can be wasteful in terms of resources and can lead to high communication complexity for large networks.

In this chapter, we consider the following problem – how to select the subset of relays of a given size K that has the highest capacity in a computationally efficient manner? In a sense, we are interested in generalizing routing over physical layer cooperation networks. In routing, we select the best one or best K path(s) over which to forward the information from a source to a destination; over physical layer cooperation networks, the corresponding operation would be selecting the best subnetwork (of a given size).

Using approximate capacity expressions similar to the one used in Chapter 2, we represent subnetwork selection as an integer optimization problem – to every relay $\mathcal{R}$ in the network, we assign a *binary* selection variable $\theta_{\mathcal{R}}$ that takes value 1 if $\mathcal{R}$ is selected and 0 if it is not. The objective (approximate capacity expression) can then be represented as a function of $\theta_{\mathcal{R}}$, which then needs to be maximized. To efficiently solve the resulting integer program, we first relax the constraint of $\theta_{\mathcal{R}}$ being a binary variable to it being a continuous variable $\theta_{\mathcal{R}} \in [0,1]$. Using properties of submodular functions and convex optimization, we show that the relaxed program is polynomial time solvable for diamond networks. The fractional optimal solutions are then rounded to obtain a feasible integer solution.

We present numerical evaluations of the performance of our algorithm on networks with random channel sets that show its superior accuracy and efficiency. In particular, the algorithm achieves an accuracy of more than 98% of the integer optimum value with a probability of 0.97 for networks of 20 relays and takes time that is less than that of an exhaustive integer optimization by factors of more than 450 for networks of 30 relays.

5.1 Related Work

Previous work on relay selection in wireless networks can be divided into three categories: (i) The work of [35] and the references therein propose algorithms to select the *single best* relay (in terms of cooperative diversity) in one-layer networks. (ii) The work of [37] and the references therein analyse the performance of heuristics for selecting a subset of relays (again, from a single layer of relays) for Amplify-and-Forward (AF) based protocols. (iii) Recent work by [39] proves upper bounds on multiplicative and additive gaps for AF-based relay selection, primarily for diamond networks. The work of [10] proves general multiplicative lower bounds on the the rate achievable by a subset of relays in a diamond network.

5.2 Problem Formulation

5.2.1 Communication Model

We consider a full-duplex layered wireless network $\mathcal{W}$ containing L layers of single-antenna nodes. The source is the singleton node in layer 0, while the destination is the singleton node in layer $L-1$. For ease of exposition, all the intermediate layers are assumed to have exactly n nodes, although our techniques can handle any configuration. The total number of relays is then $N = n(L-2)$. As shown in Fig. 5.1, each signal path from the source to the destination in a layered network gets relayed by exactly the same number of hops. The signal flow over this network can then be written as:

$$Y_i^{l+1} = \sum_{j=1}^{n} h_{ij}^l X_j^l + Z_i^{l+1} \tag{5.1}$$

where Y_i^{l+1} denotes the received signal at node $i \in [1:n]$ in layer $l+1$ ($l \in [0:L-2]$), h_{ij}^l denotes the complex channel coefficient from node j in layer l to node i in layer $l+1$, X_j^l denotes the transmitted signal from node $j \in [1:n]$ in layer l and Z_i^{l+1} denotes the i.i.d zero mean complex Gaussian noise at the receiver i in layer $l+1$. For our network, we have a *per node* power constraint, given by $\mathbb{E}[|X_j^l|^2] \leq 1$. We also normalize the noise powers to unity, i.e., $Z_i \sim$ i.i.d $\mathcal{CN}(0,1)$. Notice that the *per node* signal flow equations can be coalesced into *per layer* equations as:

$$\mathbf{Y}^{l+1} = \mathbf{H}^l \mathbf{X}^l + \mathbf{Z}^{l+1} \tag{5.2}$$

where $\mathbf{Y}^{l+1} = [Y_1^{l+1}, \ldots, Y_n^{l+1}]^T$, $\mathbf{X}^l = [X_1^l, \ldots, X_n^l]^T$, $\mathbf{H}^l$ is the MIMO channel matrix from $\mathbf{X}^l$ to $\mathbf{Y}^{l+1}$ with $\mathbf{H}^l(i,j) = h_{ij}^l$ and $\mathbf{Z}^{l+1} = [Z_1^{l+1}, \ldots, Z_n^{l+1}]^T$.

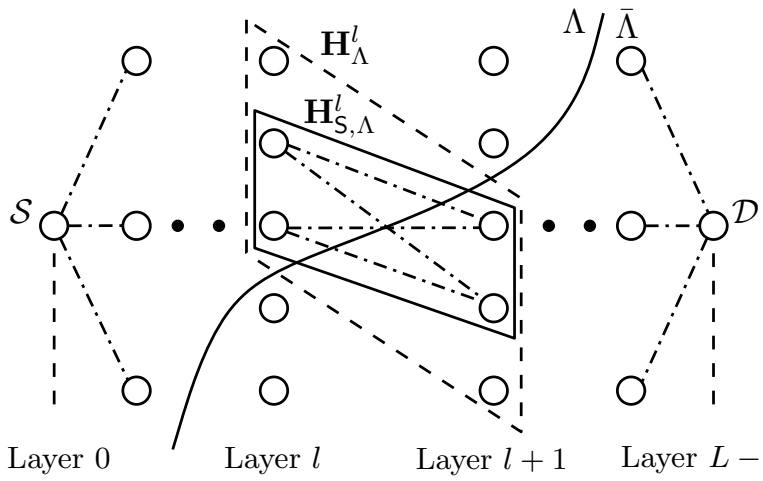


Figure 5.1 – The Gaussian full-duplex layered network with L layers having n relays each, except the first and last one.

5.2.2 Capacity Outer bounds and Rate Expressions

Since the capacity of such networks cannot be characterized exactly, approximate expressions (similar to the ones used in Chapter 2) that are a constant gap away from capacity are often used as a metric to evaluate the performance of a given relaying protocol over such networks.

A standard practice is to use inputs X_j at every network node that are picked from an i.i.d complex Gaussian distribution, i.e., $X_j \sim$ i.i.d $\mathcal{CN}(0,1)$. This is precisely the strategy used to prove the constant gap performance of the QMF and NNC schemes in [8], [44], and in view of these results, the following modified version of the cutset upper bound, termed $\bar{C}_{\mathsf{iid}}$ in [8], with the above-mentioned inputs is of interest:

$$\bar{C}_{\mathsf{iid}} = \min_{\Lambda} \sum_{l=0}^{L-1} \log\det\left(I + \mathbf{H}^l_\Lambda \mathbf{H}^{l\dagger}_\Lambda\right) \tag{5.3}$$

Here, $\mathbf{H}^l_\Lambda$ denotes the MIMO channel matrix from $\mathbf{X}^l_\Lambda$ to $\mathbf{Y}^{l+1}_{\bar{\Lambda}}$ and Λ denotes a *cut* in the network as shown in Fig. 5.1. For uniformity in dimensions, we choose to represent $\mathbf{X}^l_\Lambda$, $\mathbf{Y}^{l+1}_{\bar{\Lambda}}$ and $\mathbf{H}^l_\Lambda$ as $n \times 1$, $n \times 1$ and $n \times n$ matrices respectively, by inserting zeroes in the appropriate rows and columns as dictated by the index of elements in Λ or $\bar{\Lambda}$.

5.2.3 Subnetwork Selection

We want to select an optimal subnetwork $\mathcal{W}_S^{\mathsf{opt}}$ of $\mathcal{W}$ that maximizes the subnetwork's $\bar{C}_{\mathsf{iid}}$ expression over all subnetworks $\mathcal{W}_S$ with the following size constraint: $\mathcal{W}_S$ contains K_l relays ($K_l \geq 1$) in layer $l \in [1 : L-2]$. For this purpose, we first define a set of $n \times n$ diagonal selection matrices, $\mathbf{S}_l$, for each layer $l \in [1 : L-2]$ of the form $\mathbf{S}_l = \mathrm{diag}(\sqrt{\theta_{l1}}, \sqrt{\theta_{l2}}, \ldots, \sqrt{\theta_{ln}})$ such that $\theta_{li} \in \{0,1\}$ and $\sum_{i=1}^n \theta_{li} = K_l$. For a given

subnetwork, the matrices $\mathbf{S}_l$ will have $\theta_{li} = 1$ iff relay $i \in [1:n]$ in layer l is in the subnetwork and $\theta_{li} = 0$ otherwise.

For a given $\mathcal{W}_S$, the subnetwork's $\bar{C}_{\mathsf{iid}}$ is given as:

$$\bar{C}_{\mathsf{S},\mathsf{iid}} = \min_{\Lambda} I_{\mathsf{S},\Lambda} \tag{5.4}$$

where

$$I_{\mathsf{S},\Lambda} = \sum_{l=0}^{L-1} \log\det\left(I + \mathbf{H}^l_{\mathsf{S},\Lambda}\mathbf{H}^{l\dagger}_{\mathsf{S},\Lambda}\right) \tag{5.5}$$

Here, $\mathbf{H}^l_{\mathsf{S},\Lambda}$ denotes the (still $n \times n$) MIMO channel matrix from $\mathbf{X}^l_{\mathsf{S},\Lambda}$ to $\mathbf{Y}^{l+1}_{\mathsf{S},\bar{\Lambda}}$ where $\mathbf{X}^l_{\mathsf{S},\Lambda} = \mathbf{S}_l\mathbf{X}^l_{\Lambda}$ and $\mathbf{Y}^{l+1}_{\mathsf{S},\bar{\Lambda}} = \mathbf{S}_{l+1}\mathbf{Y}^{l+1}_{\bar{\Lambda}}$. In this setting, $\mathbf{H}^l_{\mathsf{S},\Lambda}$ can be related to $\mathbf{H}^l_{\Lambda}$ as:

$$\mathbf{H}^l_{\mathsf{S},\Lambda} = \begin{cases} \mathbf{S}_l\mathbf{H}^l_{\Lambda}\mathbf{S}_{l+1}, & l \in [1:L-2] \\ \mathbf{H}^0_{\Lambda}\mathbf{S}_1, & l = 0 \\ \mathbf{S}_{L-1}\mathbf{H}^{L-1}_{\Lambda}, & l = L-1 \end{cases} \tag{5.6}$$

Essentially, $\mathbf{H}^l_{\mathsf{S},\Lambda}$ is obtained from $\mathbf{H}^l_{\Lambda}$ by replacing with 0, the rows (resp. columns) indexed by the relays in layers l (resp. $l+1$) that are not selected. This way, we retain an $n \times n$ channel matrix at each layer l that is equivalent in terms of singular values to the $K_l \times K_{l+1}$ channel matrix at that layer.

Now, the problem of finding $\mathcal{W}^{\mathsf{opt}}_S$ essentially reduces to optimally selecting the set of matrices $\{\mathbf{S}_l\}_{\{l\in[1:L-2]\}}$. This integer optimization problem can be stated as:

$$\{\mathbf{S}^{\mathsf{opt}}_l\}_{\{l\in[1:L-2]\}} = \arg \max_{\substack{\{\mathbf{S}_l\}_{\{l\in[1:L-2]\}} \\ \mathbf{tr}(\mathbf{S}_l^2)=K_l}} \bar{C}_{\mathsf{S},\mathsf{iid}}\left(\{\mathbf{S}_l\}\right) \tag{5.7}$$

where the trace condition is equivalent to $\sum_{i=1}^{n}\theta_{li} = K_l$.

5.3 Relaxed Approximation – Diamond Networks

We first illustrate our relaxation approach for (approximately) solving the above optimization problem in (5.7) by taking the simplest example of an n-relay diamond network. In this network, there is a set of n relays in layer 1, while layer 0 and layer 2 contain the source and destination nodes respectively. For this network, we essentially have only 1 selection matrix, $\mathbf{S}_1 = \mathrm{diag}(\sqrt{\theta_{11}}, \sqrt{\theta_{12}}, \ldots, \sqrt{\theta_{1n}})$ corresponding to the relays in layer 1 that we have to optimize.

For purposes of simplification, we make the following abuse of notation in the remainder

of this section: (i) $\theta_{1i} \leftarrow \theta_i$ since there is only 1 layer of relays; (ii) $h^0_{i1} \leftarrow h^0_i$ to denote the channels from the source (indexed as node 1 in layer 0) to relay i; (iii) $h^1_{1i} \leftarrow h^1_i$ to denote the channels from relay i (in layer 1) to the destination (indexed as node 1 in layer 2).

Specializing (5.7) for the diamond network, where we wish to select the optimal subnetwork having K_1 relays, we have:

$$\{\theta_i^{\mathsf{opt}}\}_{i\in[1:n]} = \arg \max_{\substack{\{\theta_i\}_{i\in[1:n]}\in\{0,1\}:\\ \sum_{i=1}^n \theta_i = K_1}} \bar{C}^{\mathsf{dia}}_{\mathsf{S,iid}}\left(\{\theta_i\}\right) \tag{5.8}$$

where

$$\bar{C}^{\mathsf{dia}}_{\mathsf{S,iid}}(\{\theta_i\}) = \min_{\Lambda} \left\{ \log(1+\sum_{i\in\Lambda}\theta_i|h_i^0|^2) + \log(1+\sum_{i\in\bar{\Lambda}}\theta_i|h_i^1|^2) \right\} \tag{5.9}$$

5.3.1 Relaxing the Integer Program

For an approximate solution to the integer program in (5.8), we first *relax* the constraints in the problem as follows: Instead of using the integer θ_i's lying in the discrete set $\{0,1\}$, we replace them with *real* variables $\tilde{\theta}_i$'s that lie in the interval $[0,1]$. With this relaxation, the following theorem holds:

Theorem 5.3.1 *The optimization problem, defined as:*

$$\{\tilde{\theta}_i^{\mathsf{opt}}\}_{i\in[1:n]} = \arg \max_{\substack{\{\tilde{\theta}_i\}_{i\in[1:n]}\in[0,1]:\\ \sum_{i=1}^n \tilde{\theta}_i = K_1}} \bar{C}^{\mathsf{dia}}_{\mathsf{S,iid}}\left(\{\tilde{\theta}_i\}\right) \tag{5.10}$$

is a concave maximization *problem in* $\{\tilde{\theta}_i\}_{i\in[1:n]}$

Proof: Observe that the constraints on $\tilde{\theta}_i$ are *linear*. Hence, it remains to show that $\bar{C}^{\mathsf{dia}}_{\mathsf{S,iid}}$ is concave in $\{\tilde{\theta}_i\}_{i\in[1:n]}$. To this end, observe that for a given cut Λ, $(1+\sum_{i\in\Lambda}\tilde{\theta}_i|h_i^0|^2)$ and $(1+\sum_{i\in\bar{\Lambda}}\tilde{\theta}_i|h_i^1|^2)$ are *affine* functions of $\{\tilde{\theta}_i\}_{i\in[1:n]}$. Hence, $\log(1+\sum_{i\in\Lambda}\tilde{\theta}_i|h_i^0|^2)$ and $\log(1+\sum_{i\in\bar{\Lambda}}\tilde{\theta}_i|h_i^1|^2)$ are concave in $\{\tilde{\theta}_i\}_{i\in[1:n]}$, and so is their sum. Moreover, since the point wise minimum of concave functions is also concave, we can conclude that $\bar{C}^{\mathsf{dia}}_{\mathsf{S,iid}}$ is concave in $\{\tilde{\theta}_i\}_{i\in[1:n]}$, which proves the theorem. Notice here the significance of using a square root in the diagonal entries of the selection matrices, which lead to the affine functions inside the log terms. ■

Theorem 5.3.1 ensures the *existence* of a *polynomial time* algorithm (in the number of relays n) that solves the relaxed optimization problem in (5.10), for example using the

interior point method for concave maximization, provided there exists a polynomial time algorithm to find $\bar{C}^{\mathsf{dia}}_{\mathsf{S,iid}}$.

Finding $\bar{C}^{\mathsf{dia}}_{\mathsf{S,iid}}$ *a priori* consists of evaluating 2^n terms corresponding to the cuts Λ and then taking a minimum, which takes exponential time. However, it was shown in [33], that the terms inside the minimization of $\bar{C}^{\mathsf{dia}}_{\mathsf{S,iid}}$ are *submodular* in the sets Λ. Since submodular minimization can be accomplished using a polynomial (in n) number of evaluations of the mutual information terms [45], (5.10) can be solved in polynomial time.

5.3.2 Rounding the Relaxed $\tilde{\theta}_i$'s

Since the feasible set for (5.10) is a superset of (5.8), the relaxed optimal value will be greater. However, the $\{\tilde{\theta}^{\mathsf{opt}}_i\}_{i\in[1:n]}$ can have fractional values that do not correspond to an actual subnetwork of size K_1. The next step then is to *round* the fractional solution of (5.10) to a discrete solution that represents a subnetwork selection. Mathematically, a rounding is a map $f_R : \{\tilde{\theta}^{\mathsf{opt}}_i\}_{i\in[1:n]} \mapsto \{\theta^{\mathsf{sel}}_i\}_{i\in[1:n]}$ such that $\{\theta^{\mathsf{sel}}_i\}_{i\in[1:n]} \in \{0,1\}$ and $\sum_{i=1}^{n} \theta^{\mathsf{sel}}_i = K_1$. An intuitive way to round in this case would be to set $\theta^{\mathsf{sel}}_i = 1$ iff $\tilde{\theta}^{\mathsf{opt}}_i$ is among the maximum K_1 values in the set $\{\tilde{\theta}^{\mathsf{opt}}_i\}_{i\in[1:n]}$ and set $\theta^{\mathsf{sel}}_i = 0$ otherwise.

5.3.3 Applications in other capacity approximations

For n-relay diamond networks, a simpler and more approximate expression based on point-to-point link capacities has been proposed for capacity approximation in [10], given by:

$$\bar{C}^{\mathsf{dia}}_{\mathsf{P2P}} = \min_{\Lambda} \left\{ \max_{i\in\Lambda} \log(1+|h^0_i|^2) + \max_{i\in\bar{\Lambda}} \log(1+|h^1_i|^2) \right\} \tag{5.11}$$

The inherent advantage of working with (5.11) is that $\bar{C}^{\mathsf{dia}}_{\mathsf{P2P}}$ can be evaluated in $O(n \log n)$ time [10], which is faster than the polynomial time submodular minimization algorithms needed to evaluate $\bar{C}^{\mathsf{dia}}_{\mathsf{iid}}$ or $R^{\mathsf{dia}}_{\mathsf{NNC}}$. On the flip side, this approximation is not good for low SNRs and it does not generalize to multi-layered networks beyond the diamond topology. However, for diamond networks, we can still apply our relaxation framework on the $\bar{C}^{\mathsf{dia}}_{\mathsf{P2P}}$ expression to get a set of relays that (approximately) maximizes $\bar{C}^{\mathsf{dia}}_{\mathsf{P2P}}$ and see how that selected set of relays perform in terms of $\bar{C}^{\mathsf{dia}}_{\mathsf{iid}}$. In this case, the relaxed optimization problem can take the following form:

$$\{\tilde{\theta}_i^{\mathsf{opt}}\}_{i\in[1:n]} = \arg \max_{\substack{\{\tilde{\theta}_i\}_{i\in[1:n]}\in[0,1]: \\ \sum_{i=1}^n \tilde{\theta}_i = K_1}} \bar{C}^{\mathsf{dia}}_{\mathsf{S,P2P}} \left(\{\tilde{\theta}_i\}\right) \tag{5.12}$$

where

$$\bar{C}^{\mathsf{dia}}_{\mathsf{S,P2P}} \left(\{\tilde{\theta}_i\}\right) = \min_{\Lambda} \left\{ \begin{array}{l} \max_{i\in\Lambda} \tilde{\theta}_i \log(1+|h_i^0|^2) \\ +\max_{i\in\bar{\Lambda}} \tilde{\theta}_i \log(1+|h_i^1|^2) \end{array} \right\} \tag{5.13}$$

Note that unlike (5.10), (5.12) is *not* a concave optimization problem and in general, non-linear optimization algorithms can potentially get stuck in local maximas. Nevertheless, owing to the faster speed in computing $\bar{C}^{\mathsf{dia}}_{\mathsf{S,P2P}}$, it is worth giving this expression a try, and surprisingly, we show in Section 5.5 that off-the-shelf non-linear optimizers do give good results with the $\bar{C}^{\mathsf{dia}}_{\mathsf{S,P2P}}$ expression used for selection.

