Title
Precoding-Based Techniques for Multiple Unicasts in Wired and Wireless Networks

Author
Ramakrishnan, Abinesh

Publication Date
2015

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

Precoding-Based Techniques for Multiple Unicasts in Wired and Wireless Networks

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Abinesh Ramakrishnan

Dissertation Committee:
Professor Athina Markopoulou, Chair
Professor Syed Ali Jafar
Professor Animashree Anandkumar

2015

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

I would like to take this opportunity to acknowledge those who made this thesis possible.

First and foremost, I owe my deepest gratitude to my advisor, Prof. Athina Markopoulou. Her support, patience and guidance has been invaluable during the course of my academic career. She gave me the opportunity to join her group and introduced me to the field of network coding. She has greatly contributed to my intellectual maturity and the achievement of my academic goals. The directions I followed in this thesis would not have been possible without her help.

I would like to thank my committee members Prof. Syed Ali Jafar and Prof. Animashree Anandkumar for evaluating my thesis and helping me improve my research work. I am especially grateful to Prof. Syed Jafar for introducing me to the concept of interference alignment, getting me interested in the area of information theory and co-advising me on two of the chapter on my thesis. His structured way of thinking and insightful comments helped shape the research directions I followed during the course of graduate studies.

I am extremely grateful to Dr. Cedric Westphal for supervising me as an intern at at Huawei US R&D Center and giving me the opportunity to work on a challenging real-world problem (coded caching), which eventually became a part of my thesis. I would like to thank my peers and fellow graduate students: Chun Meng, Sundar Rajan Krishnamurthy, Hamid Maleki, and Abhik Kumar Das for collaborating with me on the research projects I was involved during my studies. I am grateful to Prof. Filippo Capolino for providing me the opportunity to work as a teaching assistant for his class during the early my early years in graduate school.

This work would not have been possible without the support of my family and friends. I would like to thank my parents and my sister for their emotional and moral support. I would like to thank all my friends at UC Irvine for making this journey a fun filled one. Siva, Minas, Blerim, Stelios and Rohan for the important moments we shared, the extensive discussions we had and the fun and daring activities we did. I would also like to thank all the past and present member of my lab: Hulya Seferoglu, Minas Gjoka, Fabio Soldo, Maciej Kurant, Pegah Sattari, Anh Le, Chun Meng and Blerim Cici. I have learned a lot from them through group meetings and insightful discussions.

Finally, I would like to thank the Department of Electrical Engineering and Computer Science and the Center for Pervasive Communications and Computing at UC Irvine for supporting my work. I would also like to acknowledge the support by NSF (0747110, 1028394) and AFOSR (FA9550-09-0643).

CURRICULUM VITAE

Abinesh Ramakrishnan

EDUCATION

Doctor of Philosophy in Electrical Engineering **2015**
University of California, Irvine *Irvine, California*

Master of Science in Electrical Engineering **2010**
University of California, Irvine *Irvine, California*

Bachelor of Engineering in Electronics and Communications **2008**
Madras Institute of Technology, Anna University *Chennai, India*

RESEARCH EXPERIENCE

Graduate Research Assistant **2009–2014**
University of California, Irvine *Irvine, California*

Research Intern **2013**
Huawei US Research & Development *Santa Clara, California*

TEACHING EXPERIENCE

Teaching Assistant **2009–2010**
University of California, Irvine *Irvine, California*

ABSTRACT OF THE DISSERTATION

Precoding-Based Techniques for Multiple Unicasts in Wired and Wireless Networks

By

Abinesh Ramakrishnan

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2015

Professor Athina Markopoulou, Chair

In this thesis, we study intersession coding for multiple unicasts in wired and wireless network settings. In particular, we apply alignment techniques and investigate the effect of structure of the transfer matrix to their performance. In addition, we also look at the coded caching problem and we propose an efficient delivery scheme that outperforms state-of-the-art.

The thesis is divided into three parts. In the first part, we consider the problem of network coding across three unicast sessions over a directed acyclic graph, where each unicast session has a min-cut of 1. We consider a network model in which the middle of the network can only perform random linear network coding. We adapt interference alignment technique, originally developed for the wireless interference channel, to construct a precoding-based linear scheme, which we refer to as precoding-based network alignment (PBNA). The primary difference between this setting and the wireless interference channel is that the network topology can introduce dependencies among the elements of the transfer matrix and can potentially affect the achievable rate of PBNA. We identify all these dependencies and we interpret them in terms of network topology. We also show that, depending on these network topologies, the optimal symmetric rate achieved by any precoding-based linear scheme can take only three possible values, all of which can be achieved by PBNA.

In the second part, we consider the interference channel with K transmitters and K receivers

all having a single antenna, wherein the $K \times K$ transfer matrix representing this channel has rank D (less than K). The degrees of freedom of such channels are not known as the rank-deficient transfer matrix creates algebraic dependencies between the channel coefficients. We present a modified version of the alignment scheme, to handle these dependencies while aligning interference, and derive the sufficient conditions for achieving half rate per user using this scheme. We show the difficulties in proving these sufficient condition for $K = 4$ and $K = 5$ and we also show that these sufficient conditions are not satisfied for $K \geq 6$.

Finally, we study the coded caching problem: a network with several users trying to access a database of files stored at a server through a shared bottleneck link is considered. Each user is equipped with a cache, where files can be prefetched according to a caching policy, which is mainly based on the popularities of the files. Coded caching tries to exploit coding opportunities created by cooperative caching and has been shown to significantly reduce the load on the shared link. Most prior work focused on optimizing the caching policy so as to minimize this expected load. Given the caching policy and the user demands, the problem of minimizing the load over the shared link is essentially an index coding problem. In this part of the thesis, we design a novel delivery scheme, Heterogeneous Coded Delivery (HCD), that builds on a prior scheme for the uniform demand case, but performs better in the non-uniform demand case. We evaluate this delivery scheme for different caching policies.

Chapter 1

Introduction

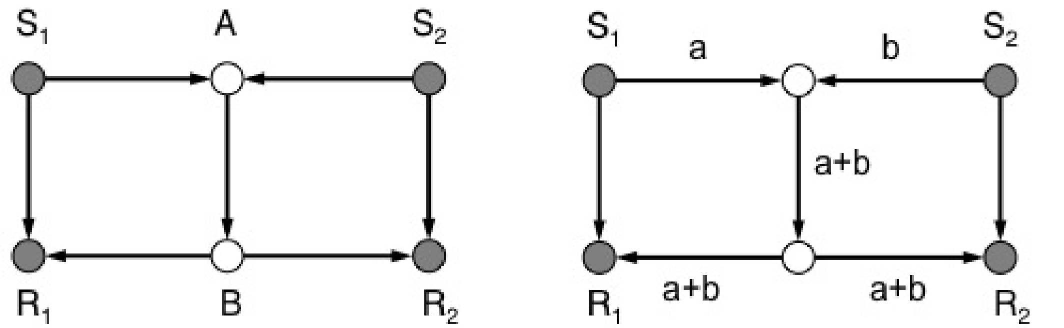
1.1 Network Coding

Today's networks are all based on the fundamental principle that the network forwards data but the information itself is processed only at the end-nodes. Network coding was introduced over a decade ago and promised to revolutionize the way we think and operate networks. The main idea of network coding was that if intermediate nodes are allowed not only to forward, but also to process and combine packets, then we can increase throughput by allowing each flow to use the full capacity of shared links [1] and facilitate distributed operation of networks via random network coding [2]. The concept of network coding builds upon the fundamental observation that even in error-free communication networks, compared to routing only approaches, applying coding operations at intermediate nodes as well as at the terminal nodes introduces extra flexibilities and new possibilities, which leads to new understandings in achieving low-cost, high-rate, or fault-tolerant data transmission. Maximizing information exchange over communication networks has been a major subject among both the information theory and the networking communities, and this breakthrough idea generated interest in these communities, especially since it showed promise to revolutionize the way we think and operate networks.