5.4 Relaxed Approximation – Multilayer Networks

For multilayer networks, the procedure is similar to the one outlined for diamond networks. The integer optimization problem over the $n(L-2)$ variables $\{\theta_{li}\}_{i\in[1:n],l\in[1:L-2]}$, as given in (5.7), can be relaxed to the corresponding continuous problem in $\{\tilde{\theta}_{li}\}$'s. Once the relaxed optimization problem is solved, the optimal fractional solution is rounded to an integer solution representing a subnetwork of appropriate size.

The objective function in the relaxed version of (5.7) is the minimum of $2^{n(L-2)}$ terms, each of which is a sum of terms of the form:

$$I^l_{\mathsf{S},\Lambda} = \log\det(I + \tilde{\mathbf{S}}_l \mathbf{H}^l_\Lambda \tilde{\mathbf{S}}^2_{l+1} \mathbf{H}^{l\dagger}_\Lambda \tilde{\mathbf{S}}_l) \tag{5.14}$$

where $\tilde{\mathbf{S}}_l = \mathrm{diag}(\sqrt{\tilde{\theta}_{l1}}, \ldots, \sqrt{\tilde{\theta_{ln}}})$, $\mathbf{tr}(\tilde{\mathbf{S}}^2_l) = K_l$ and $\mathbf{tr}(\tilde{\mathbf{S}}^2_{l+1}) = K_{l+1}$. In general, the above term is not a concave function of $\{\tilde{\theta}_{li}\}_{i\in[1:n]}$ and $\{\tilde{\theta}_{l+1,i}\}_{i\in[1:n]}$. Empirical evidence however suggests that it is *almost* concave.

Firstly, we denote by $\mathbf{x}$, the vector of variables

$$\mathbf{x} = \{\tilde{\theta}_{l1}, \ldots, \tilde{\theta}_{ln}, \tilde{\theta}_{l+1,1}, \ldots, \tilde{\theta}_{l+1,n}\} \tag{5.15}$$

Thus $I^l_{\mathsf{S},\Lambda}(\mathbf{x})$ defines a hyper-surface. For a fixed set of channel coefficients, if this surface is exactly concave, then for any pair of points $\mathbf{x}^1, \mathbf{x}^2 \in [0,1]^{2n}$ such that $\sum_{i=1}^n x_i^j = K_l$ and $\sum_{i=n+1}^{2n} x_i^j = K_{l+1}$ for $j = 1, 2$ and for every $\lambda \in [0,1]$, the following quantity must be always non-negative:

$$D(\mathbf{x}^1, \mathbf{x}^2, \lambda) = I^l_{\mathsf{S},\Lambda}(\lambda\mathbf{x}^1 + (1-\lambda)\mathbf{x}^2) - \lambda I^l_{\mathsf{S},\Lambda}(\mathbf{x}^1) - (1-\lambda) I^l_{\mathsf{S},\Lambda}(\mathbf{x}^2) \tag{5.16}$$

In our experiments, we compute the probability $P_{ccv}(\mathbf{H}^l_\Lambda)$ that a random pair of $\mathbf{x}^1, \mathbf{x}^2$ satisfies $D(\mathbf{x}^1, \mathbf{x}^2, \lambda) \geq 0$ for *all* values of $\lambda \in [0, 1]$ picked with sufficient granularity (10^{-3} to be precise). The individual channel coefficients h^l_{ij} in each instance were picked i.i.d as follows: $10\log_{10}(|h^l_{ij}|^2) \sim \mathcal{U}[0, 35]$ and $\angle h^l_{ij} \sim \mathcal{U}[0, 2\pi]$. Λ was fixed such that all nodes in layer l were in Λ and all nodes in layer $l+1$ were in $\bar{\Lambda}$ (but the results do not depend on the choice of Λ). The empirically observed value of this probability, averaged over several random channel-set instantiations, for different values of n and $K_l = K_{l+1} = n/2$ are as follows:

n	2	4	≥ 6
$\mathbb{E}_{\mathbf{H}^l_\Lambda}\left[P_{ccv}(\mathbf{H}^l_\Lambda)\right]$	0.9876	0.9997	≈ 1

This demonstrates that $I^l_{\mathsf{S},\Lambda}(x)$ is *almost concave*, implying that the relaxed version of (5.7) also has similar properties.

5.5 Numerical Evaluations

Algorithms

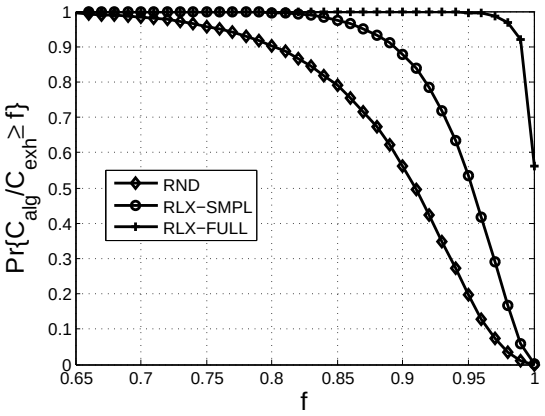
We evaluate three algorithms for subnetwork selection.

1. `RLX-FULL`: the main algorithm corresponding to the relaxed problem (5.10) for diamond networks and the relaxed version of (5.7) for multilayer networks.
2. `RLX-SMPL`: specific to diamond networks, corresponding to the relaxed problem (5.12).
3. `RND`: baseline algorithm where a subnetwork of appropriate size is selected at random.

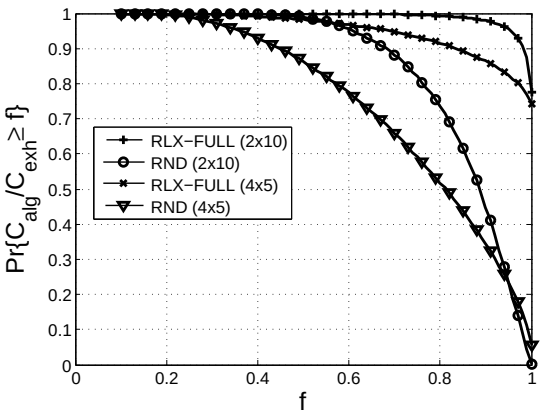
For both `RLX-FULL` and `RLX-SMPL`, the fractional optimum is rounded by picking the top K_i values in each layer and accordingly selecting the subnetwork (as described in Section 5.3.2).

Implementation

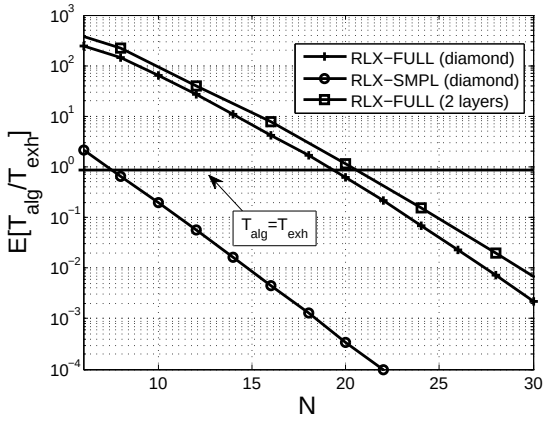
The implementations (done in `C++`) require two main modules: (i) A submodular minimization routine that evaluates $\bar{C}_{\mathsf{S},\mathsf{iid}}(\{\tilde{\mathbf{S}}_l\})$ for a specific set $\{\tilde{\mathbf{S}}_l\}$. For this, we used the `C` implementation of an algorithm based on the minimum-norm base [46], shown to be the most efficient general purpose routine in this regard. (ii) A routine that solves (5.10) and the relaxed version of (5.7). In view of Theorem 5.3.1, suitable interior-point based methods can solve (5.10) in polynomial time. However, off-the-shelf open source



(a) Accuracy–Diamond Network



(b) Accuracy–Multilayer Networks



(c) Timing Ratio

Figure 5.2 – Accuracy and Timing Performance of Algorithms

libraries to this end gave less than satisfactory results in practice. Instead, a Nelder-Mead simplex-based general purpose non-linear optimization routine from the `NLopt` library is used [47]. Specifically, the `NLOPT_LN_NELDERMEAD` function, combined with the augmented Lagrangian method in `NLOPT_AUGLAG_EQ` (to encode the size constraints) was used.

In all our experiments, for a given $(n, K_1, \ldots, K_{L-2})$ size tuple, we ran each algorithm for several (greater than 10^5) random channel-set instantiations of the network. The individual channel coefficients h_{ij}^l in each instance were picked i.i.d as follows: $10\log_{10}(|h_{ij}^l|^2) \sim \mathcal{U}[0, 35]$ and $\angle h_{ij}^l \sim \mathcal{U}[0, 2\pi]$.

5.5.1 Accuracy Results

For each random channel-set instantiation, we compute the ratio C_{alg}/C_{exh}, where C_{alg} (resp. C_{exh}) denotes $\bar{C}_{\mathsf{S,iid}}(\{\tilde{\mathbf{S}}_l\})$ of the optimal subnetwork selected by our algorithms

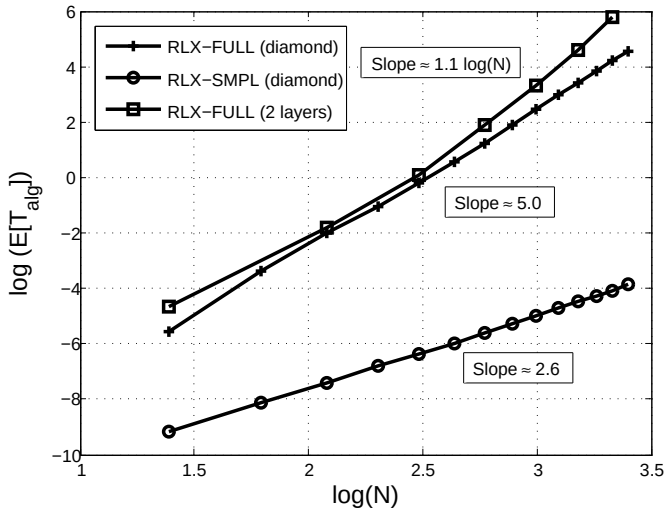


Figure 5.3 – $\log(\mathbb{E}[T_{alg}])$ vs $\log(N)$. A slope of $\delta \Rightarrow$ running time of $O(n^{\delta})$

(resp. by exhaustive search). Naturally, with this metric, the higher the C_{alg}/C_{exh} ratio, the better is the accuracy of the algorithm.

Fig. 5.2a plots the complementary c.d.f. of C_{alg}/C_{exh} for the three algorithms over diamond networks. The number of relays is $n = 20$ and the subnetwork size is $K_1 = 10$. Clearly, `RLX-FULL` produces subnetworks that have throughput equal or very close to the exhaustive optimal most of the time and both `RLX-FULL` and `RLX-SMPL` significantly outperform `RND`. Some representative values from Fig. 5.2a are as follows:

	`RLX-FULL`	`RLX-SMPL`	`RND`
$Pr\{C_{alg}/C_{exh} \geq 0.98\}$	0.9697	0.1679	0.0345
$Pr\{C_{alg}/C_{exh} \geq 0.94\}$	0.9983	0.6349	0.2724
$Pr\{C_{alg}/C_{exh} \geq 0.90\}$	0.9995	0.8782	0.5617

For multilayer networks (performance shown in Fig. 5.2b), experiments were performed for two configurations: In the first, marked as 2×10 in Fig. 5.2b, there are two intermediate layers of $n = 10$ nodes each and $K_1 = K_2 = 5$. In the second configuration, marked as 4×5 in Fig. 5.2b, there are 4 intermediate layers of 5 nodes each with a staggered size constraint of $K_1 = 2$, $K_2 = 4$, $K_3 = 2$, $K_4 = 3$. In both configurations, we see that the complementary c.d.f of `RLX-FULL` consistently outperforms that of `RND` and the benefits increase significantly at higher accuracies (i.e., higher C_{alg}/C_{exh} ratios).

5.5.2 Time Complexity

To measure the time efficiency of our algorithms, we construct the following configurations: (i) A diamond network with N relays, from which we select $N/2$ relays, and (ii) A layered network with having 2 intermediate layers of $N/2$ relays each, and we select $N/4$ relays from each layer.

For each N, we perform a large number of experiments with random channel-set instantiations and plot the average value of T_{alg}/T_{exh} in Fig. 5.2c, where T_{alg} (resp. T_{exh}) denotes the running time of our algorithms (resp. exhaustive search).

For the diamond network with $N = 30$, `RLX-FULL` is more efficient than an exhaustive search by a factor of $\mathbf{460}$, while for `RLX-SMPL`, this factor is more than $\mathbf{2.1 \times 10^6}$. This is primarily due to the much faster $O(N \log N)$ computation time of $\bar{C}^{\mathsf{dia}}_{\mathsf{P2P}}$ (in `RLX-SMPL`) w.r.t that of the submodular minimization routine for $\bar{C}^{\mathsf{dia}}_{\mathsf{S,iid}}$. For the multilayer confuguration, `RLX-FULL` gives time saving factors of more than $\mathbf{50}$ and $\mathbf{440}$ for $N = 28$ and 32 respectively.

While it is difficult to theoretically analyze the time complexity of a Nelder-Mead simplex-based algorithm for our problem, in Fig. 5.3 we give an empirical demonstration of time complexity for our implementations, where we plot $\log(T_{alg})$ (averaged over random channel-set instances) vs $\log(N)$ for the two configurations above.

For the diamond network, a fairly linear behavior is obtained, with slopes of approximately $\delta = 5.0$ and $\delta = 2.6$ for `RLX-FULL` and `RLX-SMPL`, implying that their running time is approximately $O(N^{5.0})$ and $O(N^{2.6})$ respectively. For the two-layered configuration, the slope is not constant, but a slowly growing function of N (about $1.1 \log N$ to a first approximation). Nevertheless, this is still the first systematic sub-exponential complexity ($\approx O(N^{1.1 \log N})$) algorithm for (approximately) solving the original integer optimization problem for multilayer networks, providing significant time savings w.r.t an exhaustive search (Fig. 5.2c). Also, with customized solvers (as opposed to the general purpose routines used here), further complexity gains are expected.

6 QUILT: A QMF Approach to Physical Layer Cooperation

Physical layer cooperation of a source with a single relay can significantly boost the performance of a wireless system, as shown in the theoretical work of Kramer [48] and verified by first experimental results by Duarte et. al. [49]. A natural question is, which scheme performs better when deployed in a practical system. The work of Duarte et. al. [49] on physical layer cooperation in WiFi suggests that Decode-Forward (DF) and Quantize-Map-Forward (QMF) are two good candidates. The performance of QMF and DF is competitive, yet which scheme performs best varies depending on the relative strengths of the channels that connect the source, relay and destination.

In the final chapter of this thesis, we propose and evaluate QUILT – a system for physical-layer relaying that seamlessly adapts to the underlying network configuration to achieve competitive or better performance as compared to the best current approaches. In QUILT, the relay operates on demand, i.e., is activated only if a first sequence transmitted by the source fails to be decoded by the destination. Once activated, it supports a second transmission of the source through physical layer cooperation. The relay decides opportunistically whether to use DF or QMF to recover the source sequence, on a frame-by-frame granularity and with no coordination from the source.

There are three main components to the system – source encoding, relay operation and destination decoding. QUILT uses LDPC codes specified in WiFi standards [5] to encode a sequence of information bits. The relay first attempts to decode its received signal. If decoding fails, the relay quantizes it to the closest discrete sequence and recovers a noisy version of the source sequence. In both cases, the relay interleaves it and transmits it synchronously with the source.

The decoder at the destination tries to decode the first direct transmission from the source. Only when this is unsuccessful, in which case the source and relay transmit cooperatively, the decoder makes a second attempt. In this attempt, it combines information that it has received in both the transmissions to decode the sequence sent by the source. The decoder employs a belief propagation decoding algorithm over a graphical model of the

relay network that incorporates the source LDPC code, quantization and interleaving at the relay and joint decoding at the destination.

We deploy QUILT on a WARPLAB testbed and present exhaustive performance comparisons with DF and QMF protocols through over-the-air experiments. Our experimental results demonstrate benefits up to a factor of 5 for Frame Error Rate (FER) as compared to the next best scheme and two orders of magnitude over the FER of traditional point-to-point transmissions.

6.1 Related Work

A first implementation of QIF was presented in [49] and offered comparisons with DF and AF relaying schemes. Although our work builds on [49], QUILT differs in a number of important features, that include: (a) the opportunistic use of decoding or quantizing at the relay; (b) the use of interleaving even when decoding was successful; and (c) the use of hybrid decoding. Completely new to this work is also the theoretical analysis that illustrates through outage calculations the benefits of interleaving and hybrid decoding, as well as all the experimental evaluations and comparisons. In summary, all claimed novel contributions are unique to this chapter.

The works in [50], [51] survey testbed implementations of physical layer relay schemes; the focus is on the implementation of either DF or AF schemes. A testbed based on uncoded DF in a single-relay system was investigated in [52]. A WARP radio testbed based on DF was implemented in [53]. None of these works implemented advanced error correction or broadband OFDM modulation. In [54] both (uncoded) AF and DF relaying along with distributed Alamouti-based transmission were implemented over broadband OFDM. However, this implementation lacked error correcting codes and distributed frequency-diversity coding. Apart from the relaying strategy, other issues related cooperative relaying have also been studied through implementation on testbeds; for example the experimental work in [55] and [56] focuses on the synchronization for multiple simultaneous transmissions.

The monograph in [48] surveys the recent theoretical development in cooperative relaying. QMF was originally proposed in [8] for Gaussian networks and shown to approximately achieve the network capacity. It was later extended to discrete memoryless networks in [44]. Practical coding schemes of QMF relaying with LDPC and BICM were proposed for a half-duplex single-relay cooperative MIMO system in [57] and for a full-duplex multi-relay network in [58], where in the latter interleavers as relay mapping were first proposed. The use of demodulation instead of quantization was used in [58] and [59]. None of these papers had an experimental evaluation.