Despite the promise and significant amount of research activity generated over the last decade, network coding has not yet realized its full potential. For example, there have been relatively few practical implementations of network coded systems today, the best known of which are in peer-to-peer networks [3, 4] and COPE [5] in wireless multihop networks. Given that most Internet traffic today is carried over unicast connections, understanding how to perform network coding across multiple unicasts is crucial for making the case for (or against) the deployment of network coding.

Multi-commodity Flows and the Benefit of Network Coding. Without network coding, the multiple unicast routing problem becomes the multi-commodity flow problem (MCF). The fractional multi-commodity flow problem can be solved using linear optimization. But the integer MCF is a generalization of the edge disjoint path problem. The goal in the edge disjoint path problem is to connect as many pairs (s_i, t_i) as possible using non edge intersecting paths, which is closely related to the integer MCF problem. The edge-disjoint path problem is a well-known NP-hard problem in both directed and undirected settings. The concepts of integral routing and fractional routing are analogous to the coding and transfer of scalars and vectors respectively. In directed networks, the throughput improvement due to network coding is unbounded, and may increase at the speed of $\Theta(|V|)$ as the network size grows [6, 7].

Fig. 1.1(a) shows the famous "butterfly" example, where two unicast sessions are to be set up in a directed network, with throughput requirement $(1, 1)$, this is widely known as the butterfly network. Fig. 1.1(b) shows a coded transmission scheme that realizes both sessions without violating link capacity bounds or link directions. With routing only, the rate vector $(1, 1)$ is infeasible. Note that there is only one path connecting S_1 to R_1, $S_1 \to A \to B \to R_1$, and there is only one path connecting S_2 to R_2, $S_2 \to A \to B \to R_2$. These two paths share



(a) Two unicast sessions in a directed network, each with throughput requirement 1.

(b) A coded transmission scheme that realizes these two unicast sessions

Figure 1.1: Butterfly with unicasts [8].

the unit capacity link $A \to B$, which becomes a bottleneck. The example in Fig. 1.1 contains two unicast sessions, and shows a coding advantage of 2.

Networks with larger number of sessions can be constructed according to a similar pattern, such that larger coding advantages are exhibited [6,9]. A pattern of directed multiple unicasts that favors network coding. Each S_i is connected to A and all receivers except R_i; each R_i is connected from B and all senders except S_i. As shown in Fig. 1.2, a directed network of the pattern has $2n$ unicast terminals, and two relay nodes A and B. Every sender is connected to A, every receiver is connected from B, and A is connected to B. Furthermore, every sender S_i is connected to all receivers except the matching one, R_i (which implies every receiver R_i is connected from all senders except the matching one, S_i). With network coding, each receiver can send one bit information to A, at which point it is encoded with bits from other senders, and further relayed to B and then each receiver. Besides, every sender will also send the same information to the $n-1$ receivers it is directly connected to. As a result, every receiver R_i is able to recover the 1 bit information from S_i. Without network coding, the total throughput of all the k sessions is bounded by 1 bit. This can be verified by the fact that, removing the single link from A to B disconnects every sender-receiver pair in the network. In networks conforming to this pattern, network size $|V|$ is $2n+2$, total throughput

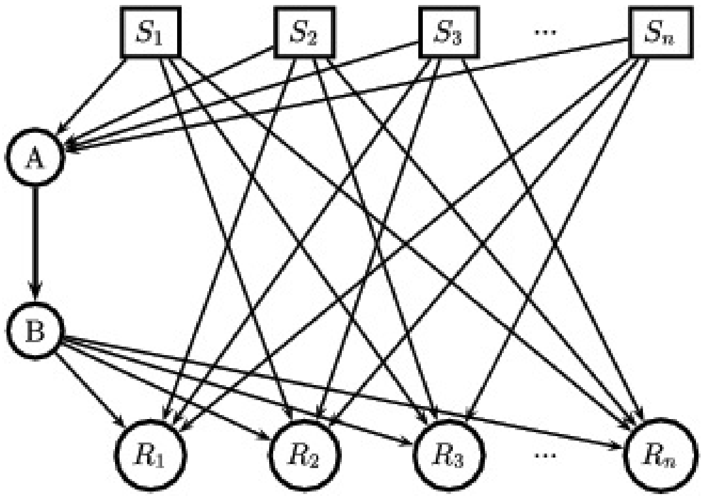


Figure 1.2: A pattern of directed multiple unicasts that favors network coding. Each S_i is connected to A and all receivers except R_i; each R_i is connected from B and all senders except R_i [6, 9].

with coding is n, and total throughput with routing only is bounded by 1. This shows that the coding advantage grows proportionally as $\Theta(|V|)$ in this case, with either integral or fractional routing [10].

The adoption of network coding ideas into practical networked systems has been hindered, so far, due to the fundamental challenge: Network Coding across different sessions is hard. In the single session case, the achievable coded throughput can be nicely characterized. A throughput demand r is feasible if and only if a directed flow of rate r can be set up from the sender to each receiver. Intra-session network coding (*i.e.,* coding within a single unicast or multicast session) is well-understood today: the achievable rates are known (*i.e.,* the minimum source-receiver min-cut) and coding schemes (deterministic or random) have been designed to achieve those rates. In contrast, finding the optimal network coding strategy in the inter-session scenario (*i.e.,* where there are more than one sessions, and receivers are interested in different sets of information) is arguably the most important open research problem in the network coding community.

1.2 Interference Alignment

The recent emergence of the idea of interference alignment for wireless networks has shown that the throughput potential of wireless networks can be orders of magnitude higher than previously believed [11]. The canonical example of interference alignment is a communication scenario where, regardless of the number of interferers, every user is able to access one half of the spectrum free from interference from other users. The key to interference alignment is the realization that the alignment of signal spaces (in time, frequency, space and codes) is relative to the observer (receiver). Since every receiver sees a different picture, signals may be constructed to cast overlapping shadows at the receivers where they constitute interference while they remain distinguishable at the receivers where they are desired. This work has generated a large amount of follow-up work and has revolutionized the information theoretic study of wireless networks.

The basic intuition behind the idea of interference alignment is illustrated in Fig. 1.3 in the context of the K user interference channel. Multiple users (transmitter-receiver pairs of the same color) wish to communicate over the same wireless channel. With vector representation for signals, the bandwidth is simply the number of signaling dimensions. In the figure we have a total of two dimensions, i.e., a normalized bandwidth of 2. Thus, each user would be able to access two signaling dimensions if he had the channel all to himself. With K competing users, conventional medium-access schemes such as TDMA/FDMA/CDMA would share the bandwidth among the users so that each user would have access to $2/K$ signaling dimensions. However, with interference alignment each user is able to access 1 signaling dimension in the null space of the interference, thus resulting in an improvement in the overall available bandwidth by a factor of $K/2$. This is possible because all interference is aligned into a one-dimensional subspace while the desired signal at each receiver "stands apart" from the interference.

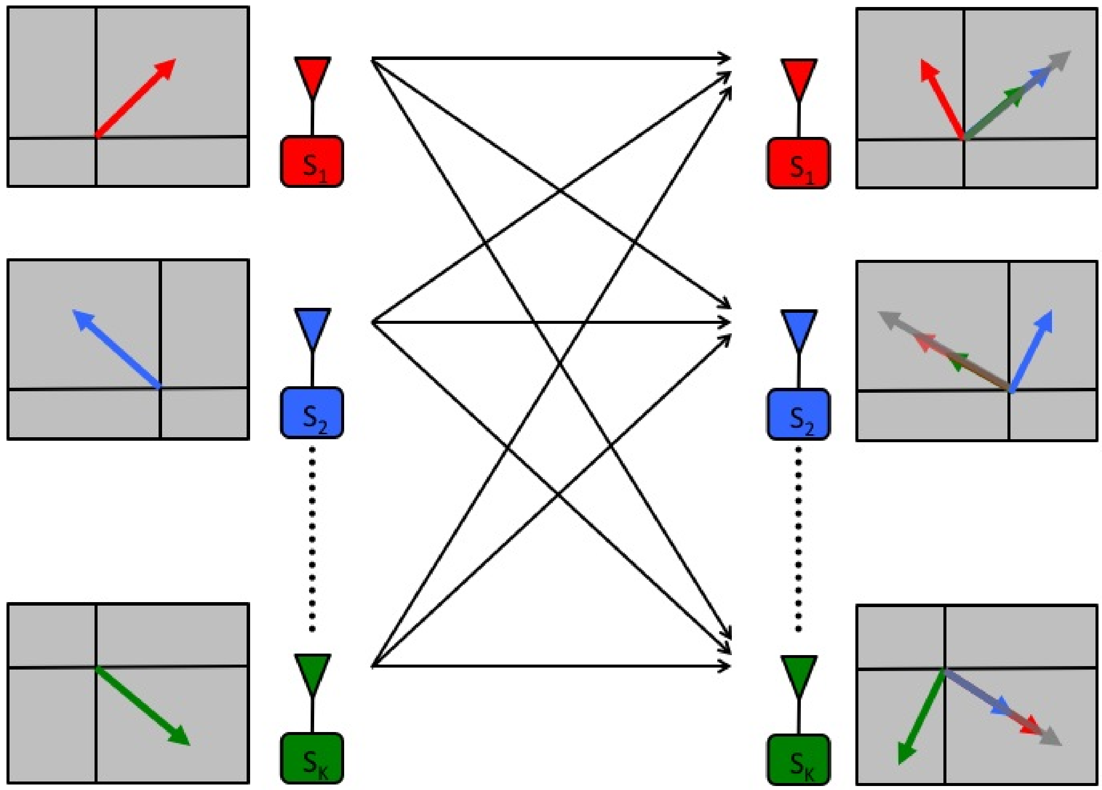
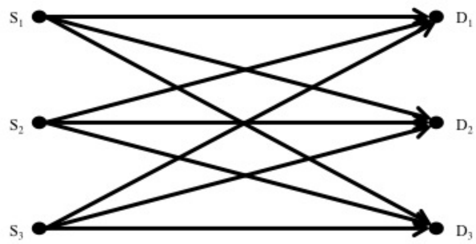


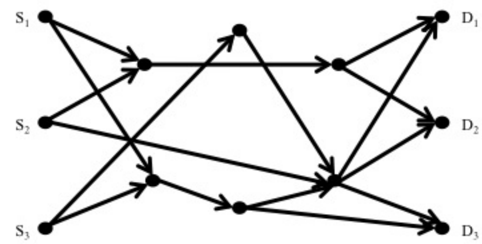
Figure 1.3: Interference Alignment Concept in the K-User Wireless Interference Channel. In a two dimensional signal vector space, all undesired signals at each receiver align within one dimension leaving the other dimension interference free for the desired signal which is not aligned with the interference [11].

Network Alignment (NA). In this dissertation, we apply interference alignment techniques (originally developed for the wireless interference channel) to network coding across different sessions (with emphasis on unicast) over directed acyclic graphs. We refer to this new approach that combines network coding and interference alignment as *Network Alignment (NA)*.

For example, consider several unicast sessions over a wireline network; this can be thought of as equivalent to supporting the same unicast sessions over a wireless interference channel with the same transfer function. This analogy is depicted in Fig.1.4. Multicast at intermediate nodes of a graph emulates broadcast in the wireless channel. Network coding across sessions at intermediate nodes emulates superposition in the wireless channel. Essentially, the entire



(a) Interference channel



(b) Multiple unicast network

Figure 1.4: Analogy between a wireless interference channel and a graph with network coding. 3 unicast sessions are established on top of either network. Both systems can be represented by a linear transfer function and are amenable to alignment techniques.

graph can be viewed as a channel, albeit a channel that is not given by nature, as it is the case in wireless, but determined by our routing and coding decisions. This has the advantage that it allows us to control the channel; however it also has the disadvantage that it introduces spatial-correlation between end-to-end paths, which is not the case, with high probability in wireless channels. Inter-session network coding across unicast sessions introduces interference that prevents receivers from decoding the packets of the session they are really interested in. Interference alignment techniques can then be applied to guarantee (at least) half the rate of each session, for any number of sessions transmitted over that network, subject to feasibility conditions. However, traditional IA techniques cannot be applied directly. The main novel challenge faced by alignment over networks is the aforementioned spatial correlation between end-to-end paths which with high probability is not present in wireless interference channels; this correlation stems from the properties of the graph structure and affects the feasibility and optimality of alignment techniques. Another difference is the operations over finite fields rather than real numbers.

1.3 Related Work

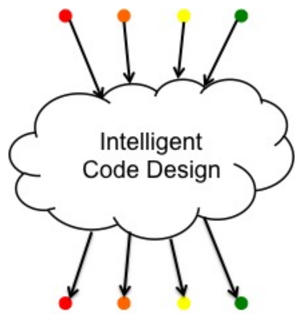
Intra-session network coding. Network coding was introduced a decade ago in the seminal paper [1] which characterized the achievable rates for multicast communication over graphs. It was followed soon after by papers that developed linear [12], algebraic [2] and other frameworks. The original work spurred a significant amount of research activity, some of which is summarized in [13, 14]. Today, intra-session network coding (for multicast or unicast) is fully understood: the throughput limits as well as efficient, deterministic and randomized, algorithms are known.

Inter-session network coding. There has been hope that the framework can be extended to solve a wider array of network capacity problems, namely inter-session network coding, which includes the practical case of multiple unicasts. inter-session network coding has demonstrated significant throughput advantage over routing algorithms [1, 6, 10, 12]. A sufficient condition for optimality of linear inter-session network coding was developed in [2]. However, scalar or even vector linear network coding [15–17] alone has been shown to be insufficient for optimal inter-session network coding [18], which includes the multiple unicast setup. In comparison to the single session scenario, the capacity region of multi-session network coding [10] has been less understood, and only for very special graphs can we characterize the capacity region, such as directed cycles [7], degree-2 three-layer directed acyclic networks [19], and special bipartite undirected graphs [7]. Even the inner/outer bounds of the rate region cannot be computed in practice [20, 21]. In short, the problem of a unifying characterization of inter-session network coding remains largely open. Only heuristic/suboptimal approaches are considered in practice, often without even performance guarantees. The lack of progress on intersession network coding is mainly attributed to its intrinsic hardness. For example, checking the existence of intersession network coding solutions is an NP-complete problem [22], its information-theoretic characterization is tightly intertwined with the fundamental regions of the entropy function [20], and linear coding is insufficient to achieve the intersession coding capacity [18]. Even when restricting our focus to only linear codes on directed acyclic networks, it is shown [23], [24] that the feasibility of linear intersession network coding is alphabet GF(q) dependent, and the complexity of determining the feasibility of linear intersession network coding is similar to that of the long standing problems of finding solutions of multi-variate polynomials. In the algebraic framework [2]: solving an intra-session linear network coding problem reduces to finding an assignment of coefficients that makes a polynomial non-zero; solving the inter-session linear network coding problem reduces to finding coefficients that make the diagonal terms of the transfer function (*i.e.,* direct paths m_{ii}) non-zero and the off-diagonal terms m_{ij} (*i.e.,* the interfering paths) zero. The latter results in solving a system of polynomial equations, with exponential number of variables [24]. With a fixed number of N coexisting sessions and a fixed alphabet size GF(q), finding a network coding solution is a polynomial time task with respect to the network size [25]. However, the complexity grows exponentially with respect to N and $b = log_2(q)$, the number of bits representing each alphabet. A good survey on the hardness of inter-session network coding can be found in [26–28]. To better understand the capacity of intersession network coding, many ongoing works focus on more tractable outer/inner bounds analysis for networks of general topology.