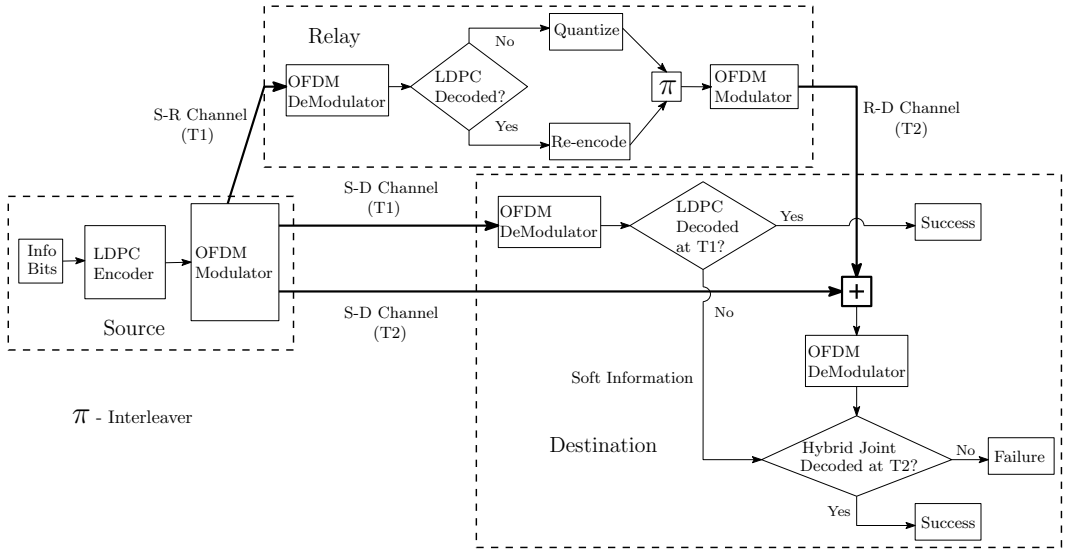


Figure 6.1 – Schematic diagram of QUILT illustrating the various components of the system. T_1 and T_2 indicate the first and second phase, respectively.

6.2 QUILT System Overview

QUILT prescribes physical layer operations for a three-node network that consists of a source, a relay and a destination, building on top of the physical layer procedures of WiFi IEEE802.11. The relay is half-duplex, i.e., it can either transmit or receive. We describe its main components in the following and also depict it schematically in Fig. 6.1.

6.2.1 Source Operation

Channel Coding

The source encodes each information packet using an LDPC code (compliant with the IEEE802.11 specifications) to create a coded packet; all transmitted packets by the source are coded. We use coding as recommended in the WiFi standards to increase the end-to-end reliability.

Broadband OFDM Modulation

We employ OFDM modulation as specified in the WiFi physical layer to combat channel frequency selectivity. After encoding, the codeword bits are first mapped to QAM symbols and then modulated using OFDM. Each coded packet the source creates results in several OFDM symbols.

6.2.2 On Demand Relaying: Two Phase Operation

Gist The source first attempts to directly transmit a packet to the destination. The relay also overhears this transmission. If the direct transmission is successful, the source proceeds with the transmission of a new packet; if unsuccessful, the source and the relay cooperatively transmit to try to help the destination decode. We thus have a two-phase operation, where the relay aids the information transfer as-needed, only when the direct transmission from the source is unsuccessful. This enables the system to adapt to the network conditions and avoid unnecessary relay transmissions when the source-destination channel is strong.

Signal Exchange

We use vectors $\mathbf{X} = [X_1, X_2, \ldots, X_m]^T$, to collect the QAM symbols transmitted across the m subcarriers of one OFDM symbol. We denote with $\mathbf{X}_s[k]$ and $\mathbf{X}_r[k]$ the transmitted signal vectors by the source and the relay, $\mathbf{Y}_r[k]$, $\mathbf{Y}_d[k]$, $\mathbf{Z}_r[k]$ and $\mathbf{Z}_d[k]$ the received vectors and the Gaussian noise at the relay and destination, and $\mathbf{H}_{ij}[k] = \text{diag}(H_{ij,1}[k], \ldots, H_{ij,m}[k])$ the channel matrix from node i to node j; $k = 1$ or 2 indicates the phase.

- Phase 1: The relay is in listening mode. The received signals per OFDM symbol are:

$$\begin{aligned} \mathbf{Y}_r[1] &= \mathbf{H}_{sr}[1]\mathbf{X}_s[1] + \mathbf{Z}_r[1] \\ \mathbf{Y}_d[1] &= \mathbf{H}_{sd}[1]\mathbf{X}_s[1] + \mathbf{Z}_d[1]. \end{aligned} \tag{6.1}$$

  If the destination cannot decode, we enter Phase 2.

- Phase 2: The source transmits an identical packet, i.e., $\mathbf{X}_s[2] = \mathbf{X}_s[1]$ for all symbols in the packet. The relay transmits $\mathbf{X}_r[2]$'s created from $\mathbf{Y}_r[1]$'s in the packet:

$$\mathbf{Y}_d[2] = \mathbf{H}_{sd}[2]\mathbf{X}_s[2] + \mathbf{H}_{rd}[2]\mathbf{X}_r[2] + \mathbf{Z}_d[2]. \tag{6.2}$$

  How the relay creates $\mathbf{X}_r[2]$'s is dealt in the next section.

6.2.3 Relay Operation in Phase 2

At a high level The relay attempts to decode and exactly recover the sequence of OFDM symbols $\mathbf{X}_s[1]$'s transmitted by the source in a packet; if it fails, it uses symbol quantization of the elements of the received symbols $\mathbf{Y}_r[1]$'s to their closest constellation points; in both cases, it interleaves the recovered sequences and transmits it synchronously

with the source, effectively creating a distributed space-frequency code.

In more detail The relay operates as follows.

- Attempts to recover the source information in a packet, using an LDPC decoder and soft information from its received vectors $\mathbf{Y}_r[1]$'s. It infers success through the CRC check.
  - If successful, it re-encodes the source information to create the same vectors $\mathbf{X}_s[1]$'s as the source.
  - If unsuccessful, it quantizes the elements of its received vectors $\mathbf{Y}_r[1]$'s to their closest constellation points, and creates a (noisy, with discrete errors) version of the $\mathbf{X}_s[1]$ vectors the source has.
- Maps the elements of the recovered vectors to bits, interleaves the resulting bit sequence with a randomly selected bit-interleaver, maps the interleaved bit sequence to signal constellation points, passes it through an OFDM modulator, and transmits it synchronously with the source.

Discussion We here discuss the reasons for selecting our particular method for sequence recovery, and for interleaving at the relay.

To recover the source sequence, if the relay can successfully decode, this is the optimal operation it can do, as it perfectly cleans up the noise. If the relay fails to decode, our symbol quantization attempts to recover a sequence that is close to the source transmission and conveys information to the destination. To achieve this, symbol quantization is not the only option: infact, the insight from the information theoretic form of QMF is that we should be using sequence quantization. For instance, a possible choice could be to select the codeword an ML decoder would identify, even if this is not the correct one; that is, use the closest codeword to the receiver signal, which amounts to quantizing to the codeword sequences. We were not able to experiment with this option, as it leads to impractical complexity both at the relay and the destination. We opted for symbol quantization that still identifies a sequence close to the transmitted one, yet has viable complexity.

Interleaving is a key component of our relay operation for two independent reasons. The first is specific to OFDM modulation: because of interleaving, the relay assigns signals received through weak or interfered subcarriers in the source-relay channel to potentially strong or cleaner subcarriers in the relay-destination channel and also induces mixing of signals from distributed terminals across subcarriers, thus achieving frequency-space diversity and significant performance benefits (see Section 6.6). This benefit is present irrespective of quantization or decoding at the relay. The second reason is specific to

QMF: as our theoretical analysis in Section 6.3 shows, the mapping that interleaving implements, outperforms random mappings for the QMF operation, offering significant benefits (see Section 6.3) even when we operate on a single subcarrier; i.e., these benefits are independent of OFDM.

6.2.4 Hybrid Decoding at the Destination

In phase 1, the destination attempts to decode using a standard LDPC decoder. If it fails, at the end of phase 2, QUILT takes advantage of the received signals in both phases to decode the source packet. For this, the destination employs a graphical structure that captures the streams received in phases 1 and 2, and adapts to whether decoding or quantization were employed at the relay. The decoder for QUILT is an adaptation of the QIF decoder in [49][58], wherein the stochastic quantizer nodes become deterministic perfect connections if the relay decoding succeeds, and are the same as in [49] otherwise. The decision is guided by a 1-bit flag that the relay transmits, to inform the destination whether the relay-decoding succeeded. Further, the log-likelihood ratio computations take into account the received soft information from both transmission phases.

6.3 Theoretical Analysis

We here provide theoretical analysis that substantiates our design choices in QUILT. We show that we gain:

- Benefits from interleaving over the conventional random mapping operation in QMF.
- Benefits from hybrid decoding at the destination.
- Benefits from opportunistic relay decoding/quantization.

For our performance evaluations, we compared information theoretical metrics, such as outage probability, through simulations over narrowband (single-carrier) flat Rayleigh-fading channels that assume infinite complexity processing at the source, the relay and the destination.

6.3.1 Performance Metric: Outage Probability

We evaluate the error performance using the classical notion of outage probability [60], i.e., the probability that a (fixed) transmission rate R is not supported by a scheme. For our calculations we assume 4-QAM constellations at the source and relay. We also assume that the channels are fading i.i.d. over the two phases (a situation commonly

encountered when the two phases occur sufficiently far apart, larger than the coherence time of the channel), independently across the three links, but the distributions in the three links may not be identical. The target rate of the transmit packet is $R = 1$ bit/s/Hz. Adapting for our two-phase on-demand relaying protocol, we have that

$$\begin{aligned} &\mathbb{P}[\text{Outage}] \\ &= \mathbb{P}\Big[\Big\{R > \mathsf{C}_{\text{P2P}}\left(h_{sd}[1]\right)\Big\} \cap \Big\{R > \mathsf{C}_{\text{R}}\left(h_{sd}[1], h_{sr}[1], h_{sd}[2], h_{rd}[2]\right)\Big\}\Big] \\ &= \mathbb{P}\Big[R > \mathsf{C}_{\text{P2P}}\left(h_{sd}[1]\right)\Big]\mathbb{P}\Big[R > \mathsf{C}_{\text{R}}\left(\cdot\right) | R > \mathsf{C}_{\text{P2P}}\left(h_{sd}[1]\right)\Big] \end{aligned} \tag{6.3}$$

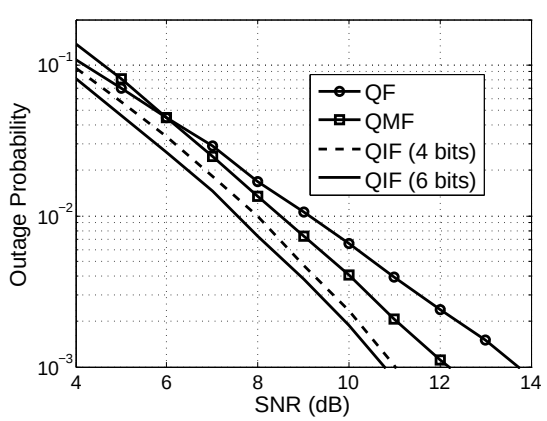
where $\mathsf{C}_{\text{P2P}}(h_{sd}[1])$ is the single-user capacity supported by channel $h_{sd}[1]$ and QAM constellation, and $\mathsf{C}_{\text{R}}\left(h_{sd}[1], h_{sr}[1], h_{sd}[2], h_{rd}[2]\right)$ is the capacity of the cooperative scheme, which depends on the particular strategy under consideration. For strategies that do not use hybrid decoding, C_{R} is just a function of $h_{sr}[1], h_{sd}[2]$ and $h_{rd}[2]$. We evaluate numerically the outage probability by using analytical expressions for $\mathsf{C}_{\text{R}}\left(h_{sr}[1], h_{sd}[2], h_{rd}[2]\right)$ for each strategy, that we derived by modifying the arguments in [8, 44]. The detailed calculations can be found in the Appendix A.2.

6.3.2 Benefits of Interleaving

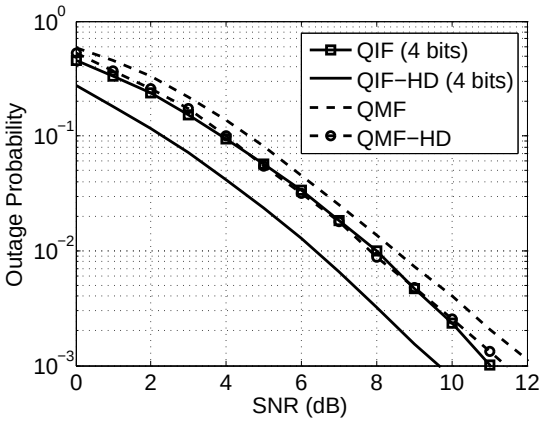
We compare the following schemes: (i) QMF: scalar quantization followed by random mapping at the relay, as in [8], (ii) QIF: scalar quantization followed by bit-level interleaving at the relay and (iii) QF (Quantize-Forward): only scalar quantization at the relay.

The plot in Fig. 6.2a is generated with all three links having i.i.d. Rayleigh fading channels with the same SNR. We observe that QIF outperforms QMF, even for very short interleaver lengths[1]. This can be intuitively explained as follows: in the original QMF relaying scheme, the random mapping at relay results in independence between the transmissions of the source and the relay. Hence the original QMF cannot harness the coherent combining power gain that may increase the performance in the moderate SNR regime. Instead, in QIF the interleaver preserves the weight of the quantized codeword and hence retains certain correlation with the transmission from the source, while providing enough mixing across source and relay terminals to guarantee spatial diversity. Indeed, we observe that QIF outperforms QF significantly, since with no mapping, QF cannot extract the full spatial diversity.

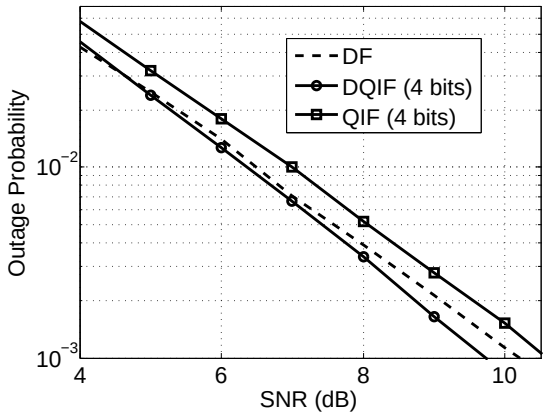
[1]Due to a multi-letter vector channel representation, it is only feasible to numerically evaluate the expressions for short length interleavers. However, we do see that performance improves with length of the interleaver. Thus the theoretical plots for QIF in this section are much more pessimistic than the long-length interleavers that we use in our over-the-air experiments.



(a) Outage performance for QF, QMF, and QIF. All channels are i.i.d.



(b) Outage performance for QMF and QIF with and without hybrid decoding. All channels are i.i.d.



(c) Outage performance for QIF, DF, and DQIF. $\mathsf{SNR}_{RD} = 4\mathsf{SNR}_{SD} = 4\mathsf{SNR}_{SR}$. X-axis denotes SNR_{SD}.

Figure 6.2 – Outage performance of different relaying schemes.

6.3.3 Benefits of Hybrid Decoding

In Fig. 6.2b (where again all three links have i.i.d. Rayleigh fading channels) we verify that hybrid decoding leads to a significantly improved performance for QIF and QMF. The versions of QMF and QIF with hybrid decoding are labeled QMF-HD and QIF-HD respectively. The gain observed is well expected as the signal received in Phase 1 contains information that can improve the decoding performance. Interestingly, the gain for hybrid decoding in QIF, roughly 1.5dB, is almost double of that in QMF.

6.3.4 Benefits of Opportunistic Decoding or Quantization

We compare the following schemes: (i) DF: relay decodes and forwards if it can, else does not cooperate (ii) QIF: as mentioned above, and (iii) DQIF (Decode/Quantize-Interleave-Forward): the relay opportunistically decodes and forwards if possible, else performs QIF.

In Fig. 6.2c, where all three links have Rayleigh fading channels, but the SNR in the relay-destination link is four times stronger than that in the source-destination and source-relay links, we observe the benefit of opportunistic decoding when the reception at the relay is weak. In particular, while DF slightly outperforms QIF, DQIF is also shown to extract the combined benefits of both DF and QIF. Moreover, we must point out that the theoretical demonstrations for QIF are carried out with short length interleavers and the performance of QIF improves with interleaver length (see Fig. 6.2a). In real-world experiments, we use long interleavers that will provide better performance than the demonstrations in this section show. The relative superiority of DF and QIF will of course vary with channel conditions, but a combination of the two appears to be a promising scheme in terms of universality.

6.4 System Implementation

6.4.1 Cooperative Schemes Implemented

Below we give a description and motivation of the schemes we analyze via experiments using our deployed testbed. The cooperative schemes implemented are summarized in Table 6.1. The relay operations we consider in our experiments are:

	Quantize-Forward	Quantize-Interleave-Forward (QIF)	Decode-Forward (DF)	Decode-Interleave-Forward (DIF)	Decode-Interleave-Quantize-Interleave-Forward (DIQIF)
No Hybrid Decoding	QF	QIF	DF	DIF	DIQIF
With Hybrid Decoding (HD)	QF-HD	QIF-HD	DF-HD	DIF-HD	**QUILT= DIQIF-HD**

Table 6.1 – Implemented schemes when the relay is active. The relay operations (columns) and the destination operations in Phase 2 (rows), are described in Section 6.4.1.

- *Quantize-Forward (QF):* Scalar quantization and subsequent forwarding by the relay.

- *Quantize-Interleave-Forward (QIF):* Scalar quantization followed by bit-level interleaving of the quantized sequences by the relay and subsequent forwarding.

- *Decode-Forward (DF):* Decoding at the relay if possible and transmit a 2×1 Alamouti jointly with the source. If decoding at the relay is not possible it remains silent.

- *Decode-Interleave-Forward (DIF):* Decoding at the relay if possible and transmit bit-level interleaved signal. If decoding at the relay is not possible it remains silent.

- *Decode-Interleave-Quantize-Interleave-Forward (DIQIF):* DIF if relay decoding succeeds; QIF otherwise.

We note that DIF was not considered in our single-carrier theoretical analysis. We implemented this for our (OFDM-based) over-the-air experiments to provide DF an option to exploit the frequency diversity across subcarriers that the interleaver in QIF was inherently providing.

For Phase 2, the destination operations we consider are:

- *No Hybrid Decoding:* The decoding at destination only uses the signal received in Phase 2.

- *On Demand Hybrid Decoding (HD):* The destination first attempts to decode with only the signal received in Phase 2. If this decoding fails, then the destination attempts to decode again but this second time with both the signals received in Phase 1 and Phase 2.

To further demonstrate the utility of cooperation, we implement the following baseline scheme:

- *Direct Transmission (DT):* In this baseline scheme (without the need of a relay) in Phase 2, the source repeats the Phase 1 signal. We also consider DT with the possibility of hybrid decoding, termed DT-HD.

Also, note that, in the nomenclature used in Table 6.1, QUILT refers to DIQIF-HD, which is essentially the all-encompassing system that is the cornerstone of this chapter.

6.4.2 Frame Structure

We designed our system to emulate the physical layer procedures of WiFi (IEEE802.11). Each transmitted frame consists of a preamble and the payload. We next describe the preamble and payload fields for the two phases of communication for the schemes we implemented.

Preamble

The preamble structure follows what is used in 802.11 systems: it consists of training sequences for Automatic Gain Control (TAGC), training for timing synchronization (TSYNC), and training for channel estimation (TCHE). The training for channel estimation is used to estimate not only the channel but also to estimate the carrier frequency offset.

Phase 1: In this phase, only the source transmits. The preamble structure it transmits is shown in Fig. 6.3.

Phase 2: In DT, the source transmits the same preamble it transmitted in Phase 1 and the relay remains silent. For all other schemes (QF, QIF, DF, DIF, DIQIF) we deal with joint transmissions from the source and the relay as follows. The TAGC is sent by the source and relay simultaneously. However, we introduce a cyclic shift between the TAGC waveforms sent by the source and the relay to avoid accidental nulling. We send the TSYNC as well as all TCHE fields orthogonally over time, as shown in Fig. 6.3. Orthogonality for TSYNC ensures that the destination can solve timing synchronization from at least one of two TSYNC sequences, and thus, even if one of the channels happens to be very noisy, it can still synchronize. Orthogonality for TCHE ensures clean channel estimates for separate links (a similar approach is used for MIMO channel training implementations).