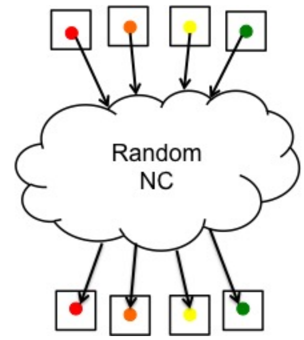
Capacity outer bounds have been proposed based on the generalized edge cut condition of the underlying graph and the associated information-theoretic arguments, including fundamental regions in the entropy space [20], entropy calculus [29], the network-sharing bound [19], the information dominance condition [7], and the edge-cut bounds [30]. The achievability results, i.e., the inner bound of the capacity region, is generally determined by linear programming in a similar fashion to that of solving fractional multi-commodity flow, including the butterfly-based construction, where the original network is decomposed into many butterfly substructures [8] and then linear programming is used for finding the corresponding network resource allocation [31], and the pollution-treatment with powerset-based flow division [32]. The capacity region of a special conservative requirement is studied in [33]. Another type of approach classifies the coded traffic by the participating sessions. The traffic flow that is a mixture of N unicast sessions can then be treated as generated from a single multicast session with N symbols, to which one can apply the min-cut/max-flow (mcMF) theorem [34], [32]. Recently, a new characterization has been discovered for pairwise inter-session network coding, which mixes only two symbols of two coexisting unicast/multicast sessions [28]. When combined with the superposition principle, the pairwise coding results are used to derive new achievable rates for N coexisting sessions [35]. A nice survey of the prior works on the inner and outer bounds of the capacity region can be found in [26, 28].

To make the problem tractable, some heuristic approaches consider only very restrictive classes of codes, either in terms of coding operations allowed (*e.g.,* XOR operations) and/or in terms of coding subgraph (*e.g.,* all packets are coded/decoded hop-by-hop in [5]; butterfly structures consider two hops [36]; tiling of patterns is considered in [37]). The most restrictive case is XOR coding between pairs of flows: this allows for a network flow formulation of the problem [8] known as butterfly packing, but was shown to bring only moderate improvement compared to traditional routing, depending on the network topology.

Interference alignment in wireless networks. Interference alignment was first introduced in [38] as a coding technique for the two-user Multiple-Input MultipleOutput (MIMO) X channel. Subsequently, the same principle was applied to the interference channel in [11]. In particular [11] showed interference alignment achieves the maximum degrees of freedom of the interference channel. Currently, IA schemes have been found for a variety of networks, these schemes include linear alignment schemes over signal spaces [38], asymptotic alignment schemes over a large number of dimensions [11], asymmetric complex signaling schemes [39], interference alignment and cancelation schemes [40] and ergodic alignment schemes [41]. [42] has a nice review of the fundamental concepts of interference alignment presented in [11].



(a) Constructive and systematic approach: Intelligent design of codes at the intermediate nodes to achieve desired rate.



(b) Distributed and random approach : Coding coefficient takes random values and the sources are responsible for coding their symbols to achieve desired rate.

Figure 1.5: Network Coding Approaches.

1.4 Motivation

The process of mixing information in the intermediate nodes of a network, as shown in Fig. 1.5, can be done in two ways: (*i*) Centralized and systematic approach, where we design the coding coefficient at the node by carefully selecting the subsets of the flow to combine and the locations in the graph where coding should be performed; and (*ii*) Distributed and random approach, where the coding coefficient take random values.

The first approach is hard because of the combinatorial nature of the problem: one needs to enumerate and select from all possible subsets of the flows that can be the combined together (code construction) and from all possible locations in the graph where coding/decoding should be performed (coding subgraph); unlike the intra-session case, these two problems also need to be solved jointly. An alternative way to think about the hardness of the problem is in the algebraic framework [2]: solving an intra-session linear network coding problem reduces to finding an assignment of coefficients that makes a polynomial non-zero; solving the inter-session linear network coding problem reduces to finding coefficients that make the diagonal terms of the transfer function (*i.e.,* direct paths m_{ii}) non-zero and the off-diagonal terms m_{ij} (*i.e.,* the interfering paths) zero. The latter results in solving a system of polynomial equations, with exponential number of variables [24].

The second approach of using random coding coefficients is easily implementable and is also useful in scenarios where we don't have intelligence in the intermediate nodes of the network. However, the achievable rates are restricted and the schemes for achieving these rates have to be applied at the source or sink nodes. This is the primary motivation for looking into precoding-based techniques. Network structure plays a central role in determining the rate region for both the approaches discussed above as it directly affects the way the flows are combined together. So it is important to understand how the network structure affects the achievable rates especially in the precoding based approach.

1.5 Outline

The rest of the thesis is organized into three chapters. In Chapter 2, we start by looking at the simple, yet non-trivial case of three unicast sessions ($K = 3$) in a network with min-cut of 1 per session. In this chapter, we apply an interference alignment techniques called *Precoding-Based Network Alignment (PBNA)* and present the minimal set of necessary and sufficient conditions for the feasibility of the alignment scheme. We will also present a graph theoretic characterization of these conditions and derive the optimal rates achievable by any precoding based linear scheme. We will also show that these optimal rates are indeed achievable using *PBNA* scheme (with slight modifications in some cases).

After thoroughly analyzing and successfully characterizing the precoding-based linear rates for the networks with three session, we move on to more than three sessions ($K > 3$). Alignment techniques becomes more complex as we increase the number of session. In Chapter 3, we focus on certain sub-classes of network topologies, namely we look at wireless interference channels with a *rank deficient transfer matrix*. We present a modified alignment scheme to deal with the algebraic dependencies introduced by the rank deficiency and analyze the challenges faced in characterizing the rate regions for these networks as we increase the number of sessions.

In Chapter 4, we look at the content distribution networks with caches at the end user. The side information provided by the content of the cache induces index coding instances, which is a very interesting multiple unicast problem with more than three sessions. We propose an efficient delivery scheme called the *heterogeneous Coded Caching*, *HCD* for short and evaluate the performance of this scheme.