Payload

The payload consists of OFDM symbols, i.e., contains data and pilot subcarriers as described in 802.11.

Phase 1: For all the schemes, we transmit the exact same payload waveform, which corresponds to an OFDM-based single transmitter single receiver antenna system.

Phase 2: In DT, the source transmits the same payload it transmitted in Phase 1 and the relay remains silent. In QF, QIF, DIF and DIQIF, the source retransmits the same payload it transmitted in Phase 1, and the relay transmits its received and processed signal. In the DF, if the relay has successfully decoded (the CRC passed), we have the source and relay payload implement a 2×1-antenna distributed Alamouti code to provide

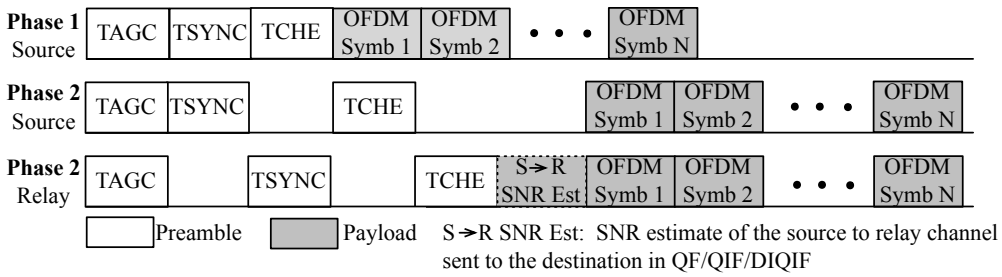


Figure 6.3 – Time diagram.

spatial diversity as in [49]. In DF and DIF, if the relay cannot decode then it remains silent.

In QF, QIF and DIQIF schemes, the payload contains one more OFDM symbol than the DF and DIF schemes which is only sent by the relay. This extra OFDM symbol is used to forward an estimate of the source–relay SNR to the destination, which needs to employ it during iterative decoding. The relay first estimates the SNR and quantizes it to one of 40 possible values ranging from -10 to 30 dB (in steps of 1 dB). We can describe these 40 values using 6 bits. For QF and QIF we repeat these 6 bits 8 times, modulate them with BPSK and allocate them to 48 data subcarriers in the OFDM symbol used to forward the SNR information. For DIQIF, in addition to the 6 bits of SNR information, we also send one extra bit to the destination to notify if the relay decoded successfully or not. The 7 bits for DIQIF are sent in the 48 data subcarriers of the extra OFDM symbol.

We note that, for decoding the payload, an estimation of the effective noise variance is required by the LDPC decoder for computation of the log-likelihood ratios. For the estimation of SNR and effective noise variance, we follow the same approach as presented in [49].

As per the 802.11 standard, each OFDM symbol in the payload consists of a total of 48 data subcarriers, 4 pilot subcarriers and 12 unused subcarriers. The 4 pilot subcarriers are used for residual phase noise and CFO correction. For the joint source and relay transmissions, we synchronize the carrier and timing between the source and relay by sharing a wire connection between them as shown in Fig. 6.5–the same approach as presented in [49]. Yet, we would like to mention that recent work on distributed transmissions has shown that it possible to also achieve accurate timing and carrier synchronization in a distributed manner (see [61, 55, 62]); these protocols are enabled by implementing a large part of the mechanisms in real time in the FPGA to achieve fast turnaround times. Incorporating this into WARPLab, although feasible, was not our focus.

6.5 Iterative Decoder Design

Let $\mathcal{A}$ be the discrete channel-alphabet at the source and the relay. In our implementations, we use a standard 2^k-QAM (specifically 16-QAM) alphabet as $\mathcal{A}$. Let N denote the transmission blocklength. It will be beneficial to rewrite the network model equations in (6.1) and (6.2) in terms of the vectors in $\mathcal{A}^N$. For Phase 1, we have

$$\begin{aligned} \mathbf{y}_r &= \mathbf{h}_{sr}\mathbf{x}_s + \mathbf{z}_r \\ \mathbf{y}_d^1 &= \mathbf{h}_{sd}^1\mathbf{x}_s + \mathbf{z}_d^1. \end{aligned} \tag{6.4}$$

If the destination cannot decode, we enter Phase 2.

$$\mathbf{y}_d^2 = \mathbf{h}_{sd}^2\mathbf{x}_s + \mathbf{h}_{rd}\mathbf{x}_r + \mathbf{z}_d^2. \tag{6.5}$$

Here $\mathbf{x}_s, \mathbf{x}_r \in \mathcal{A}^N$ are the symbol vectors transmitted by the source and the relay and $\mathbf{y}_r$, $\mathbf{y}_d^k$, $\mathbf{z}_r$ and $\mathbf{z}_d^k$ (all $\in \mathbb{C}^N$) are the received vectors and the Gaussian noise at the relay and destination in Phase $k \in \{1, 2\}$. $\mathbf{h}_{ij}^k = \text{diag}(h_{ij,1}^k, \ldots, h_{ij,N}^k)$ is the channel matrix from node i to node j in Phase $k \in \{1, 2\}$.

6.5.1 Encoding and Relaying

The source encodes the information bit vector $\mathbf{u}$ into the symbol vector $\mathbf{x}_s$. If decoding at the destination from $\mathbf{y}_d^1$ is unsuccessful, then the relay tries to decode $\mathbf{y}_r$. If the decoding is unsuccessful, it quantizes $\mathbf{y}_r$ symbol wise to the nearest symbol in $\mathcal{A}$. In either case, we denote the result by $\mathbf{x}_q$. This is then interleaved using an interleaver π to produce the final transmit vector $\mathbf{x}_r$ at the relay. In summary,

$$\mathbf{y}_r \to \mathbf{x}_q \xrightarrow{\pi} \mathbf{x}_r \tag{6.6}$$

Since interleaving is a one-to-one operation, we can write

$$\mathbf{x}_r = \pi(\mathbf{x}_q) \text{ and } \mathbf{x}_q = \pi^{-1}(\mathbf{x}_r) \tag{6.7}$$

where π^{-1} is the inverse permutation of π.

To describe the decoder structure in detail, we initially focus on the binary communication problem, i.e $\mathcal{A} = \{\pm 1\}$. We defer the extension to non-binary constellations to section 6.5.3. For our source codebook $\mathcal{C}$, we use an LDPC code of the desired rate with a code-membership function $\mathbb{1}_{\{\mathbf{x}_s \in \mathcal{C}\}}$.

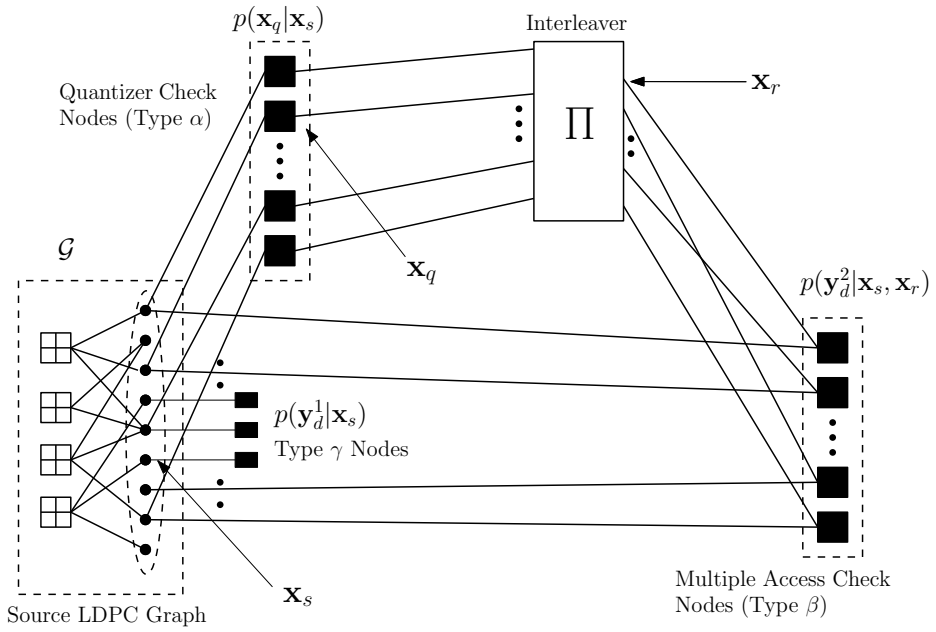


Figure 6.4 – Decoding Graph for binary signaling with 1-bit scalar quantizers

6.5.2 Iterative DIQIF Decoder

We derive a compound graphical model to perform iterative decoding of the source bits $\mathbf{x}_s$. In contrast to point-to-point decoding, the network graphical model for our iterative QMF decoder has to incorporate the following additional features: (i) It should explicitly characterize of the function-nodes representing the quantization and multiple-access operations, together with their own set of message passing rules, and (ii) It should have a well defined information-exchange schedule among the components of the graphical model. In the following, we detail the derivation and key features of the graphical model.

Sum-Product decomposition of *a posteriori* probability

The decoding rule for the bit-wise MAP decoder for the i-th bit $x_{s,i}$ of the source codeword reads:

$$\hat{x}_{s,i}^{MAP}(\mathbf{y}_d^1, \mathbf{y}_d^2) = \text{argmax}_{x_{s,i}\in\{\pm 1\}} \sum_{\sim x_{s,i}} p(\mathbf{x}_s|\mathbf{y}_d^1, \mathbf{y}_d^2) \tag{6.8}$$

Note that, for hybrid decoding, we use the signals received in both the phases. Here we use the standard notation $\sim$ in front of $x_{s,i}$ to denote all variables other than $x_{s,i}$. The

a-posteriori probability, $p(\mathbf{x}_s|\mathbf{y}_d^1, \mathbf{y}_d^2)$, can be expressed as,

$$p(\mathbf{x}_s|\mathbf{y}_d^1, \mathbf{y}_d^2) = \frac{1}{p(\mathbf{y}_d^1, \mathbf{y}_d^2)} \sum_{\sim \mathbf{x}_s, \mathbf{y}_d^1, \mathbf{y}_d^2} p(\mathbf{x}_s, \mathbf{x}_q, \mathbf{x}_r, \mathbf{y}_d^1, \mathbf{y}_d^2) \tag{6.9}$$

$$p(\mathbf{x}_s|\mathbf{y}_d^1, \mathbf{y}_d^2) \overset{(a)}{\propto} \sum_{\sim \mathbf{x}_s, \mathbf{y}_d^1, \mathbf{y}_d^2} p(\mathbf{x}_s) \cdot p(\mathbf{x}_q|\mathbf{x}_s) \cdot p(\mathbf{x}_r|\mathbf{x}_q) \cdot p(\mathbf{y}_d^1|\mathbf{x}_s) \cdot p(\mathbf{y}_d^2|\mathbf{x}_s, \mathbf{x}_r) \tag{6.10}$$

$$\overset{(b)}{\propto} \sum_{\sim \mathbf{x}_s, \mathbf{y}_d^1, \mathbf{y}_d^2} p(\pi^{-1}(\mathbf{x}_r)|\mathbf{x}_s) \cdot p(\mathbf{y}_d^1|\mathbf{x}_s) \cdot p(\mathbf{y}_d^2|\mathbf{x}_s, \mathbf{x}_r) \cdot \mathbb{1}_{\{\mathbf{x}_s \in \mathcal{C}\}} \tag{6.11}$$

(a) follows from the fact that $\mathbf{x}_s \leftrightarrow \mathbf{x}_q \leftrightarrow \mathbf{x}_r \leftrightarrow \mathbf{y}_d^2$ and $\mathbf{x}_s \leftrightarrow \mathbf{y}_d^1$ form Markov chains and $\mathbf{y}_d^1$ and $\mathbf{y}_d^2$ are independent given $\mathbf{x}_s$. (b) follows from the uniform distribution on the source codeword, the one-to-one relationship between $\mathbf{x}_q$ and $\mathbf{x}_r$ and the code membership constraints. Also, from the memoryless property of the channel, the terms $p(\pi^{-1}(\mathbf{x}_r)|\mathbf{x}_s)$, $p(\mathbf{y}_d^1|\mathbf{x}_s)$ and $p(\mathbf{y}_d^2|\mathbf{x}_s, \mathbf{x}_r)$ further factorize on a symbol-by-symbol basis as shown in the decoder graph in Fig. 6.4. The decoding problem thus reduces to computing the marginal of a factorized function and choosing the value that maximizes the marginal.

Structure of the decoder

As shown in Fig. 6.4, the compound graph contains the graphs corresponding to the source codebook ($\mathcal{G}$) and the relay interleaver π. In addition to the variable and check nodes in $\mathcal{G}$, two other types of function nodes enter the graph structure:

- The source-to-relay (type α) function nodes, which connect the $\mathbf{x}_s$ and $\mathbf{x}_q$ variable nodes, and represent the functions $p(\mathbf{x}_q|\mathbf{x}_s)$ at the relay. These nodes correspond to the decode/quantize operation at the relays. Note that if the relay is successful in decoding, then $p(\mathbf{x}_q|\mathbf{x}_s) = 1$ and these nodes become redundant.
- The multiple-access (type β) function nodes, connecting the $\mathbf{x}_r$ and $\mathbf{x}_s$ variable nodes, and representing the function $p(\mathbf{y}_d^2|\mathbf{x}_s, \mathbf{x}_r)$. The information-exchange between the variables at the source and relay via these nodes is an important ingredient in harnessing the benefits of co-operation from the relays.

Also note that the variable nodes in $\mathcal{G}$ also receive soft information extracted from the transmission in Phase 1 from the nodes of type γ that represent the function $p(\mathbf{y}_d^1|\mathbf{x}_s)$.

Message passing rules

The decoding proceeds via a message-passing algorithm on the decoding graph. We set all messages flowing through the edges to be in *log-likelihood ratio* form, i.e of the

form $\ln \frac{p(x=+1)}{p(x=-1)}$. The messages passed from every variable node and also from the check nodes in $\mathcal{G}$ follow usual belief propagation (BP) message passing rules as summarized in [2]. However, we need to illustrate the message passing rules for the type α and type β function nodes that are new to the network graphical model.

Each *type* α function node c is connected to a variable node v_0 in $\mathcal{G}_0$ and to another node v_i in Π. The messages passed from the *type* α function node to node v_0 in $\mathcal{G}$ and to node v_i in Π are given by

$$\begin{aligned} m_{c\to v_0}^{\alpha\to\mathcal{G}} &= \ln \frac{p_{+1|+1}\cdot p(v_i=+1)+p_{-1|+1}\cdot p(v_i=-1)}{p_{+1|-1}\cdot p(v_i=+1)+p_{-1|-1}\cdot p(v_i=-1)} \\ &= \ln \frac{p_{+1|+1}e^{m_{v_i\to c}^{\Pi\to\alpha}}+p_{-1|+1}}{p_{+1|-1}e^{m_{v_i\to c}^{\Pi\to\alpha}}+p_{-1|-1}} \qquad (6.12) \\ m_{c\to v_i}^{\alpha\to\Pi} &= \ln \frac{p_{+1|+1}\cdot p(v_0=+1)+p_{+1|-1}\cdot p(v_0=-1)}{p_{-1|+1}\cdot p(v_0=+1)+p_{-1|-1}\cdot p(v_0=-1)} \\ &= \ln \frac{p_{+1|+1}e^{m_{v_0\to c}^{\mathcal{G}\to\alpha}}+p_{+1|-1}}{p_{-1|+1}e^{m_{v_0\to c}^{\mathcal{G}\to\alpha}}+p_{-1|-1}} \qquad (6.13) \end{aligned}$$

where $p_{\pm1|\pm1}$ denote the transition probabilities, $p(v_i|v_0)$, where $v_0 \in \mathbf{x}_s$ and $v_i \in \mathbf{x}_q$, and are obtained from the channel statistics.

Each *type* β function node c is connected to a variable node v_1 in $\mathcal{G}$ and to v_2 in Π. Using similar marginalizations of the corresponding functions as in the case of type α nodes, the messages passed from the type β nodes are derived as

$$m_{c\to v_1}^{\beta\to\mathcal{G}} = \ln \frac{p_{+1,+1}e^{m_{v_2\to c}^{\Pi\to\beta}}+p_{+1,-1}}{p_{-1,+1}e^{m_{v_2\to c}^{\Pi\to\beta}}+p_{-1,-1}} \qquad (6.14)$$

$$m_{c\to v_2}^{\beta\to\Pi} = \ln \frac{p_{+1,+1}e^{m_{v_1\to c}^{\mathcal{G}\to\beta}}+p_{-1,+1}}{p_{+1,-1}e^{m_{v_1\to c}^{\mathcal{G}\to\beta}}+p_{-1,-1}} \qquad (6.15)$$

where $p_{\pm1,\pm1}$ here represents the pdf (evaluated at the observation $\mathbf{y}_d^1$) of channel output, conditioned on $v_1 \in \mathbf{x}_s$ and $v_2 \in \mathbf{x}_r$ respectively. The messages $m_{v_2\to c}^{\Pi\to\beta}$ are derived from the messages $m_{c\to v_i}^{\alpha\to\Pi}$ by applying the permutation defined by π. Similarly, the messages $m_{v\to c}^{\Pi\to\alpha}$ are computed by applying π^{-1} to $m_{c\to v}^{\beta\to\Pi}$.

Decoding Schedule

Having defined the message-passing rules for the variable and function nodes, it remains to specify the schedule for information exchange in the compound graphical model. We will start from the type β nodes and push the llrs towards the source graph $\mathcal{G}$ through Π and the type α nodes. After performs local message-passing within $\mathcal{G}$, the llrs are pushed

back through the type α nodes and Π towards the type β nodes. This process continues until a hard decision is taken on the value of the source bits $\mathbf{x}_s$, based on the sign of the corresponding messages at the variable nodes in $\mathcal{G}$, which is obtained by adding the messages from all incident edges. Such a schedule makes good use of the parallelism inherent in the network structure and is illustrated in the algorithm below.

Algorithm 2 Decoding Schedule

Initialize: $m_{v\to c}^{\Pi\to\beta} = m_{v\to c}^{\mathcal{G}\to\beta} = 0$
$m_{c\to v}^{\alpha\to\mathcal{G}_1} = m_{c\to v}^{\alpha\to\mathcal{G}_2} = 0$
Initialize: $m_{c\to v}^{\beta\to\mathcal{G}_1}$, $m_{c\to v}^{\beta\to\mathcal{G}_2}$ from channel observations
for $l = 1$ to *max_iter* **do**
Compute $m_{c\to v}^{\beta\to\mathcal{G}}$ and $m_{c\to v}^{\beta\to\Pi}$ using (6.14) and (6.15).
Compute $m_{v\to c}^{\Pi\to\alpha}$ by applying π^{-1} to $m_{c\to v}^{\beta\to\Pi}$.
Compute $m_{c\to v}^{\alpha\to\mathcal{G}}$ using (6.12).
Run ω rounds of belief propagation on $\mathcal{G}$ with $m_{c\to v}^{\alpha\to\mathcal{G}}$, $m_{c\to v}^{\beta\to\mathcal{G}}$ and $m_{c\to v}^{\gamma\to\mathcal{G}}$ as input.
Update $m_{v\to c}^{\mathcal{G}\to\alpha}$ and $m_{v\to c}^{\mathcal{G}\to\beta}$
Compute $m_{c\to v}^{\alpha\to\Pi}$ using (6.13), from which compute $m_{v\to c}^{\Pi\to\beta}$ by applying π.
end for

After a fixed number of global iterations, a hard decision is taken on the value of the source bits $\mathbf{x}_s$, based on the sign of the corresponding messages at the variable nodes which is obtained by adding the messages from all incident edges. Note that as in the original QMF strategy, no hard decisions are made on the relay transmissions.

6.5.3 Non-binary signaling

To extend our scheme to a 2^k-QAM constellation, we adopt a 2-step procedure–namely coding in binary, followed by modulating the coded bits to the transmit constellation. The (non-binary) code and membership functions then factorize as

$$\mathbb{1}_{\{\mathbf{x}_s\in\mathcal{C}\}} = \mathbb{1}_{\{\mathbf{c}_s\in\mathcal{C}'\}}\cdot\mathbb{1}_{\{\Psi(\mathbf{c}_s)=\mathbf{x}_s\}} \tag{6.16}$$

where $\mathbf{c}_s$ denotes the *binary* coded and mapped vectors at the source and Ψ is the vector extension of the constellation map. $\mathcal{C}'$ denotes the binary source codebook. From the perspective of the decoder, each type α node now has k connecting edges to $\mathcal{G}$ and Π while each type β check node has k connecting edges to $\mathcal{G}$ and Π. The description of the iterative decoder for the DIQIF scheme can easily be specialized or adapted to the other schemes that we consider.

6.6 Experimental evaluation

In this section, we experimentally evaluate QUILT and compare it with alternative cooperative communication strategies. We first describe our performance metrics (Section 6.6.1)

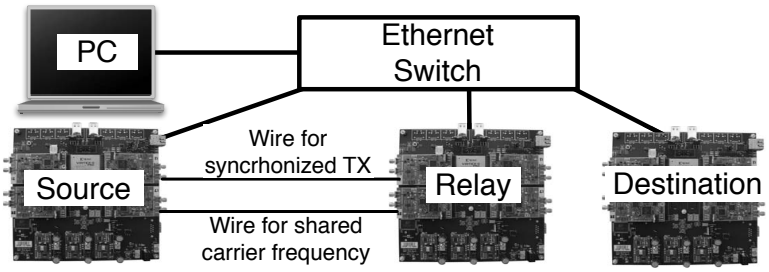


Figure 6.5 – Node and host PC configuration

and testbed (Section 6.6.2), then present our experimental results (Sections 6.6.3, 6.6.4, 6.6.5 and 6.6.6).

6.6.1 Performance Metrics

We consider the following metrics:

- *Frame-Error Rate* (FER): The percentage of source packets that were not decoded after both phases.
- *Throughput*: The number of information bits successfully delivered to the destination per channel use (bps/Hz).

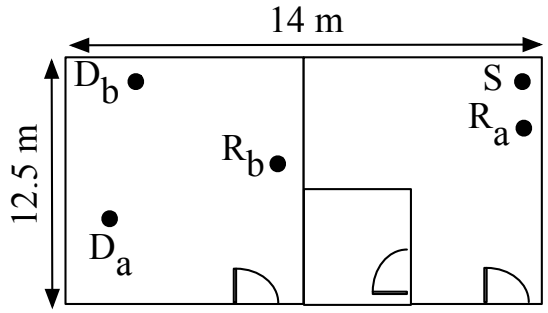
6.6.2 Testbed

We used the WARP SDR hardware to implement the source, relay and destination nodes in our testbed. We used the WARPLab framework to interact with the WARP hardware via a host PC running MATLAB. The host PC was connected to the nodes via an Ethernet switch as shown in Figure 6.5. The samples to be transmitted by a node were generated in MATLAB and downloaded to the transmit buffer of the corresponding node. The host PC triggered a real-time over-the-air transmission and reception by the nodes. The samples received at a node were read by the host PC and processed in MATLAB. The transmitted waveforms were centered at 2.4 GHz and had a 20 MHz bandwidth.

We evaluate the performance of the protocols for different experiment scenarios which were obtained by keeping the source fixed and varying the relay and destination placement and source and relay powers. The node locations for each of the three scenarios considered are shown in Fig. 6.6 and the Received Signal Strength Indicator (RSSI) for each link for each scenario is shown in Fig. 6.7.

For each setting, we ran the experiment for at least 2500 coded frames. In all experiments, we used randomly chosen bit-interleavers of length equal to that of an LDPC codeword.

We used 16-QAM constellations with a coding rate of 3/4.



	Scenario 1	Scenario 2	Scenario 3
Source Location	S	S	S
Relay Location	R_a	R_b	R_b
Destination Location	D_a	D_b	D_a

R_a and R_b: Locations considered for the relay
D_a and D_b: Locations considered for the destination

Figure 6.6 – Node placement illustrating the topologies considered.

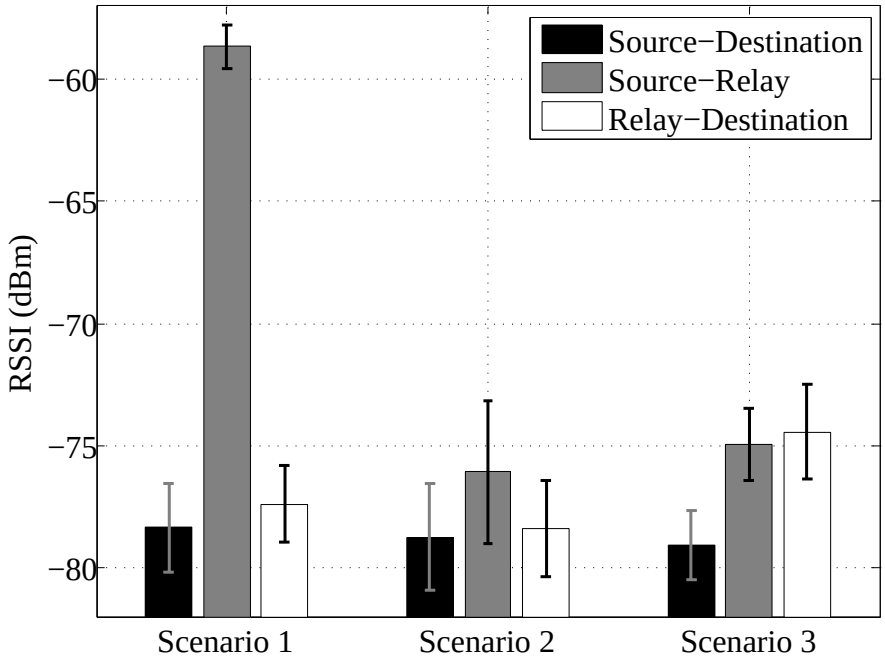
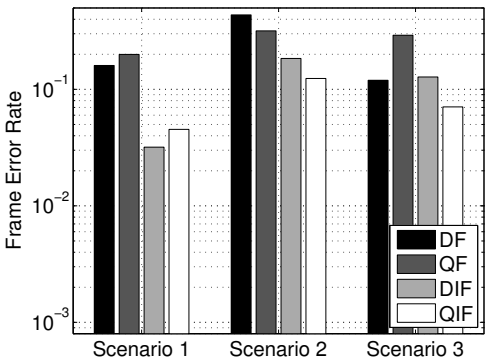


Figure 6.7 – RSSIs for the different settings considered.

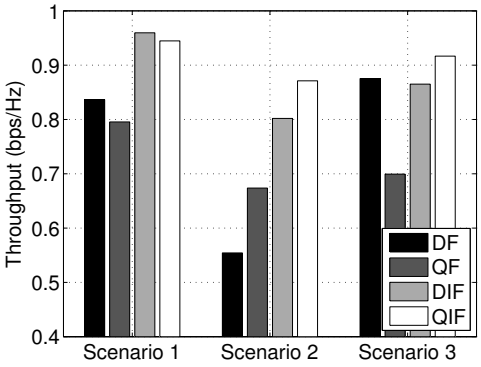
6.6.3 Evaluation of Interleaving

We observed in Section 6.3 that interleaving can significantly improve the outage probability of QIF vs. QF[2] (see Figure 6.2a). The theoretical evaluation was only possible for short interleavers, and across a single subcarrier. The question is: how much interleaving helps when we use long interleavers across subcarriers?

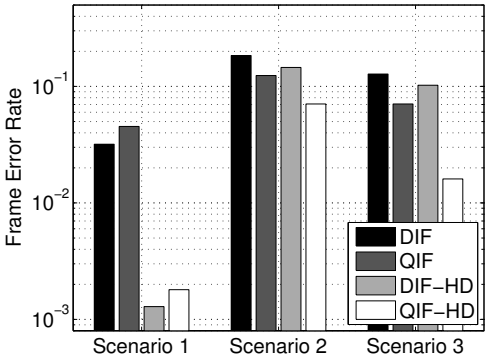
[2]We emphasize once again that the random mapping version of QMF in [8] is not an *implementable* strategy due to complexity limitations. Moreover, we have shown in Section 6.3 that QIF outperforms random mapping.



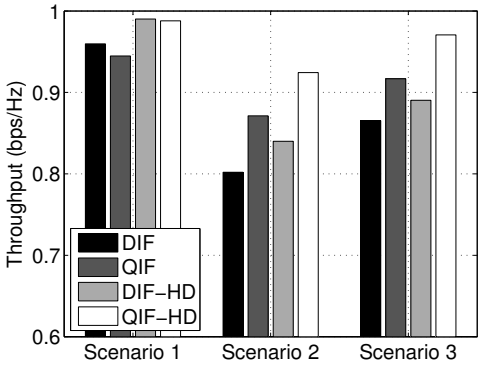
(a) FER benefits of interleaving.



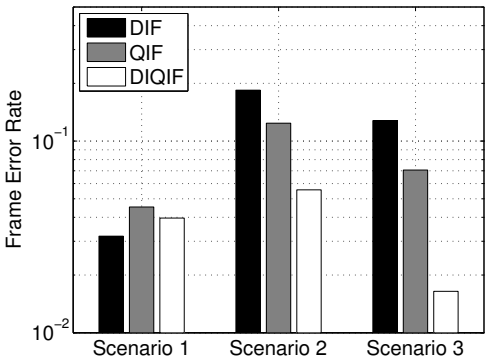
(b) Throughput benefits of interleaving.



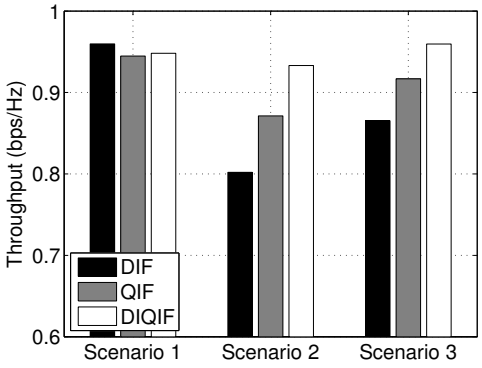
(c) FER benefits of hybrid decoding.



(d) Throughput benefits of hybrid decoding.



(e) FER benefits of opportunistic decoding.



(f) Throughput benefits of opportunistic decoding.

Figure 6.8 – FER and throughput benefits of interleaving, hybrid decoding, and opportunistic decoding.

Fig. 6.8a and 6.8b present the performance of DF, QF, DIF and QIF. We note that for these experiments, we allowed DF to implement an Alamouti code when the source and the relay cooperatively transmit in the phase 2, thus achieving full spatial diversity. We

make the following observations.

First, QIF outperforms QF in all three scenarios, with **throughput gains** ranging from **15**% to **30**%. We expected significant benefits, as interleaving enables to capture space-frequency diversity. Infact, it was shown in [58] that interleaving is sufficient to extract full spatial diversity from distributed transmissions for single carrier systems; here, we have the additional benefit of capturing frequency diversity through mixing signals across OFDM subcarriers.

Second, although DF achieves full spatial diversity due to the Alamouti code, DIF still offers benefits, up to an impressive **45**% throughput gain (Scenario 2, Fig. 6.8b). This reflects the additional frequency diversity gain from interleaving.

Third, Scenario 1 provides evidence that DIF can in some cases outperform QIF.

6.6.4 Evaluation of Hybrid Decoding

Next, we investigate the effect of relay-assisted hybrid decoding. Fig. 6.8c and 6.8d compare the performance of DIF and QIF, which in the second phase utilize only the second transmission for decoding, with that of DIF-HD and QIF-HD, which combine the received signals in both phases 1 and 2 when decoding. We observe that:

First, hybrid decoding consistently offers benefits for both QIF and DIF across all the three scenarios, for instance up to **25 times** FER improvement (in Scenario 1, Fig. 6.8c).

Second, hybrid decoding makes a more significant difference when the channels are less noisy, i.e., we start with lower FER, as is the case in Scenario 1. This is because there are comparatively fewer errors in the erroneous codewords, which can be corrected with hybrid decoding.

Third, hybrid decoding can help QIF more than DIF, as we see in Scenarios 2 and 3. This is because with DIF, when the relay cannot decode it remains silent in phase 2; while with QIF the relay always transmits potentially useful information that can be leveraged through hybrid decoding across both phases, which is reflected in the QIF-HD performance.

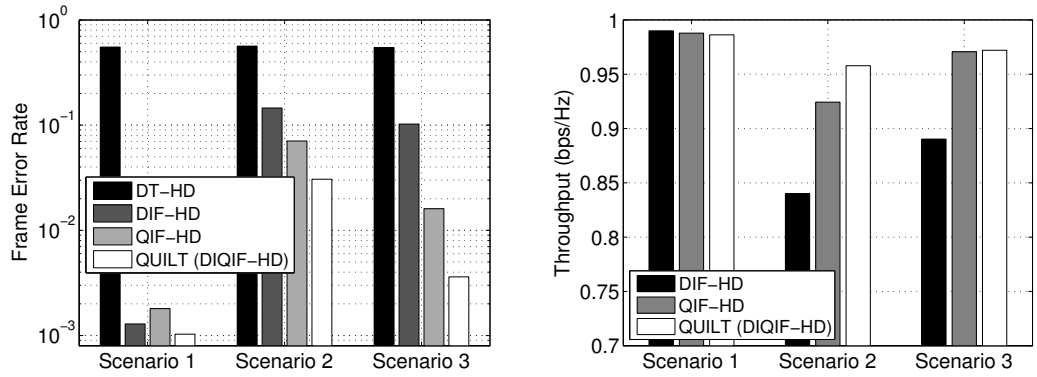
6.6.5 Evaluation of Opportunistic Decoding or Quantizing

To explore the performance of opportunistic decoding/quantizing at the relay, Fig. 6.8e and 6.8f compare the FER and throughput of DIF and QIF vs. DIQIF. We find that:

DIQIF, that implements opportunistic decoding/quantizing, has competitive or better performance than the next best scheme, as high as **a factor of 8** over DIF and **a**

factor of 5 over QIF (as in Scenario 3, Fig. 6.8e). The benefits of DIQIF are more pronounced when the source-to-relay link is weak, as is the case in Scenarios 2 and 3. This is because, in such cases the relay cannot decode, and DIF cannot exploit the relay-destination channel, while DIQIF can. Moreover, although QIF outperforms DIF in terms of FER, there exist frames where relay decoding is possible, and the opportunistic DIQIF decoding enables to clean them up from the source-relay noise, thus boosting the end-to-end performance. In Scenario 1, on the other hand, the source-relay link is very strong and supports relay decoding almost all the time; the DIQIF relay also performs decoding, but has the added requirement of communicating a 1-bit flag to inform the destination whether it decoded; we believe it is errors in this bit that result in the marginal penalty of the DIQIF performance over DIF.

6.6.6 Putting it All Together: Evaluation of QUILT



(a) FER benefits of QUILT.

(b) Throughput benefits of QUILT.

Figure 6.9 – Performance of QUILT.

We compare in Fig. 6.9a the FER performance of QUILT with (i) DIF-HD and QIF-HD, the most competing strategies implemented in this chapter, and (ii) DT-HD, direct transmissions with hybrid decoding, to benchmark the performance of a system without a relay. We observe the following:

First, we note FER gains of over **2 orders of magnitude** of our relaying strategies vs. DT-HD (in Scenario 1, Fig. 6.9a), clearly illustrating the benefits of relaying.

Second, QUILT has competitive or better performance than the next best scheme, up to **a factor of 5** over QIF-HD (in Scenario 3, Fig. 6.9a). In Scenario 1, where the source-relay link is very strong, we observed very few errors for DIF-HD, QIF-HD and QUILT (even after running the experiments in this scenario for over 4000 frames) as hybrid decoding cleans up most errors in this setup, leading to similar performance across the three schemes (marginally better for QUILT).

Since we operate at quite low FERs, we note that the vast majority of transmissions are successful and thus, the difference in fraction of frames correctly decoded does not lead to discernible throughput differences in Fig. 6.9b. However, when operating at higher FERs, we believe that the FERs trends evidenced in Fig. 6.9a will lead to more significant throughput differences, as was the case in Fig. 6.8d, 6.8f.

Overall, we find that QUILT, by synthesizing opportunistic selection of decoding/quantizing and interleaving at the relay with hybrid decoding at the destination, achieves universally competitive performance across all the scenarios we examined.

Discussion and Open Problems

In this thesis, we have considered several problems in wireless networks from the perspective of reducing the complexity of coding, scheduling and relaying. In the first five chapters we presented various theoretical results on low complexity coding and relaying while in the final chapter we presented an approach for implementing a low complexity relaying scheme in practice.

Several open questions remain that can be a focus of future work. We list a few possibilities from each of the six chapters below.

- Pliable Index Coding – In Chapter 1, we considered two extreme scenarios where the source either has full side information (PICOD) or very limited side information (OB-PICOD). Future work will include generalizing the results to a situation where the source has partial side information (PICOD and OB-PICOD will be two special cases) and also considering a scenario where there are weights (preferences) associated with messages (index coding and PICOD will be two special cases).

- Complexity of Schedules – In Chapter 2, we conjectured and proved (for $n \leq 6$) that the number of active states in the approximately optimal schedule in an n-relay half-duplex diamond network is atmost $n+1$. Future work will include proving the conjecture analytically for any n and developing polynomial time algorithms for finding the optimal schedule.

- Routing Strategies – In Chapter 3, we showed that simple routing strategies using 2 relays and 2 scheduling states achieve at least half the capacity of an n-relay half-duplex diamond network, approximately. Future work will involve generalizing these results to scenarios where more than two relays and/or more than two relaying states are used.

- Local Scheduling – In Chapter 4, we showed that relaying strategies using only local channel state information and randomized switching can achieve a significant fraction of an 2-relay half-duplex diamond network. Future work will involve proving similar results for any n-relay network and also proving the claims for finite number of switches greater than one.

- Relay Selection – In Chapter 5, we formulated the problem of selecting a subnetwork of relays that has the highest capacity in a layered relay network as an optimization problem and presented approximation algorithms for it. Future work will involve proving approximation guarantees for the algorithms we propose and developing faster ones.
- Practical Relaying – In Chapter 6, we presented the design of a practical relaying scheme using opportunistic decoding and quantize-and-interleave operation at the relays. Future work will involve generalizing the scheme for more than one relays, integrating synchronization and developing faster decoding algorithms.

To summarize, the results presented in this thesis point to several open problems, both theoretical and practical, in the domain of low complexity scheduling, relaying and coding.

A Appendix

A.1 Performance of $C^2_{rnd}(2)$

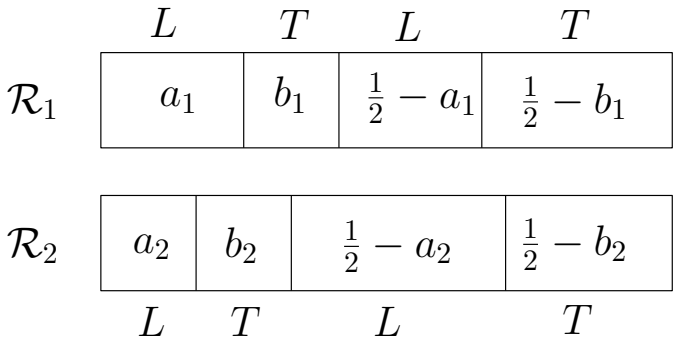


Figure A.1 – One of the 20 possible configurations of the switching points of $\mathcal{R}_1$ and $\mathcal{R}_2$.