Chapter 2

Network Alignment

2.1 Introduction

In this chapter, we are interested in developing a systematic alignment approach for network coding across multiple unicasts. We are also interested in understanding how the feasibility and performance vary based on on the network structure. We focus on a useful special case: network coding for three unicast sessions, which is the smallest, yet non-trivial, instance of the problem and which can be used as a building block for network coding across multiple unicasts. The two unicast session case has already been analyzed extensively and we have approaches and algorithms which can take full advantage of this case. Also, the concept of interference alignment would become pointless when the number of users are less than 2. So we focus on three unicast session case with hopes that a thorough understanding of this case would lead to the development of a more general method that could be extended to more the three session cases. We consider a network model, in which the middle of the network only performs random linear network coding, and restrict our approaches to *precoding-based linear schemes*, where the senders use precoding matrices to encode source symbols. We start by taking a look at two widely used approaches to achieve interference alignment and we show how a particular closed form method (Eigenvector method) of the precoding-based approach can be applied to the our case of wired network with three unicast sessions employing inter-session network coding. More importantly, we also analyze the main precoding-based method which uses asymptotic symbol extension. We derive the condition for the feasibility of alignment in both of above mentioned methods and try to provide a better understanding of their relation to the network structure. We analyze the three session network in further depth and show that even though alignment techniques may provide a systematic way to achieve half the min-cut, they are not strictly beneficial, as alternative approaches from current literature can also achieve the same rate. However, this holds only for the unicast network with three sessions. Towards the end of this chapter we present examples with more than three sessions or more than unit min-cut per session, where it is necessary to employ alignment strategies to achieve better rates.

2.2 Problem Formulation

2.2.1 Notation

The following notations are used as a standard throughout this text. The bold faced notation $\mathbf{A}$ is used typically to represent a matrix, we use $span(\mathbf{A})$ to denote the span of its columns and $rank(\mathbf{A})$ to denote its rank. $\mathbb{F}_p$ is used to denote the finite fied $0, 1, \ldots, p-1$, where p is a prime /prime power. $G = (V, E)$, represents a finite, directed, acyclic graph, where V is the set of nodes and E is the set of directed links (edges). An edge $e = (u, v) \in E$ is an ordered pair of nodes, for which u and v are termed as the tail and the head of e respectively. A path P from node u to v is defined as an ordered set of edges $\{(u, w_1), (w_1, w_2), \ldots, (w_n, v)\} \subseteq E$ such that the head of the previous edge. $\mathcal{P}$ represents a collection of some paths $P^{(n)}$'s. The superscript n implies the n^{th} path.

2.2.2 System Model

Consider a network represented by a finite, directed acyclic graph $G = (V, E)$. We assume that every directed link between a pair of nodes represents an error-free channel, and that the transmissions across different links do not interfere with each other in any way. There are K source nodes, $S_1, S_2, \ldots, S_3$, and K destination nodes, $D_1, D_2, \ldots, D_3$ and each S_i communicates only with D_i. The messages transmitted by different sources are assumed to be independent of each other. These messages are encoded and transmitted in form of symbols from finite field $\mathbb{F}_p$. For the sake of simplicity, we assume that every link in E has a capacity of one symbol (from $\mathbb{F}_p$) per channel use.

We use linear network coding at every node in G. The coefficients for linear combination of symbols at each node come from $\mathbb{F}_p$. We consider these coefficients to be variables, say $\{\xi_1, \xi_2, \ldots, \xi_s\}$ (s is a parameter dependent on the network topology), and define the vector $\underline{\xi} \triangleq [\xi_1 \quad \xi_2 \quad \cdots \quad \xi_s]$. A network coding scheme refers to choosing a suitable assignment for $\underline{\xi}$, from $\mathbb{F}_p^s$.

Suppose the mincut between S_i and D_i is $c_i \in \mathbb{N}$. Let the channel uses be indexed as $t = 1, 2, \ldots$. Then

$$\mathbf{y}_i(t) = \sum_{j=1}^{3} \mathbf{M}_{ij}(\underline{\xi})\mathbf{x}_j(t), \quad i = 1, 2, \ldots, K \tag{2.1}$$

where $\mathbf{x}_i(t) \in \mathbb{F}_p^{c_i \times 1}$ is the input vector at S_i during the t^{th} channel use, $\mathbf{y}_i(t)$ is the $c_i \times 1$ output vector at D_i during the t^{th} channel use and $\mathbf{M}_{ij}(\underline{\xi})$ is the $c_i \times c_j$ transfer matrix between S_j and D_i. Note that the entries of $\mathbf{y}_i(t)$ and $\mathbf{M}_{ij}(\underline{\xi})$ are multivariate polynomials from the polynomial ring $\mathbb{F}_p[\underline{\xi}]$ for all i, j. Since D_i needs to decode only $\mathbf{x}_i(t)$ from $\mathbf{y}_i(t)$, the presence of transfer matrices $\mathbf{M}_{ij}(\underline{\xi})$, $i \neq j$, hinders the decodability (act as "interference") at every destination. We refer to these as "interference transfer matrices".

By the Max-flow-min-cut Theorem, S_i can transmit at most c_i symbols to D_i per channel use (here channel use refers to usage of one assignment of $\underline{\xi}$ from $\mathbb{F}_p^s$). The generalized Max-flow-min-cut Theorem, studied in [2], states that multiple unicast connections in G can achieve a maximum throughout of c_i for every source-destination pair (S_i, D_i), iff there exists an assignment of $\underline{\xi}$ in $\mathbb{F}_p^s$, say $\underline{\xi}_0$, such that $\mathbf{M}_{ij}(\underline{\xi}_0) = 0$ for $i \neq j$ and $\mathbf{M}_{ii}(\underline{\xi}_0)$ is a full-rank matrix. However, there exists a broad class of networks for which such an assignment of $\underline{\xi}$ does not exist, thereby making multiple unicast at maximum throughput infeasible.

2.3 Three Session Unicast With Unit Min-Cut

In this work, we focus on a special case of the above model, namely a three-user multiple unicast network ($K = 3$) with unit mincut ($c_i = 1$) for every source-destination pair, *i.e.,* network G has 3 source nodes S_1, S_2, S_3 and 3 destination nodes D_1, D_2, D_3 and the three input-output relations in (2.1) can be rewritten as:

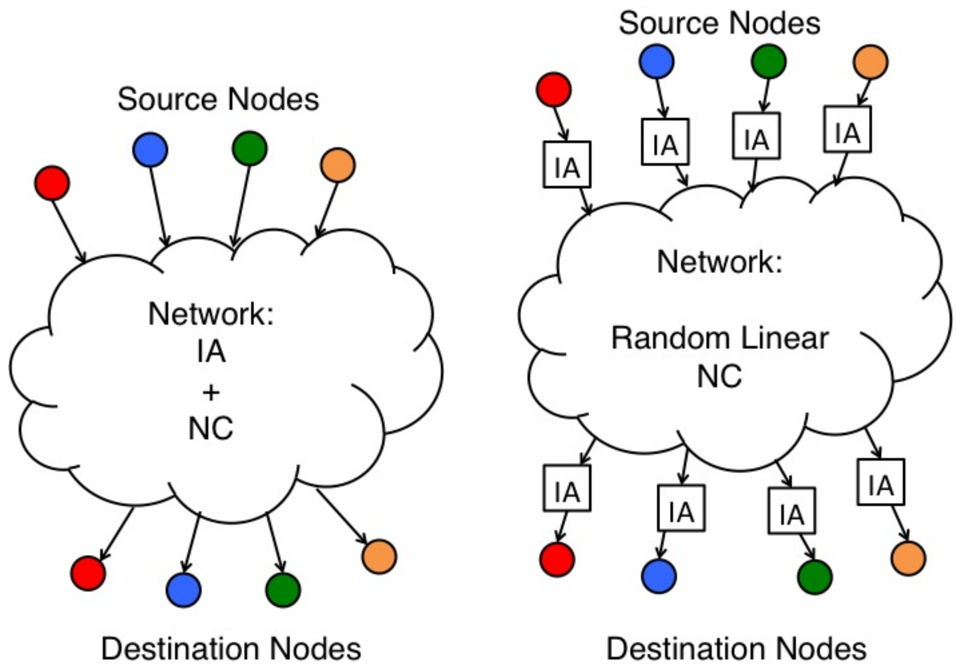
$$\begin{aligned}
y_1(t) &= m_{11}(\underline{\xi})x_1(t) + m_{12}(\underline{\xi})x_2(t) + m_{13}(\underline{\xi})x_3(t), \\
y_2(t) &= m_{21}(\underline{\xi})x_1(t) + m_{22}(\underline{\xi})x_2(t) + m_{23}(\underline{\xi})x_3(t), \\
y_3(t) &= m_{31}(\underline{\xi})x_1(t) + m_{32}(\underline{\xi})x_2(t) + m_{33}(\underline{\xi})x_3(t),
\end{aligned}$$

where $x_i(t), y_i(t)$ and $m_{ij}(\underline{\xi})$ are the scalar equivalents of $\mathbf{x}_i(t), \mathbf{y}_i(t)$ and $\mathbf{M}_{ij}(\underline{\xi})$ respectively. Moreover, we have $x_i(t) \in \mathbb{F}_p$ and $y_i(t), m_{ij}(t) \in \mathbb{F}_p[\underline{\xi}]$ for $i, j = 1, 2, 3$. Note that $m_{ii}(\underline{\xi})$, $i = 1, 2, 3$, are non-trivial polynomials. Also by construction, $m_{ii}(\underline{\xi})$ cannot be a non-zero constant and $m_{ii}(\underline{\xi}) \not\equiv c m_{ij}(\underline{\xi})$, where $c \in \mathbb{F}_p$. Thus, these non-trivial polynomials are exclusive functions of $\underline{\xi}$. We refer to $m_{ij}(\xi)$, $i, j = 1, 2, 3$, as *channel transfer functions.* We also refer to polynomials $m_{ii}(\underline{\xi})$, $i = 1, 2, 3$, as "direct paths" and polynomials $m_{ij}(\underline{\xi})$, $i \neq j$, as *interference paths.*

2.4 Network Alignment Approaches

Various approaches have been pursued to achieve interference alignment in wireless networks. In particular, within the linear framework, two distinct approaches have been used extensively. The first is the asymptotic IA (also known as "symbol-extension") approach, originally introduced in [11], where IA is achieved in the limit of large number of signaling dimensions. The second approach known as ergodic IA, first introduced in [41], has an opportunistic flavor. While the asymptotic approach is known to be much more widely applicable, the ergodic IA approach is also applicable to a fairly broad class of commonly studied wireless networks and is in general much more efficient – in many cases achieving theoretical capacity region as shown in [43].

In this work, we are interested in applying and adapting these methods from wireless channels to network coding over DAGs. Fig. 2.1 depicts at high-level the two general approaches for network alignment: coding at the edge of the network (*e.g.,* using the eigen vector method [11], or using the asymptotic scheme in [11, 44]) or coding in the middle of the network (*e.g.,* using an ergodic scheme similar to [41]). We discuss both of these approaches in more detail.



(a) Alignment by coding in the middle of the network. Here, intermediate nodes perform linear network coding so as to guarantee alignment at the receivers.

(b) Alignment by coding at the edge of the network. Here, intermediate nodes perform random linear network coding, and the sources are responsible for coding their symbols so as to guarantee alignment.

Figure 2.1: Network alignment approaches.

2.5 Coding at the edge of the network

This approach is closer to the wireless paradigm. It uses only random linear network coding at the intermediate nodes and essentially creates a random linear channel between the sources and destination nodes. The intelligence and the processing burden is pushed to the edge, much like the wireless setting where the channel is decided by nature and all the coding/signal processing algorithms are applied only at the source and destination nodes. But unlike the wireless setting, the channels here are not necessarily independent of one another, they might have spatial correlation depending on the network structure. The downside to this approach is that it has a significant computational overhead at the end nodes and the more general method of this approach usually requires a very large amount of data due to its asymptotic nature.

2.5.1 Perfect Alignment - Eigenvector Method

The literature on interference alignment [11], describes a closed form scheme which uses eigenvectors of certain matrices to achieve perfect alignment of interferences in the 3 user interference channel setup. Since we are focusing on the 3 user case of the multiple unicast problem, we apply the above method to our model. In order to be able to make use of this method, we must employ vector network coding scheme in our model. In vector network coding, the source transmits vectors of length L, while the intermediate nodes process and combine their incoming packets by multiplying them with $L \times L$ coding matrices. The coding matrices play a similar role as coding coefficients in scalar coding. In our case, we consider that the source transmits vectors of length $L = 2$ and correspondingly the coding matrices are of size 2×2. Let Ξ_i represent the coding matrix corresponding to a coding coefficient ξ_i. The input-output relations in G can be written as

$$\mathbf{y}_1(b) = \mathbf{M}_{11}(b)\mathbf{x}_1(b) + \mathbf{M}_{12}(b)\mathbf{x}_2(b) + \mathbf{M}_{13}(b)\mathbf{x}_3(b), \tag{2.2}$$

$$\mathbf{y}_2(b) = \mathbf{M}_{21}(b)\mathbf{x}_1(b) + \mathbf{M}_{22}(b)\mathbf{x}_2(b) + \mathbf{M}_{23}(b)\mathbf{x}_3(b), \tag{2.3}$$

$$\mathbf{y}_3(b) = \mathbf{M}_{31}(b)\mathbf{x}_1(b) + \mathbf{M}_{32}(b)\mathbf{x}_2(b) + \mathbf{M}_{33}(b)\mathbf{x}_3(b), \tag{2.4}$$

$\mathbf{x}_i$ is the a 2×1 vector version of x_i and is defined as

$$\mathbf{x}_i(b) = \begin{bmatrix} x_i(2t-1) \\ x_i(2t) \end{bmatrix} \tag{2.5}$$

where t represents the symbol-time index and b represents the block-vector-time index. Similarly, $\mathbf{y}_i$ is a 2×1 vector version of y_i. $\mathbf{M}_{ij}(b)$ is a 2 diagonal matrix. We introduce "precoding" matrices $\mathbf{V}_1(b), \mathbf{V}_2(b), \mathbf{V}_3(b)$ for S_1, S_2, S_3 respectively, used to encode the message vectors into 2-length symbol vectors to be transmitted during the bth block. Moreover, the following relations hold:

$$\mathbf{x}_i(b) = \mathbf{V}_i(b) z_i(b), \quad i = 1, 2, 3. \tag{2.6}$$

where z_i is a message of user i. Note that $\mathbf{V}_i(b)$ is dependent only on the coding matrices chosen in the bth block. Hence, $\mathbf{V}_i(b)$ can potentially vary across blocks. Alternatively, we can think of this model as a system where the sources transmit the same message over two successive channel uses and the coding at the intermediate nodes are performed with in these two channel uses.

Since the system is memoryless across blocks, we focus our attention only on the bth symbol-extension. For notational convenience, we drop the symbol-extension index b. This gives the following modified input-output relations:

$$\begin{aligned}
\mathbf{y}_1 &= \mathbf{M}_{11}\mathbf{V}_1 z_1 + \mathbf{M}_{12}\mathbf{V}_2 z_2 + \mathbf{M}_{13}\mathbf{V}_3 z_3, && (2.7)\\
\mathbf{y}_2 &= \mathbf{M}_{21}\mathbf{V}_1 z_1 + \mathbf{M}_{22}\mathbf{V}_2 z_2 + \mathbf{M}_{23}\mathbf{V}_3 z_3, && (2.8)\\
\mathbf{y}_3 &= \mathbf{M}_{31}\mathbf{V}_1 z_1 + \mathbf{M}_{32}\mathbf{V}_2 z_2 + \mathbf{M}_{33}\mathbf{V}_3 z_3, && (2.9)
\end{aligned}$$

Now let's look at the coding matrices, if all these matrices have a diagonal structure, then it would become impossible to use eigenvector method as any matrix resulting from any linear operation of these coding matrix will again have a diagonal structure and the eigenvector of such a matrix will always be $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ or $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$. So we conclude that the coding matrices must not have a diagonal structure.