In this section, we show that when the link strengths in a 2-relay half-duplex diamond network are all equal (assumed to be 1 here), then $C^2_{rnd}(2) = 0.7$. In this case, $C^2_{lp} = 1$, so their ratio remains 0.7. For $\sigma = 2$, there are two switches from $L \to T$ and one switch from $T \to L$ as shown in Fig. A.1. Let a_1 be the fraction of time $\mathcal{R}_1$ is in L for the first time and b_1 be the fraction of time $\mathcal{R}_1$ is in T for the first time. Since the locally optimal fraction of listening for both relays is $\frac{1}{2}$, $\mathcal{R}_1$ will be in L for $\frac{1}{2} - a_1$ and in T for $\frac{1}{2} - b_1$ in the second time. The same holds for $\mathcal{R}_2$, where a_2 and b_2 are the corresponding quantities for $\mathcal{R}_2$, as shown in the figure.

According to our randomized scheduling strategy, the values of a_1, b_1, a_2 and b_2 lie uniformly at random in the interval $[0, \frac{1}{2}]$. Depending on their values, the switching points of $\mathcal{R}_2$ can be in different relative positions with respect to the switching points of $\mathcal{R}_1$. In fact, there are precisely $\binom{6}{3} = 20$ ways in which the three switching points of $\mathcal{R}_2$ can be placed. Our proof strategy will be to calculate the value of $C^2_{rnd}(2)$ conditioned on each configuration and then take their sum.

The optimal value of **LP** is

$$C' = \min\{p_1 + p_2 + p_3, p_1 + 2p_2 + p_4, p_1 + 2p_3 + p_4, p_2 + p_3 + p_4\} \tag{A.1}$$

The value of C' for a configuration $i \in [20]$ (denoted by C'_i) can be computed by first computing the values of p_1, p_2, p_3, p_4 in terms of a_1, b_1, a_2, b_2. In addition, the configuration also imposes constraints on the values of a_1, b_1, a_2, b_2. This defines a region that is a part of $[0, \frac{1}{2}]^4$, which we denote by Γ_i. To find the expected value of C'_i, we need to integrate C'_i over Γ_i.

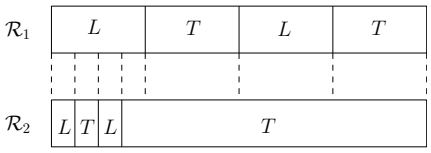
$$\mathbb{E}[C'_i] = \int_{\Gamma_i} 16 \cdot C'_i \cdot da_1 db_1 da_2 db_2 \tag{A.2}$$

The factor of 16 comes because the P.d.f of each of the variables a_1, b_1, a_2, b_2 is 2 for the range $[0, \frac{1}{2}]$ and 0 otherwise.

We are now in a position to compute $\mathbb{E}[C'_i]$ for each $i \in [1, 20]$. A pictorial representation is shown for each configuration, along with the constraints on a_1, b_1, a_2, b_2 that defines Γ_i. The following global constraints are always present and are not shown explicitly.

$$0 \leq a_1, b_1, a_2, b_2 \leq \frac{1}{2} \tag{A.3}$$

Configuration 1

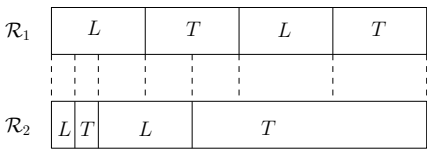


Fractions	Constraints
$p_1 = a_2 + (\frac{1}{2} - a_2) = \frac{1}{2}$	None
$p_2 = b_2 + (a_1 - (\frac{1}{2} + b_2)) + (\frac{1}{2} - a_1) = 0$	$a_1 \geq \frac{1}{2} + b_2$
$p_3 = 0$	None
$p_4 = b_1 + (\frac{1}{2} - b_1) = \frac{1}{2}$	None

$$C'_1 = \frac{1}{2} \tag{A.4}$$

$$\mathbb{E}[C'_1] = 0 \tag{A.5}$$

Configuration 2

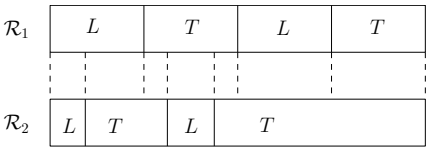


Fractions	Constraints
$p_1 = a_2 + (a_1 - (a_2 + b_2)) = a_1 - b_2$	$a_1 \geq a_2 + b_2$
$p_2 = b_2 + (\frac{1}{2} - a_1)$	None
$p_3 = \frac{1}{2} + b_2 - a_1$	None
$p_4 = (a_1 + b_1 - (\frac{1}{2} + b_2)) + (\frac{1}{2} - b_1) = a_1 - b_2$	$a_1 + b_1 \geq \frac{1}{2} + b_2$

$$C'_2 = 1 - a_1 + b_2 \tag{A.6}$$

$$\mathbb{E}[C'_2] = 16 \int_0^{\frac{1}{2}} \int_{b_2}^{\frac{1}{2}} \int_{\frac{1}{2}(1-2a_1+2b_2)}^{\frac{1}{2}} \int_0^{a1-b2} (1 - a_1 + b_2)\, da_2 db_1 da_1 db_2 = \frac{7}{120} \tag{A.7}$$

Configuration 3

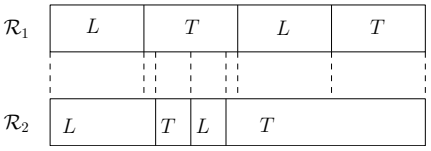


Fractions	Constraints
$p_1 = a_1$	None
$p_2 = (a_1 - a_2) + (\frac{1}{2} - a_1) = \frac{1}{2} - a_2$	$a_1 \geq a_2$
$p_3 = \frac{1}{2} - a_2$	None
$p_4 = (a_2 + b_2 - a_1) + (a_1 + b_1 - (\frac{1}{2} + b_2)) + (\frac{1}{2} - b_1) = a_2$	$a_2 + b_2 \geq a_1, a_1 + b_1 \geq \frac{1}{2} + b_2$

$$C'_3 = 1 - a_2 \tag{A.8}$$

$$\mathbb{E}[C'_3] = 16 \int_0^{\frac{1}{2}} \int_{b_2}^{\frac{1}{2}} \int_{a_1-b_2}^{a_1} \int_{\frac{1}{2}(1-2a_1+2b_2)}^{\frac{1}{2}} (1 - a_2)\, db_1 da_2 da_1 db_2 = \frac{7}{240} \tag{A.9}$$

Configuration 4

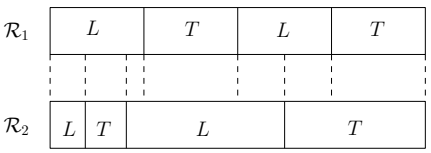


Fractions	Constraints
$p_1 = a_1$	None
$p_2 = \frac{1}{2} - a_1$	None
$p_3 = (a_2 - a_1) + (\frac{1}{2} - a_2) = \frac{1}{2} - a_1$	$a_2 \geq a_1$
$p_4 = b_2 + (a_1 + b_1 - (\frac{1}{2} + b_2)) + (\frac{1}{2} - b_1) = a_1$	$a_1 + b_1 \geq \frac{1}{2} + b_2$

$$C'_4 = 1 - a_1 \tag{A.10}$$

$$\mathbb{E}[C'_4] = 16 \int_0^{\frac{1}{2}} \int_{\frac{1}{2}(1-2b_1)}^{\frac{1}{2}} \int_0^{\frac{1}{2}(2a_1+2b_1-1)} \int_{a_1}^{\frac{1}{2}} (1 - a_1)\, da_2 db_2 da_1 db_1 = \frac{7}{240} \tag{A.11}$$

Configuration 5



Fractions	Constraints
$p_1 = a_2 + (a_1 - (a_2 + b_2)) + (b_2 + \frac{1}{2} - (a_1 + b_1)) = \frac{1}{2} - b_1$	$a_1 \geq a_2 + b_2, b_2 + \frac{1}{2} \geq a_1 + b_1$
$p_2 = b_2 + (\frac{1}{2} + b_1 - (\frac{1}{2} + b_2)) = b_1$	$b_1 \geq b_2$
$p_3 = b_1$	None
$p_4 = \frac{1}{2} - b_1$	None

$$C_5' = 1 - a_1 \tag{A.12}$$

$$\mathbb{E}[C_5'] = 16 \int_0^{\frac{1}{4}} \int_0^{b_1} \int_0^{a_1} \int_0^{a_1-a_2} (\frac{1}{2} + b_1)\, db_2 da_2 da_1 db_1 + \tag{A.13}$$

$$16 \int_0^{\frac{1}{4}} \int_{b_1}^{\frac{1}{2}-b_1} \int_0^{a_1-b_1} \int_0^{b_1} (\frac{1}{2} + b_1)\, db_2 da_2 da_1 db_1 + \tag{A.14}$$

$$16 \int_0^{\frac{1}{4}} \int_{b_1}^{\frac{1}{2}-b_1} \int_{a_1-b_1}^{a_1} \int_0^{a_1-a_2} (\frac{1}{2} + b_1)\, db_2 da_2 da_1 db_1 + \tag{A.15}$$

$$16 \int_0^{\frac{1}{4}} \int_{\frac{1}{2}-b_1}^{\frac{1}{2}} \int_0^{a_1-b_1} \int_{a_1+b_1-\frac{1}{2}}^{b_1} (\frac{1}{2} + b_1)\, db_2 da_2 da_1 db_1 + \tag{A.16}$$

$$16 \int_0^{\frac{1}{4}} \int_{\frac{1}{2}-b_1}^{\frac{1}{2}} \int_{a_1-b_1}^{\frac{1}{2}-b_1} \int_{a_1+b_1-\frac{1}{2}}^{a_1-a_2} (\frac{1}{2} + b_1)\, db_2 da_2 da_1 db_1 + \tag{A.17}$$

$$16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_0^{\frac{1}{2}-b_1} \int_0^{a_1} \int_0^{a_1-a_2} (\frac{1}{2} + b_1)\, db_2 da_2 da_1 db_1 + \tag{A.18}$$

$$16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}-b_1}^{b_1} \int_0^{\frac{1}{2}-b_1} \int_{a_1+b_1-\frac{1}{2}}^{a_1-a_2} (\frac{1}{2} + b_1)\, db_2 da_2 da_1 db_1 + \tag{A.19}$$

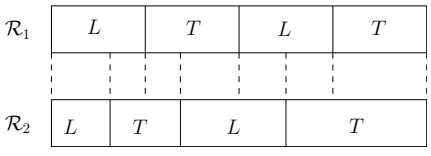
$$16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} \int_0^{a_1-b_1} \int_{a_1+b_1-\frac{1}{2}}^{b_1} (\frac{1}{2} + b_1)\, db_2 da_2 da_1 db_1 + \tag{A.20}$$

$$16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} \int_{a_1-b_1}^{\frac{1}{2}-b_1} \int_{a_1+b_1-\frac{1}{2}}^{a_2-a_1} (\frac{1}{2} + b_1)\, db_2 da_2 da_1 db_1 \tag{A.21}$$

$$= \frac{7}{3840} + \frac{1}{160} + \frac{13}{3840} + \frac{1}{192} + \frac{7}{3840} + \frac{1}{480} + \frac{17}{3840} + \frac{1}{480} + \frac{1}{480} \tag{A.22}$$

$$= \frac{7}{240} \tag{A.23}$$

Configuration 6



Fractions	Constraints
$p_1 = a_2 + (\frac{1}{2} + b_2 - (a_1 + b_1)) = \frac{1}{2} + a_2 + b_2 - a_1 - b_1$	$\frac{1}{2} + b_2 \geq a_1 + b_1$
$p_2 = (a_1 - a_2) + (\frac{1}{2} + b_1 - (\frac{1}{2} + b_2)) = a_1 + b_1 - a_2 - b_2$ $a_1 \geq a_2$,$b_1 \geq b_2$	None
$p_3 = a_1 + b_1 - a_2 - b_2$	$a_1 + b_1 \geq a_2 + b_2$
$p_4 = (a_2 + b_2 - a_1) + (\frac{1}{2} - b_1) = \frac{1}{2} + a_2 + b_2 - a_1 - b_1$	$a_2 + b_2 \geq a_1$

$$C_6' = \frac{1}{2} + a_1 + b_1 - a_2 - b_2 \tag{A.24}$$

$$\mathbb{E}[C_6'] = 16 \int_0^{\frac{1}{24}} \int_0^{b_1} \int_0^{a_1} \int_{a_1-b_2}^{a_1} (\frac{1}{2} + a_1 + b_1 - a_2 - b_2)\, da_2 db_2 da_1 db_1 + \tag{A.25}$$

$$16 \int_0^{\frac{1}{24}} \int_0^{b_1} \int_{a_1}^{b_1} \int_0^{a_1} (\frac{1}{2} + a_1 + b_1 - a_2 - b_2)\, da_2 db_2 da_1 db_1 + \tag{A.26}$$

$$16\int_0^{\frac{1}{24}}\int_{b_1}^{\frac{1}{2}-b_1}\int_0^{b_1}\int_{a_1-b_2}^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.27}$$

$$16\int_0^{\frac{1}{24}}\int_{\frac{1}{2}-b_1}^{\frac{1}{2}}\int_{a_1+b_1-\frac{1}{2}}^{b_1}\int_{a_1-b_2}^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.28}$$

$$16\int_{\frac{1}{24}}^{\frac{1}{2}}\int_0^{b_1}\int_0^{a_1}\int_{a_1-b_2}^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.29}$$

$$16\int_{\frac{1}{24}}^{\frac{1}{2}}\int_0^{b_1}\int_{a_1}^{b_1}\int_0^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.30}$$

$$16\int_{\frac{1}{24}}^{\frac{1}{2}}\int_{b_1}^{\frac{1}{2}-b_1}\int_0^{b_1}\int_{a_1-b_2}^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.31}$$

$$16\int_{\frac{1}{24}}^{\frac{1}{2}}\int_{\frac{1}{2}-b_1}^{\frac{1}{2}}\int_{a_1+b_1-\frac{1}{2}}^{b_1}\int_{a_1-b_2}^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.32}$$

$$16\int_{\frac{1}{2}}^{\frac{1}{2}}\int_0^{\frac{1}{2}-b_1}\int_0^{a_1}\int_{a_1-b_2}^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.33}$$

$$16\int_{\frac{1}{2}}^{\frac{1}{2}}\int_0^{\frac{1}{2}-b_1}\int_{a_1}^{b_1}\int_0^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.34}$$

$$16\int_{\frac{1}{2}}^{\frac{1}{2}}\int_{\frac{1}{2}-b_1}^{b_1}\int_{a_1+b_1-\frac{1}{2}}^{a_1}\int_{a_1-b_2}^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.35}$$

$$16\int_{\frac{1}{2}}^{\frac{1}{2}}\int_{\frac{1}{2}-b_1}^{b_1}\int_{a_1}^{b_1}\int_0^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1+ \tag{A.36}$$

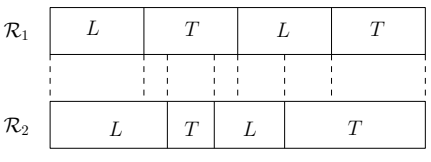
$$16\int_{\frac{1}{2}}^{\frac{1}{2}}\int_{b_1}^{\frac{1}{2}}\int_{a_1+b_1-\frac{1}{2}}^{b_1}\int_{a_1-b_2}^{a_1}(\frac{1}{2}+a_1+b_1-a_2-b_2)\,da_2db_2da_1db_1 \tag{A.37}$$

$$\mathbb{E}[C_6'] = \frac{7}{6635520}+\frac{31}{29859840}+\frac{41}{933120}+\frac{25}{11943936}+\frac{2245}{1327104}+\frac{9325}{5971968} \tag{A.38}$$

$$+\frac{575}{186624}+\frac{38855}{11943936}+\frac{1}{512}+\frac{1}{192}+\frac{23}{1920}+\frac{43}{1920}+\frac{11}{1536} \tag{A.39}$$

$$=\frac{7}{120} \tag{A.40}$$

Configuration 7



Fractions	Constraints
$p_1 = a_1 + (\frac{1}{2} + b_2 - (a_1 + b_1))$	$\frac{1}{2} + b_2 \geq a_1 + b_1$
$p_2 = b_1 - b_2$	$b_1 \geq b_2$
$p_3 = (a_2 - a_1) + (a_1 + b_1 - (a_2 + b_2)) = b_1 - b_2$	$a_2 \geq a_1,\ a_1 + b_1 \geq a_2 + b_2$
$p_4 = \frac{1}{2} + b_2 - b_1$	None

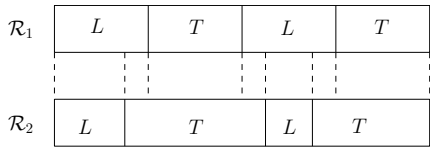
$$C_7' = 1 - a_1 \tag{A.41}$$

$$\mathbb{E}[C_7'] = 16 \int_0^{\frac{1}{2}} \int_0^{\frac{1}{2}-b_1} \int_{a_1}^{a_1+b_1} \int_0^{a_1-a_2+b_1} (\frac{1}{2} + b_1 - b_2)\, db_2 da_2 da_1 db_1 + \tag{A.42}$$

$$16 \int_0^{\frac{1}{2}} \int_{\frac{1}{2}-b_1}^{\frac{1}{2}} \int_{a_1}^{\frac{1}{2}} \int_{a_1+b_1-\frac{1}{2}}^{a_1-a_2+b_1} (\frac{1}{2} + b_1 - b_2)\, db_2 da_2 da_1 db_1 \tag{A.43}$$

$$= \frac{7}{240} + \frac{7}{240} = \frac{7}{120} \tag{A.44}$$

Configuration 8

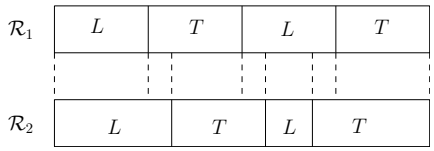


Fractions	Constraints
$p_1 = \frac{1}{2}$	None
$p_2 = (a_1 - a_2) + (a_2 + b_2 - (a_1 + b_1)) + (\frac{1}{2} + b_1 - (\frac{1}{2} + b_2)) = 0$	$a_1 \geq a_2$,$a_2 + b_2 \geq a_1 + b_1$, $b_1 \geq b_2$
$p_3 = 0$	None
$p_4 = \frac{1}{2}$	None

$$C_8' = \frac{1}{2} \tag{A.45}$$

$$\mathbb{E}[C_8'] = 0 \tag{A.46}$$

Configuration 9



Fractions	Constraints
$p_1 = \frac{1}{2} + a_1 - a_2$	None
$p_2 = (a_2 + b_2 - (a_1 + b_1)) + (\frac{1}{2} + b_1 - (\frac{1}{2} + b_2)) = 0$	$a_2 + b_2 \geq a_1 + b_1$, $b_1 \geq b_2$
$p_3 = a_2 - a_1$	$a_2 \geq a_1$
$p_4 = (a_1 + b_1 - a_2) + \frac{1}{2} - b_1 = \frac{1}{2} + a_1 - a_2$	$a_1 + b_1 \geq a_2$

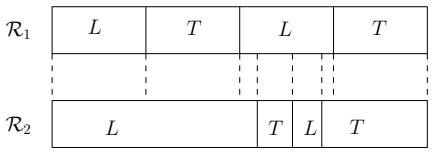
$$C_9' = \frac{1}{2} + a_2 - a_1 \tag{A.47}$$

$$\mathbb{E}[C_9'] = 16 \int_0^{\frac{1}{2}} \int_0^{\frac{1}{2}-b_1} \int_{a_1}^{a_1+b_1} \int_{a_1-a_2+b_1}^{b_1} (\frac{1}{2} + a_2 - a_1)\, db_2 da_2 da_1 db_1 + \tag{A.48}$$

$$16 \int_0^{\frac{1}{2}} \int_{\frac{1}{2}-b_1}^{\frac{1}{2}} \int_{a_1}^{\frac{1}{2}} \int_{a_1-a_2+b_1}^{b_1} (\frac{1}{2} + a_2 - a_1)\, db_2 da_2 da_1 db_1 \tag{A.49}$$

$$= \frac{7}{240} + \frac{7}{240} = \frac{7}{120} \tag{A.50}$$

Configuration 10



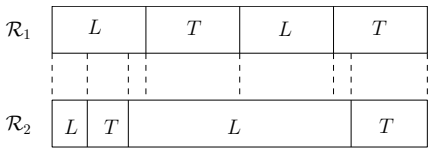
Fractions	Constraints
$p_1 = a_1 + (a_2 - (a_1 + b_1)) + (\frac{1}{2} - a_2) = \frac{1}{2} - b_1$	$a_2 \geq a_1 + b_1$
$p_2 = b_2 + (\frac{1}{2} + b_1 - (\frac{1}{2} + b_2)) = b_1$	$b_1 \geq b_2$
$p_3 = b_1$	None
$p_4 = \frac{1}{2} - b_1$	None

$$C'_{10} = \frac{1}{2} + b_1 \tag{A.51}$$

$$\mathbb{E}[C'_{10}] = 16 \int_0^{\frac{1}{2}} \int_0^{\frac{1}{2}-b_1} \int_0^{b_1} \int_{a_1+b_1}^{\frac{1}{2}} (\frac{1}{2} + b_1)\, da_2 db_2 da_1 db_1 \tag{A.52}$$

$$= \frac{7}{240} \tag{A.53}$$

Configuration 11



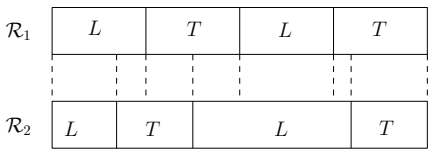
Fractions	Constraints
$p_1 = a_2 + (a_1 - (a_2 + b_2)) = a_1 - b_2$	$a_1 \geq a_2 + b_2$
$p_2 = b_2 + \frac{1}{2} - a_1$	None
$p_3 = b_1 + (\frac{1}{2} + b_2 - (\frac{1}{2} + b_1)) = b_2$	$b_2 \geq b_1$
$p_4 = \frac{1}{2} - b_2$	None

$$C'_{11} = \frac{1}{2} + b_1 \tag{A.54}$$

$$\mathbb{E}[C'_{11}] = 16 \int_0^{\frac{1}{2}} \int_{b_2}^{\frac{1}{2}} \int_0^{b_1} \int_0^{a_1-b_2} (\frac{1}{2} + b_1)\, da_2 db_1 da_1 db_2 \tag{A.55}$$

$$= \frac{7}{240} \tag{A.56}$$

Configuration 12

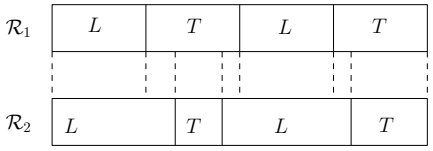


Fractions	Constraints
$p_1 = \frac{1}{2} + a_2 - a_1$	None
$p_2 = a_1 - a_2$	$a_1 \geq a_2$
$p_3 = (a_1 + b_1 - (a_2 + b_2)) + (\frac{1}{2} + b_2 - (\frac{1}{2} + b_1)) = a_1 - a_2$	$a_1 + b_1 \geq a_2 + b_2$, $b_2 \geq b_1$
$p_4 = (a_2 + b_2 - a_1) + (\frac{1}{2} - b_2) = \frac{1}{2} + a_2 - a_2$	$a_2 + b_2 \geq a_1$

$$C'_{12} = \frac{1}{2} + a_1 - a_2 \tag{A.57}$$

$$\begin{aligned}
\mathbb{E}[C'_{12}] =& 16 \int_0^{\frac{1}{2}} \int_0^{a_1} \int_0^{a_1-b_1} \int_{a_1-a_2}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.58)}\\
& 16 \int_0^{\frac{1}{2}} \int_0^{a_1} \int_{a_1-b_1}^{a_1} \int_{b_1}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.59)}\\
& 16 \int_0^{\frac{1}{2}} \int_{a_1}^{\frac{1}{2}-a_1} \int_0^{a_1} \int_{b_1}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.60)}\\
& 16 \int_0^{\frac{1}{2}} \int_{\frac{1}{2}-a_1}^{\frac{1}{2}} \int_0^{a_1+b_1-\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.61)}\\
& 16 \int_0^{\frac{1}{2}} \int_{\frac{1}{2}-a_1}^{\frac{1}{2}} \int_{a_1+b_1-\frac{1}{2}}^{a_1} \int_{b_1}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.62)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_0^{\frac{1}{2}-a_1} \int_0^{a_1-b_1} \int_{a_1-a_2}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.63)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_0^{\frac{1}{2}-a_1} \int_{a_1-b_1}^{a_1} \int_{b_1}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.64)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}-a_1}^{\frac{1}{2}} \int_0^{a_1+b_1-\frac{1}{2}} \int_{a_1-a_2}^{\frac{1}{2}} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.65)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}-a_1}^{\frac{1}{2}} \int_{a_1+b_1-\frac{1}{2}}^{a_1-b_1} \int_{a_1-a_2}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.66)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}-a_1}^{\frac{1}{2}} \int_{a_1-b_1}^{a_1} \int_{b_1}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.67)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}}^{a_1} \int_0^{a_1-b_1} \int_{a_1-a_2}^{\frac{1}{2}} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.68)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}}^{a_1} \int_{a_1-b_1}^{a_1+b_1-\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.69)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}}^{a_1} \int_{a_1+b_1-\frac{1}{2}}^{a_1} \int_{b_1}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.70)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{a_1}^{\frac{1}{2}} \int_0^{a_1+b_1-\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 + && \text{(A.71)}\\
& 16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{a_1}^{\frac{1}{2}} \int_{a_1+b_1-\frac{1}{2}}^{a_1} \int_{b_1}^{a_1-a_2+b_1} (\frac{1}{2} + a_1 - a_2)\, db_2 da_2 db_1 da_1 && \text{(A.72)}\\
& = \frac{13}{7680} + \frac{1}{640} + \frac{1}{320} + \frac{13}{7680} + \frac{1}{640} + \frac{43}{7680} + \frac{1}{640} + \frac{17}{3840} + \frac{1}{128} && \text{(A.73)}\\
& + \frac{19}{3840} + \frac{17}{3840} + \frac{1}{128} + \frac{19}{3840} + \frac{43}{7680} + \frac{1}{640} && \text{(A.74)}\\
& = \frac{7}{120} && \text{(A.75)}
\end{aligned}$$

Configuration 13

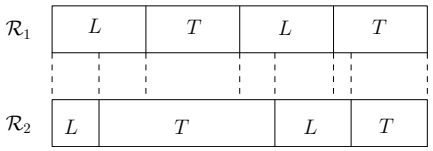


Fractions	Constraints
$p_1 = \frac{1}{2}$	None
$p_2 = 0$	None
$p_3 = (a_1 + b_1 - (a_2 + b_2)) + (\frac{1}{2} + b_2 - (\frac{1}{2} + b_1)) + (a_2 - a_1) = 0$	$a_1 + b_1 \geq a_2 + b_2$,$b_2 \geq b_1$,$a_2 \geq a_1$
$p_4 = \frac{1}{2}$	None

$$C'_{13} = \frac{1}{2} \tag{A.76}$$

$$\mathbb{E}[C'_{13}] = 0 \tag{A.77}$$

Configuration 14



Fractions	Constraints
$p_1 = a_2 + (\frac{1}{2} + b_1 - (a_2 + b_2)) = \frac{1}{2} + b_1 - b_2$	$\frac{1}{2} + b_1 \geq a_2 + b_2$
$p_2 = (a_1 - a_2) + (a_2 + b_2 - (a_1 + b_1)) = b_2 - b_1$	$a_1 \geq a_2$, $a_2 + b_2 \geq a_1 + b_1$
$p_3 = (\frac{1}{2} + b_2 - (\frac{1}{2} + b_1)) = b_2 - b_1$	$b_2 \geq b_1$
$p_4 = \frac{1}{2} + b_1 - b_2$	None

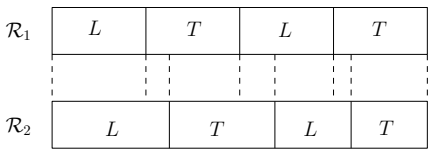
$$C'_{14} = \frac{1}{2} + b_2 - b_1 \tag{A.78}$$

$$\mathbb{E}[C'_{14}] = 16 \int_0^{\frac{1}{2}} \int_0^{b_1} \int_{a_2}^{\frac{1}{2}+a_2-b_1} \int_{a_1-a_2+b_1}^{\frac{1}{2}} (\frac{1}{2} + b_2 - b_1)\, db_2 da_1 da_2 db_1 + \tag{A.79}$$

$$16 \int_0^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} \int_{a_2}^{\frac{1}{2}} \int_{a_1-a_2+b_1}^{\frac{1}{2}-a_2+b_1} (\frac{1}{2} + b_2 - b_1)\, db_2 da_1 da_2 db_1 \tag{A.80}$$

$$= \frac{7}{240} + \frac{7}{240} = \frac{7}{120} \tag{A.81}$$

Configuration 15



Fractions	Constraints
$p_1 = a_1 + (\frac{1}{2} + b_1 - (a_2 + b_2)) = \frac{1}{2} + a_1 + b_1 - a_2 - b_2$	$\frac{1}{2} + b_1 \geq a_2 + b_2$
$p_2 = a_2 + b_2 - a_1 - b_1$	$a_2 + b_2 \geq a_1 + b_1$
$p_3 = (a_2 - a_1) + (\frac{1}{2} + b_2 - (\frac{1}{2} + b_1)) = a_2 + b_2 - a_1 - b_1$	$a_2 \geq a_1$, $b_2 \geq b_1$
$p_4 = (a_1 + b_1 - a_2) + (\frac{1}{2} - b_2)$	$a_1 + b_1 \geq a_2$

$$C'_{15} = \frac{1}{2} + a_2 + b_2 - a_1 - b_1 \tag{A.82}$$

$$\mathbb{E}[C'_{15}] = 16 \int_0^{\frac{1}{2}} \int_0^{b_1} \int_{a_1}^{b_1} \int_{b_1}^{\frac{1}{2}} (\frac{1}{2} + a_2 + b_2 - a_1 - b_1)\, db_2 da_2 da_1 db_1 + \tag{A.83}$$

$$16 \int_0^{\frac{1}{2}} \int_0^{b_1} \int_{b_1}^{a_1+b_1} \int_{b_2}^{\frac{1}{2}-a_2+b_1} (\frac{1}{2} + a_2 + b_2 - a_1 - b_1)\, db_2 da_2 da_1 db_1 + \tag{A.84}$$

$$16 \int_0^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}-b_1} \int_{a_1}^{a_1+b_1} \int_{b_1}^{\frac{1}{2}-a_2+b_1} (\frac{1}{2} + a_2 + b_2 - a_1 - b_1)\, db_2 da_2 da_1 db_1 + \tag{A.85}$$

$$16 \int_0^{\frac{1}{2}} \int_{\frac{1}{2}-b_1}^{\frac{1}{2}} \int_{a_1}^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}-a_2+b_1} (\frac{1}{2} + a_2 + b_2 - a_1 - b_1)\, db_2 da_2 da_1 db_1 + \tag{A.86}$$

$$16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_0^{\frac{1}{2}-b_1} \int_{a_1}^{b_1} \int_{b_1}^{\frac{1}{2}} (\frac{1}{2} + a_2 + b_2 - a_1 - b_1)\, db_2 da_2 da_1 db_1 + \tag{A.87}$$

$$16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_0^{\frac{1}{2}-b_1} \int_{b_1}^{a_1+b_1} \int_{b_1}^{\frac{1}{2}-a_2+b_1} (\frac{1}{2} + a_2 + b_2 - a_1 - b_1)\, db_2 da_2 da_1 db_1 + \tag{A.88}$$

$$16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}-b_1}^{b_1} \int_{a_1}^{b_1} \int_{b_1}^{\frac{1}{2}} (\frac{1}{2} + a_2 + b_2 - a_1 - b_1)\, db_2 da_2 da_1 db_1 + \tag{A.89}$$

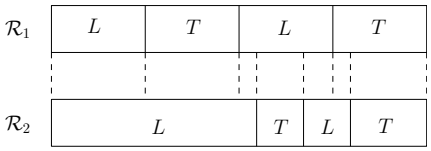
$$16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{\frac{1}{2}-b_1}^{b_1} \int_{b_1}^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}-a_2+b_1} (\frac{1}{2} + a_2 + b_2 - a_1 - b_1)\, db_2 da_2 da_1 db_1 + \tag{A.90}$$

$$16 \int_{\frac{1}{2}}^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} \int_{a_1}^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}-a_2+b_1} (\frac{1}{2} + a_2 + b_2 - a_1 - b_1)\, db_2 da_2 da_1 db_1 \tag{A.91}$$

$$\mathbb{E}[C'_{15}] = \frac{3}{320} + \frac{1}{128} + \frac{1}{96} + \frac{1}{640} + \frac{5}{384} + \frac{1}{240} + \frac{13}{1920} + \frac{7}{1920} + \frac{1}{640} \tag{A.92}$$

$$= \frac{7}{120} \tag{A.93}$$

Configuration 16



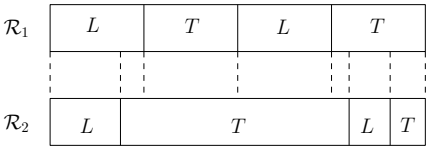
Fractions	Constraints
$p_1 = a_1 + (a_2 - (a_1 + b_1)) + (\frac{1}{2} + b_1 - (a_2 + b_2)) = \frac{1}{2} - b_2$	$a_2 \geq a_1 + b_1$, $\frac{1}{2} + b_1 \geq a_2 + b_2$
$p_2 = b_2$	None
$p_3 = b_1 + (\frac{1}{2} + b_2 - (\frac{1}{2} + b_1)) = b_2$	$b_2 \geq b_1$
$p_4 = \frac{1}{2} - b_2$	None

$$C'_{16} = \frac{1}{2} + b_2 \tag{A.94}$$

$$\mathbb{E}[C'_{16}] = 16 \int_0^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}-a_2+b_1} \int_0^{a_2-b_1} (\frac{1}{2} + b_2)\, da_1 db_2 da_2 db_1 \tag{A.95}$$

$$= \frac{7}{240} \tag{A.96}$$

Configuration 17



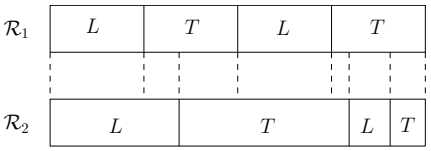
Fractions	Constraints
$p_1 = a_2$	None
$p_2 = (a_1 - a_2) + (\frac{1}{2} - a_1) = \frac{1}{2} - a_2$	$a_1 \geq a_2$
$p_3 = \frac{1}{2} - a_2$	None
$p_4 = b_1 + (a_2 + b_2 - (\frac{1}{2} + b_1)) + (\frac{1}{2} - b_2) = a_2$	$a_2 + b_2 \geq \frac{1}{2} + b_1$

$$C'_{17} = 1 - a_2 \tag{A.97}$$

$$\mathbb{E}[C'_{17}] = 16 \int_0^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} \int_{a_2}^{\frac{1}{2}} \int_{\frac{1}{2}-a_2+b_1}^{\frac{1}{2}} (1 - a_2)\, db_2 da_1 da_2 db_1 \tag{A.98}$$

$$= \frac{7}{240} \tag{A.99}$$

Configuration 18



Fractions	Constraints
$p_1 = a_1$	None
$p_2 = \frac{1}{2} - a_1$	None
$p_3 = (a_2 - a_1) + (\frac{1}{2} - a_2) = \frac{1}{2} - a_1$	$a_2 \geq a_1$
$p_4 = (a_1 + b_1 - a_2) + (a_2 + b_2 - (\frac{1}{2} + b_1)) = a_1$	$a_1 + b_1 \geq a_2$, $a_2 + b_2 \geq \frac{1}{2} + b_1$

$$C'_{18} = 1 - a_1 \tag{A.100}$$

$$\mathbb{E}[C'_{18}] = 16 \int_0^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} \int_{a_2-b_1}^{a_2} \int_{\frac{1}{2}-a_2+b_1}^{\frac{1}{2}} (1 - a_1)\, db_2 da_1 da_2 db_1 \tag{A.101}$$

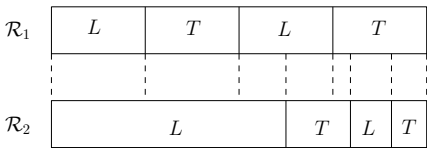
$$= \frac{7}{240} \tag{A.102}$$

Configuration 19

$$C'_{19} = 1 - a_2 + b_1 \tag{A.103}$$

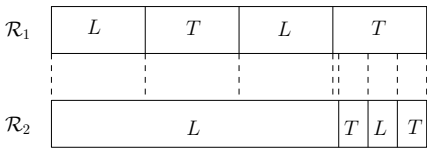
$$\mathbb{E}[C'_{19}] = 16 \int_0^{\frac{1}{2}} \int_{b_1}^{\frac{1}{2}} \int_0^{a_2-b_1} \int_{\frac{1}{2}-a_2+b_1}^{\frac{1}{2}} (1 - a_2 + b_1)\, db_2 da_1 da_2 db_1 \tag{A.104}$$

$$= \frac{7}{120} \tag{A.105}$$



Fractions	Constraints
$p_1 = a_1 + (a_2 - (a_1 + b_1)) = a_2 - b_1$	$a_2 \geq a_1 + b_1$
$p_2 = \frac{1}{2} + b_1 - a_2$	None
$p_3 = \frac{1}{2} + b_1 - a_2$	None
$p_4 = (a_2 + b_2 - (\frac{1}{2} + b_1)) + (\frac{1}{2} - b_2) = a_2 - b_1$	$a_2 + b_2 \geq \frac{1}{2} + b_1$

Configuration 20



Fractions	Constraints
$p_1 = \frac{1}{2}$	None
$p_2 = 0$	None
$p_3 = b_1 + (a_2 - (\frac{1}{2} + b_1)) + (\frac{1}{2} - a_2) = 0$	$a_2 \geq \frac{1}{2} + b_1$
$p_4 = \frac{1}{2}$	None

$$C'_{20} = \frac{1}{2} \tag{A.106}$$

$$\mathbb{E}[C'_{20}] = 0 \tag{A.107}$$

Therefore, the total expected value, which is the value of $C^2_{rnd}(2)$ is

$$C^2_{rnd}(2) = \sum_{i=1}^{20} \mathbb{E}[C'_i] = \frac{7}{10} \tag{A.108}$$

A.2 Outage Calculations

Quantized Forwarding (QF)
In the absence of relay-assisted hybrid decoding, we have,

$$\mathsf{C}_{\mathrm{R}}^{\mathrm{QF}} = I(X; Y[2]), \tag{A.109}$$

where $I(\cdot;\cdot)$ denotes the mutual information. Since the relay quantizes its received signal, the overall transformation of the source signal can be represented as an end-to-end channel whose capacity can be evaluated as above. This capacity computation can be done numerically as no closed form expressions exist for such (scalar) quantized channels with QAM inputs. For hybrid decoding, the achievable rate can be evaluated as,

$$\mathsf{C}_{\mathrm{R}}^{\mathrm{QF}} = I(X; Y[1], Y[2]), \tag{A.110}$$

which can then be again numerically computed to yield the outage probability by using (6.3).