*Challenges:*On the other hand, if we consider all the entries of the coding matrices to be some random values from a finite field, then the eigenvalues and eigenvectors of any matrix, resulting from the linear operation of the coding matrices, might not belong to the finite field. Thus, we will assign a triangular structure to the coding matrices in order to overcome the hurdles stated above.

$$\Xi_i = \begin{bmatrix} \xi_i^{(1)} & 0 \\ \xi_i^{(3)} & \xi_i^{(2)} \end{bmatrix} \tag{2.10}$$

The eigenvalues of a triangular matrix are the diagonal entries of the matrix, which obviously belong to the finite field as all the entries come from a finite field and correspondingly we can also conclude the eigenvectors also belong to the same finite field.

Now, similar to [11], we perform "interference alignment" by imposing the following constraints on the precoding matrices for alignment and exact recovery of messages:

$$D_1 \quad : \quad span(\mathbf{M}_{12}\mathbf{V}_2) = span(\mathbf{M}_{13}\mathbf{V}_3) \tag{2.11}$$

$$rank\,[\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{12}\mathbf{V}_2] = 2 \tag{2.12}$$

$$D_2 \quad : \quad span(\mathbf{M}_{23}\mathbf{V}_3) = span(\mathbf{M}_{21}\mathbf{V}_1) \tag{2.13}$$

$$rank\,[\mathbf{M}_{22}\mathbf{V}_2 \quad \mathbf{M}_{21}\mathbf{V}_1] = 2 \tag{2.14}$$

$$D_3 \quad : \quad span(\mathbf{M}_{32}\mathbf{V}_2) = span(\mathbf{M}_{31}\mathbf{V}_1) \tag{2.15}$$

$$rank\,[\mathbf{M}_{33}\mathbf{V}_3 \quad \mathbf{M}_{31}\mathbf{V}_1] = 2 \tag{2.16}$$

since the resulting vectors are are 2×1 the *span* operation just indicates that they are not a scaled version of one another.

Consider conditions (2.11), (2.13) and (2.15), these result in a cycle of conditions as shown below,

$$\begin{aligned} span(\mathbf{V}_1) &= span(\mathbf{M}_{31}^{-1}\mathbf{M}_{32}\mathbf{V}_2) & (2.17) \\ span(\mathbf{V}_2) &= span(\mathbf{M}_{12}^{-1}\mathbf{M}_{13}\mathbf{V}_3) & (2.18) \\ span(\mathbf{V}_3) &= span(\mathbf{M}_{23}^{-1}\mathbf{M}_{21}\mathbf{V}_1) & (2.19) \end{aligned}$$

which results in,

$$span(\mathbf{V}_1) = span((\mathbf{M}_{31}^{-1}\mathbf{M}_{32}\mathbf{M}_{12}^{-1}\mathbf{M}_{13}\mathbf{M}_{23}^{-1}\mathbf{M}_{21})\mathbf{V}_1) \tag{2.20}$$

We define $\mathbf{T} = \mathbf{M}_{31}^{-1}\mathbf{M}_{32}\mathbf{M}_{12}^{-1}\mathbf{M}_{13}\mathbf{M}_{23}^{-1}\mathbf{M}_{21}$. These will hold only if the $\mathbf{M}_{ij}$'s are invertible. We can think of the diagonal entries of $\mathbf{M}_{ij}$'s as different realizations of the scalar polynomial m_{ij},

$$\mathbf{M}_{ij} = \begin{bmatrix} m_{ij}^{(1)} & 0 \\ \tilde{m}_{ij} & m_{ij}^{(2)} \end{bmatrix} \tag{2.21}$$

where $\tilde{m}_{ij}$ is a polynomial in $\begin{bmatrix} \underline{\xi}^{(1)} & \underline{\xi}^{(2)} & \underline{\xi}^{(3)} \end{bmatrix}$ and is different from m_{ij}, no proper relation could be determined between the two. . So saying that the matrix $\mathbf{M}_{ij}$ is invertible is the same as saying that the two instances of the polynomial m_{ij} is non-zero. By Schwartz-Zippel lemma (see Appendix A.1) we know that the non-trivial polynomial will have a non-zero realization almost surely. Schwartz-Zippel lemma also ensures that the polynomial $\tilde{m}_{ij}$ also has a non-zero realization, so that the triangular structure of the matrix is not affected.

Choosing the precoding vector $\mathbf{V}_1$ as the eigenvector of the matrix $\mathbf{T}$. This choice of the vector $\mathbf{V}_1$ ensures that the condition (2.20) is satisfied. It is important to note that we consider the eigenvector corresponding to the eigenvalue $\lambda_1 = \mathbf{T}(1,1)$ (*i.e.,* the $1st$ diagonal entry of $\mathbf{T}$),

$$\mathbf{V}_1 = \begin{bmatrix} 1 \\ \psi/(\lambda_1 - \lambda_2) \end{bmatrix} \tag{2.22}$$

where $\psi = \mathbf{T}(2,1)$. The other eigenvector would be $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ (corresponding to eigenvalue $\lambda_2 = \mathbf{T}(2,2)$) and this would not work for achieving alignment.

The other precoding matrices can be obtained from $\mathbf{V}_1$ using the following relation,

$$\mathbf{V}_2 = \mathbf{M}_{32}^{-1}\mathbf{M}_{31}\mathbf{V}_1 \tag{2.23}$$

$$\mathbf{V}_3 = \mathbf{M}_{23}^{-1}\mathbf{M}_{21}\mathbf{V}_1 \tag{2.24}$$

This choice of the precoding vectors ensures that the interferences are aligned at the destination. In addition to this, the conditions (2.12), (2.14) and (2.16) also need to be satisfied. Let's focus on the requirements for the condition (2.14) to hold, the other two can be derived in a similar fashion.

$[\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{12}\mathbf{V}_2]$ is a 2×2 square matrix, it is full rank iff its determinant is non-zero. $\mathbf{M}_{11}$ is the transfer function matrix for the direct path from S_1 to D_1, so it has non-zero diagonal elements which in turn implies that $\mathbf{M}_{11}$ is full rank. We have,

$$[\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{12}\mathbf{V}_2] = \left[\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{12}\mathbf{M}_{32}^{-1}\mathbf{M}_{31}\mathbf{V}_1\right] \tag{2.25}$$