Quantize-Interleave-Forward (QIF)
We evaluate an interleaver that operates over a block of length K as follows:

$$\mathbf{X}_{\mathrm{R}} = \Pi(\hat{\mathbf{Y}}_{\mathrm{R}})$$

where Π denotes a specific permutation on the quantized sequence $\hat{Y}_{\mathrm{R}}$. This permuted sequence is transmitted by the relay. We can numerically evaluate the rate for the interleaved scheme by using K-letter mutual information characterization, which is similar to a vector version of (A.109)-(A.110) while including the aforementioned interleaver operation.

Quantize-Map-Forward (QMF):
In the originally proposed information-theoretic QMF [8], the mapping codebook at the relay is generated randomly. The analytical result in [8] can be used to evaluate outage probability after doing a simple generalization to QAM constellations for transmission and quantization. We can then numerically evaluate the achievable rate. This can be done for both the link cooperation scheme as well as hybrid decoding.

Decode-Forward (DF)
In Phase 2, if the relay can decode from its Phase 1 reception, it re-encodes the decoded message and transmit it, so that coherent cooperation is attained. If the relay cannot decode, it keeps silent. The outage event can be evaluated as follows.

$$\begin{aligned}&\text{Outage} \iff \\ &\{R > \mathsf{C}_{\mathrm{P2P}}\left(h[1]\right)\} \cap \Big\{ \{R \leq \mathsf{C}_{\mathrm{P2P}}\left(h_r[1]\right) \text{ and } R > \mathsf{C}_{\mathrm{MISO}}\left(h[2], g[2]\right)\} \cup \\ &\{R > \mathsf{C}_{\mathrm{P2P}}\left(h_r[1]\right) \text{ and } R > \mathsf{C}_{\mathrm{P2P}}\left(h[2]\right)\} \Big\}\end{aligned} \tag{A.111}$$

Opportunistic-Decoding QIF (DQIF)
A natural way to combine QIF and DF relaying is the following: if the relay can decode, it performs DF as above; otherwise, it performs QIF instead of keeping silent. With this opportunistic scheme at the relay, the outage event can be evaluated as follows.

$$\begin{aligned}&\text{Outage} \iff \\ &\{R > \mathsf{C}_{\mathrm{P2P}}\left(h[1]\right)\} \cap \Big\{ \{R \leq \mathsf{C}_{\mathrm{P2P}}\left(h_r[1]\right) \text{ and } R > \mathsf{C}_{\mathrm{MISO}}\left(h[2], g[2]\right)\} \cup \\ &\Big\{R > \mathsf{C}_{\mathrm{P2P}}\left(h_r[1]\right) \text{ and } R > \mathsf{C}_{\mathrm{R}}^{\mathrm{QIF}}\left(h_r[1], h[2], g[2]\right)\Big\} \Big\}\end{aligned} \tag{A.112}$$

Bibliography

[1] International Telecommunication Union, "The World in 2014 - ICT Facts and Figures," *Available at http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf*, 2014.

[2] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[3] A. Chakrabarti, E. Erkip, A. Sabharwal, and B. Aazhang, "Code designs for cooperative communication," *IEEE Signal Processing Magazine*, vol. 24, pp. 16–26, Sept 2007.

[4] A. E. Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2012.

[5] "Local and metropolitan area networks–specific requirements part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," *IEEE Std 802.11-2012*, 2012.

[6] Y. Birk and T. Kol, "Informed-source coding-on-demand (ISCOD) over broadcast channels," in *Proc. of the IEEE INFOCOM*, pp. 1257–1264, 1998.

[7] S. El Rouayheb, A. Sprintson, and C. Georghiades, "On the relation between the index coding and the network coding problems," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 1823–1827, 2008.

[8] A. Avestimehr, S. Diggavi, and D. Tse, "Wireless network information flow: A deterministic approach," *IEEE Transactions on Information Theory*, vol. 57, pp. 1872–1905, April 2011.

[9] A. Ozgur and S. Diggavi, "Approximately achieving gaussian relay network capacity with lattice-based qmf codes," *IEEE Transactions on Information Theory*, vol. 59, pp. 8275–8294, Dec 2013.

[10] C. Nazaroglu, A. Ozgur, and C. Fragouli, "Wireless network simplification: the gaussian n-relay diamond network," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 2472–2476, 2011.

[11] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," in *Proc. of the IEEE Symposium on Foundations of Computer Science*, pp. 197–206, 2006.

[12] E. Lubetzky and U. Stav, "Non-linear index coding outperforming the linear optimum," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 161–168, 2007.

[13] N. Alon, A. Hassidim, E. Lubetzky, U. Stav, and A. Weinstein, "Broadcasting with side information," in *Proc. of the IEEE Symposium on Foundations of Computer Science*, pp. 823–832, 2008.

[14] M. Langberg and A. Sprintson, "On the hardness of approximating the network coding capacity," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 315–319, 2008.

[15] A. Blasiak, R. Kleinberg, and E. Lubetzsky, "Index coding via linear programming," *Available at http://arxiv.org/abs/1004.1379*, 2010.

[16] A. Blasiak, R. Kleinberg, and E. Lubetzsky, "Lexicographic products and the power of non-linear network coding," in *Proc. of the IEEE Symposium on Foundations of Computer Science*, pp. 609–618, 2011.

[17] H. Maleki, V. Cadambe, and S. Jafar, "Index coding: An interference alignment perspective," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 2236–2240, 2012.

[18] M. Chaudhry, Z. Asad, A. Sprintson, and M. Langberg, "On the complementary index coding problem," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 244–248, 2011.

[19] S. Dau, V. Skachek, and Y. Chee, "Optimal index codes with near-extreme rates," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 2241–2245, 2012.

[20] S. Dau, V. Skachek, and Y. Chee, "On secure index coding with side information," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 983–987, 2011.

[21] S. Dau, V. Skachek, and Y. Chee, "Index coding and error correction," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 1787–1791, 2011.

[22] Y. Berliner and M. Langberg, "Index coding with outerplanar side information," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 806–810, 2011.

[23] A. Tehrani, A. Dimakis, and M. Neely, "Bipartite index coding," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 2246–2250, 2012.

[24] M. Neely, A. Tehrani, and Z. Zhang, "Dynamic index coding for wireless broadcast networks," in *Proceedings of the IEEE INFOCOM*, pp. 316–324, 2012.

[25] L. Ong and C. K. Ho, "Optimal index codes for a class of multicast networks with receiver side information," in *Proc. of the IEEE International Conference on Communications*, pp. 2213–2218, June 2012.

[26] A. Le, A. Tehrani, A. Dimakis, and A. Markopoulou, "Instantly decodable network codes for real-time applications," in *Proc. of the IEEE International Symposium on Network Coding*, pp. 1–6, June 2013.

[27] T. J. Schaefer, "The complexity of satisfiability problems," in *Proc. of the ACM Symposium on Theory of Computing*, pp. 216–226, 1978.

[28] C. Fragouli and E. Soljanin, *Network Coding Fundamentals*, vol. 2 of *Foundations and Trends in Networking*. 2007.

[29] D. Dubhashi and A. Panconesi, *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2012.

[30] S. Brahma, A. Ozgur, and C. Fragouli, "Simple schedules for half-duplex networks," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 1112–1116, 2012.

[31] H. Bagheri, A. Motahari, and A. Khandani, "On the capacity of the half-duplex diamond channel," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 649–653, 2010.

[32] M. Cardone, D. Tuninetti, R. Knopp, and U. Salim, "Gaussian half-duplex relay networks: Improved constant gap and connections with the assignment problem," *IEEE Transactions on Information Theory*, vol. 60, no. 6, pp. 3559–3575, 2014.

[33] F. Parvaresh and R. H. Etkin, "Efficient capacity computation and power optimization for relay networks," *IEEE Transactions on Information Theory*, vol. 60, no. 3, pp. 1782–1792, 2014.

[34] C. Nazaroglu, J. Ebrahimi, A. Ozgur, and C. Fragouli, "Network simplification: The gaussian diamond network with multiple antennas," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 79–83, 2011.

[35] R. Tannious and A. Nosratinia, "Spectrally-efficient relay selection with limited feedback," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 1419–1428, October 2008.

[36] A. Bletsas, H. Shin, and M. Win, "Cooperative communications with outage-optimal opportunistic relaying," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 3450–3460, September 2007.

[37] J. Cai, X. Shen, J. W. Mark, and A. Alfa, "Semi-distributed user relaying algorithm for amplify-and-forward wireless relay networks," *IEEE Transactions on Wireless Communications*, vol. 7, pp. 1348–1357, April 2008.

[38] Y. Zhao, R. Adve, and T. J. Lim, "Improving amplify-and-forward relay networks: optimal power allocation versus selection," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 3114–3123, August 2007.

[39] S. Agnihotri, S. Jaggi, and M. Chen, "Analog network coding in general snr regime: Performance of network simplification," in *Proc. of the IEEE Information Theory Workshop*, pp. 632–636, Sept 2012.

[40] D. Luenberger and Y. Ye, *Linear and Nonlinear Programming*. Springer, 2008.

[41] M. Jorgovanovic, M. Weiner, D. Tse, B. Nikolic, I.-H. Wang, and V. Nagpal, "Relay scheduling and interference cancellation for quantize-map-and-forward cooperative relaying," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 1959–1963, July 2013.

[42] S. Brahma and C. Fragouli, "A simple relaying strategy for diamond networks," in *Proc. of the IEEE International Symposium on Information Theory*, pp. 1922–1926, 2014.

[43] G. Kramer, "Models and theory for relay channels with receive constraints," in *Proc. of the Allerton Conference on Communication, Control, and Computing*, pp. 1312–1321, 2004.

[44] S. Lim, Y.-H. Kim, A. El Gamal, and S.-Y. Chung, "Noisy network coding," *IEEE Transactions on Information Theory*, vol. 57, pp. 3132–3152, May 2011.

[45] J. B. Orlin, "A faster strongly polynomial time algorithm for submodular function minimization," *Mathematical Programming*, vol. 118, no. 2, pp. 237–251, 2009.

[46] S. Fujishige and S. Isotani, "A submodular function minimization algorithm based on the minimum-norm base," *Pacific Journal of Optimization*, vol. 7, pp. 3–17, 2011.

[47] "Nlopt nonlinear optimization library," in *Available at http://ab-initio.mit.edu/wiki/index.php/NLopt*.

[48] G. Kramer, I. Marić, and R. Yates, "Cooperative communications," *Foundations and Trends in Networking*, vol. 1, no. 3, pp. 271–425, 2006.

[49] M. Duarte, A. Sengupta, S. Brahma, C. Fragouli, and S. Diggavi, "Quantize-map-forward (QMF) relaying: An experimental study," in *Proceedings of the ACM MobiHoc*, pp. 227–236, July 2013.

[50] T. Korakis, M. Knox, E. Erkip, and S. Panwar, "Cooperative network implementation using open-source platforms," *IEEE Communications Magazine*, vol. 47, no. 2, pp. 134–141, 2009.

[51] G. Bradford and J. N. Laneman, "A survey of implementation efforts and experimental design for cooperative communications," in *Proceedings of the IEEE ICASSP*, pp. 5602–5605, 2010.

[52] G. Bradford and J. N. Laneman, "An experimental framework for the evaluation of cooperative diversity," in *Proceedings of the IEEE CISS*, pp. 641–645, March 2009.

[53] M. Knox and E. Erkip, "Implementation of cooperative communications using software defined radios," in *Proceedings of the IEEE ICASSP*, pp. 5618–5621, March 2010.

[54] P. Murphy, *Design, Implementation, and Characterization of a Cooperative Communications System*. PhD thesis, Rice University, 2010.

[55] H. Rahul, H. Hassanieh, and D. Katabi, "SourceSync: a distributed wireless architecture for exploiting sender diversity," in *Proceedings of the ACM SIGCOMM*, pp. 171–182, August 2010.

[56] X. Zhang and K. G. Shin, "DAC: Distributed asynchronous cooperation for wireless relay networks," in *Proceedings of the IEEE INFOCOM*, pp. 1064–1072, March 2010.

[57] V. Nagpal, I. Wang, M. Jorgovanovic, D. Tse, and B. Nikolic, "Coding and system design for quantize-map-and-forward relaying," *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 1423–1435, Aug 2013.

[58] A. Sengupta, S. Brahma, A. Ozgur, C. Fragouli, and S. Diggavi, "Graph-based codes for quantize-map-and-forward relaying," in *Proceedings of the IEEE Information Theory Workshop*, pp. 140–144, October 2011.

[59] E. Atsan, R. Knopp, S. Diggavi, and C. Fragouli, "Towards integrating quantize-map-forward relaying into LTE," in *Proceedings of the IEEE Information Theory Workshop*, pp. 212–216, September 2012.

[60] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, May 2005.

[61] P. Murphy and A. Sabharwal, "Design, implementation, and characterization of a cooperative communications system," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 2534–2544, July 2011.

[62] H. V. Balan, R. Rogalin, A. Michaloliakos, K. Psounis, and G. Caire, "AirSync: Enabling distributed multiuser MIMO with full spatial multiplexing," *IEEE/ACM Transactions on Networking*, no. 99, 2013.

Curriculum Vitae

Siddhartha Brahma

EPFL IC ARNI
Building INR, Station 14
1015 Lausanne, Switzerland
Email: sidbrahma@gmail.com
siddhartha.brahma@epfl.ch
Phone: +41-786-396-298

Education

2010–2014	**PhD in Computer, Communication and Information Sciences** École Polytechnique Fédérale de Lausanne (EPFL), Switzerland GPA: 5.72/6.00, Advisor: Prof. Christina Fragouli
2005–2008	**Masters in Computer Science** Princeton University, USA GPA: 3.67/4.00, Advisor: Prof. Robert E. Tarjan
2001–2005	**Bachelors in Computer Science and Engineering** Indian Institute of Technology (IIT), Kharagpur, India GPA: 9.71/10.00, Advisor: Prof. Sudebkumar P. Pal

Publications

1. S. Brahma and C. Fragouli, "On the complexity of scheduling in half-duplex diamond networks", *Submitted to IEEE Transactions on Information Theory.*

2. S. Brahma and C. Fragouli, "Pliable index coding", *Submitted to IEEE Transactions on Information Theory.*

3. S. Brahma, A. Sengupta and C. Fragouli, "Switched local schedules for diamond networks", *IEEE Information Theory Workshop*, 2014, Hobart, Australia.

4. S. Brahma and C. Fragouli, "Structure of optimal schedules in diamond networks", *IEEE International Symposium on Information Theory*, 2014, Honolulu, USA.

5. S. Brahma, A. Sengupta and C. Fragouli, "Efficient subnetwork selection in relay networks", *IEEE International Symposium on Information Theory*, 2014, Honolulu, USA.

6. S. Brahma and C. Fragouli, "A simple relaying strategy for diamond networks", *IEEE International Symposium on Information Theory*, 2014, Honolulu, USA.

7. S. Brahma, A. Sengupta, M. Duarte, C. Fragouli and S. Diggavi, "QUILT: A decode/quantize-interleave-transmit approach to cooperative relaying", *IEEE INFOCOM*, 2014, Toronto, Canada.

8. S. Brahma, C. Fragouli and A. Ozgur, "On the structure of approximately optimal schedules for half-duplex diamond networks", *Allerton Conference on Communication, Control, and Computing*, 2013, Allerton, USA.

9. M. Duarte, A. Sengupta, S. Brahma, C. Fragouli and S. Diggavi, "Quantize-map-forward (QMF) relaying: An experimental study", *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* , 2013, Bangalore, India. **(BEST PAPER AWARD)**

10. S. Brahma and C. Fragouli, "Pliable index coding: The multiple requests case", *IEEE International Symposium on Information Theory*, 2013, Istanbul, Turkey.

11. S. Brahma, A. Ozgur and C. Fragouli, "Simple schedules for half duplex networks", *IEEE International Symposium on Information Theory*, 2012, MIT, USA.

12. S. Brahma and C. Fragouli, "Pliable index coding", *IEEE International Symposium on Information Theory*, 2012, MIT, USA.

13. A. Sengupta, S. Brahma, A. Ozgur, C. Fragouli and S.Diggavi, "Graph-based codes for quantize-map-and-forward relaying", *IEEE Information Theory Workshop*, 2011, Paraty, Brazil.

14. A. Sarkar, A. Banerjee, N. Banerjee, S. Brahma, B. Kartikeyan, M. Chakraborty and K.L. Majumder, "Landcover classification in MRF context using Dempster-Shafer fusion for multisensor imagery", *IEEE Transactions on Image Processing*, vol 14(5), pp. 634–645, 2005.

15. S. Brahma, S.P. Pal and D. Sarkar, "A linear worst-case lower bound on the number of holes inside regions visible due to multiple diffuse reflections", *Journal of Geometry*, vol 81(1), pp. 5–14, 2004.

16. S. Brahma, P.H.D. Ramakrishna and S.P. Pal, "A new and novel approach for finding upper bounds on the distance of an approximate zero from an exact zero of a univariate polynomial", *International Journal of Computer Mathematics*, vol 81(12), pp. 1549–1557, 2004.

17. A. Sarkar, N. Banerjee, P. Nair, A. Banerjee, S. Brahma, B. Kartikeyan and K.L. Majumder, "A MRF based segmentation approach to classification using Dempster-Shafer fusion for multisensor imagery", *International Conference on Image Analysis and Recognition*, 2004, Porto, Portugal.

18. S. Brahma, S. Macharla, S.P. Pal and S.K. Singh, "Fair leader election through randomized voting", *International Conference on Distributed Computing and Internet Technology*, 2004, Bhubaneshwar, India.

Work Experience

2010 **Software Engineer at Google.com, India**
As a member of the Google MapMaker team, I contributed to the development of the next generation of mapping software that uses crowdsourcing to create online maps.

2008–2009 **Search Engineer at a startup - Guruji.com, India**
As a part of the core team of engineers, I designed, implemented and tested algorithms for several stages in the search engine pipeline. The role involved mining relevant information from huge amounts of noisy data like webpages and weblogs using techniques from advanced algorithms, machine learning, advanced data structures and natural language processing. Key contributions include duplicate detection, phonetic stemmer, link spam detection, template detection, news clustering and hostrank of the webgraph.

Projects

1. NOWIRE (Network Coding for Wireless Networks) - Part of my doctoral research contributed to the aims of this ERC funded project, where the goal was to develop fundamentally new techniques using network coding and wireless cooperation to enhance the throughput of networks.

2. CONECT (Cooperative Networking for High Capacity Transport Architectures) - Part of my doctoral research also contributed to this FP7 project, where the goal was to explore the possibilities of physical layer cooperation in wireless networks, drastically improving from current suboptimal strategies of strict signal separation and treating multiple signals as noise.

3. ISRO Project on Landcover Classification - During my Bachelors degree, I worked on an Indian Space Research Organisation (ISRO) project under the Grant for the Development of Landcover Classification Methodology with Fusion of Data from Different Sensors (Ref. 10/4/416).