The above matrix will have a zero determinant *iff* the following condition holds,

$$c\mathbf{M}_{11}\mathbf{V}_1 = \mathbf{M}_{12}\mathbf{M}_{32}^{-1}\mathbf{M}_{31}\mathbf{V}_1 \tag{2.26}$$

$$c\mathbf{V}_1 = (\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{M}_{32}^{-1}\mathbf{M}_{31})\mathbf{V}_1 \tag{2.27}$$

where c is some arbitrary constant. If we define $\mathbf{U} = \mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{M}_{32}^{-1}\mathbf{M}_{31}$, then (2.27) can be true *iff* c and $\mathbf{V}_1$ are respectively one of the eigenvalues and eigenvectors of $\mathbf{U}$. Even though the matrix multiplication of the 2×2 triangular matrices is not commutative, it is interesting to note that the diagonal entires alone still seem to remain commutative, *i.e.,* if A, B are two 2×2 lower triangular matrices, then even though $AB \neq BA$, we can see that the diagonal entries of AB and BA are the same. The condition (2.27) can be written as,

$$\begin{bmatrix} u(1,1) & 0 \\ u(2,1) & u(2,2) \end{bmatrix} \begin{bmatrix} 1 \\ \psi/(\lambda_1 - \lambda_2) \end{bmatrix} = c \begin{bmatrix} 1 \\ \psi/(\lambda_1 - \lambda_2) \end{bmatrix} \tag{2.28}$$

$$
\begin{bmatrix} \dfrac{m_{12}^{(1)}m_{31}^{(1)}}{m_{11}^{(1)}m_{32}^{(1)}} & 0 \\ u(2,1) & \dfrac{m_{12}^{(2)}m_{31}^{(2)}}{m_{11}^{(2)}m_{32}^{(2)}} \end{bmatrix} \begin{bmatrix} 1 \\ \psi/(\lambda_1-\lambda_2) \end{bmatrix} = c \begin{bmatrix} 1 \\ \psi/(\lambda_1-\lambda_2) \end{bmatrix} \tag{2.29}
$$

inspecting the above equation row wise, we can see that (2.29) can be simplified into the following two conditions

$$
\frac{m_{12}^{(1)}m_{31}^{(1)}}{m_{11}^{(1)}m_{32}^{(1)}} = c \tag{2.30}
$$

$$
\psi\left(c-\frac{m_{12}^{(2)}m_{31}^{(2)}}{m_{11}^{(2)}m_{32}^{(2)}}\right) = u(2,1)(\lambda_1-\lambda_2) \tag{2.31}
$$

(2.30) implies that the eigenvalue $\mathbf{U}$ is $\alpha_1 = \dfrac{m_{12}^{(1)}m_{31}^{(1)}}{m_{11}^{(1)}m_{32}^{(1)}} = c$. We know that the other eigenvalue is $\alpha_2 = \dfrac{m_{12}^{(2)}m_{31}^{(2)}}{m_{11}^{(2)}m_{32}^{(2)}}$.

$$
\psi\left(\alpha_1-\alpha_2\right) = u(2,1)(\lambda_1-\lambda_2) \tag{2.32}
$$

Understanding the conditions. These conditions are quite complex and it is not clear how to derive the requirements for the sufficiency of alignment in terms of network structure from them. But from (2.27), it is evident that the condition $\mathbf{M}_{11} \neq \mathbf{M}_{12}\mathbf{M}_{32}^{-1}\mathbf{M}_{31}$ is necessary to use the interference alignment approach. In terms of the network structure this condition can be simply stated as $m_{11}\left(\underline{\xi}\right) \neq \frac{m_{12}(\underline{\xi})m_{31}(\underline{\xi})}{m_{32}(\underline{\xi})}$.

In order to under stand the requirements of the network for the necessary and sufficient conditions of alignment, we shift our focus to the asymptotic alignment scheme where we deal with diagonal matrices. The asymptotic case is also interesting because it has the potential to be extended and applied to the cases with more than 3 users.

2.5.2 Precoding-Based Network Alignment

The asymptotic alignment scheme with symbol extension was introduced for wireless interference channels in [11] and later applied to multiple unicast network with network coding inn [44]. The scheme achieves asymptotic alignment ,*i.e.,* for large number of symbols, by pre-coding only at the sources of a network, while the rest of the network performs random linear network coding. The pre-coding matrices depend on the network transfer functions $m_{ij}(\underline{\xi})$'s. It is worth noting that the dependencies between $m_{ij}(\underline{\xi})$'s due to the network structure impose some conditions for symbol-extended alignment to be feasible. This is unlike the wireless setting, where $m_{ij}(\underline{\xi})$'s are channel gains and typically assumed to be independent.

Consider the symbol-extension resulting from $(2n+1)$ successive channel uses. The choice of $\underline{\xi}$ from $\mathbb{F}_p^s$ can possibly vary with each channel use, and this choice is denoted as $\underline{\xi}^{(k)}$ for the kth channel use. The symbol-extended version of the input-output relations in G can be written as

$$\mathbf{y}_1(b) = \mathbf{M}_{11}(b)\mathbf{x}_1(b) + \mathbf{M}_{12}(b)\mathbf{x}_2(b) + \mathbf{M}_{13}(b)\mathbf{x}_3(b), \tag{2.33}$$

$$\mathbf{y}_2(b) = \mathbf{M}_{21}(b)\mathbf{x}_1(b) + \mathbf{M}_{22}(b)\mathbf{x}_2(b) + \mathbf{M}_{23}(b)\mathbf{x}_3(b), \tag{2.34}$$

$$\mathbf{y}_3(b) = \mathbf{M}_{31}(b)\mathbf{x}_1(b) + \mathbf{M}_{32}(b)\mathbf{x}_2(b) + \mathbf{M}_{33}(b)\mathbf{x}_3(b), \tag{2.35}$$

which is similar to the i/o relations in section 2.5.1, except here $\mathbf{x}_i$ is a $(2n+1)\times 1$ vector representing the $(2n+1)$-length symbol-extended version of x_i and is defined as

$$\mathbf{x}_i(b) = \begin{bmatrix} x_i((2n+1)(t-1)+1) \\ x_i((2n+1)(t-1)+2) \\ \vdots \\ x_i((2n+1)t) \end{bmatrix} \tag{2.36}$$

where t represents the symbol-time index and b represents the block-vector-time index for the entire $(2n+1)$-length symbol-extension. Similarly, $\mathbf{y}_i$ is a $(2n+1)\times 1$ vector representing the $(2n+1)$ symbol extended version of y_i. $\mathbf{M}_{ij}(b)$ is a $(2n+1)\times(2n+1)$ diagonal matrix with the the (k,k)th entry as $m_{ij}(\underline{\xi}^{((2n+1)(t-1)+k)})$ for $k = 1, 2, \ldots, (2n+1)$. In [11, 44], precoding matrices $\mathbf{V}_1(b), \mathbf{V}_2(b), \mathbf{V}_3(b)$ were introduced for S_1, S_2, S_3 respectively, these are used to encode the message vectors into $(2n+1)$-length symbol vectors to be transmitted during the bth symbol-extension. This means that $\mathbf{V}_1(b)$ is a $(2n+1)\times(n+1)$ matrix. Moreover, the following relations hold:

$$\mathbf{x}_i(b) = \mathbf{V}_i(b)\mathbf{z}_i(b), \quad i = 1, 2, 3. \tag{2.37}$$

Note that $\mathbf{V}_i(b)$ is dependent only on the linear coding coefficients chosen in the bth symbol-extension. Hence, $\mathbf{V}_i(b)$ can potentially vary across symbol-extensions.

Here again, since the system is memoryless across blocks, it is fine to focus only on the bth symbol-extension. So the modified input-output relations is obtained by dropping the symbol-extension index b, and it is similar to (2.7)-(2.9). Then similar to [11], [44] performed "interference alignment" by imposing the following constraints on the precoding matrices for alignment and exact recovery of messages:

$$
\begin{aligned}
D_1 \quad &: \quad span(\mathbf{M}_{12}\mathbf{V}_2) = span(\mathbf{M}_{13}\mathbf{V}_3) && (2.38)\\
& \quad\; rank\left[\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{12}\mathbf{V}_2\right] = (2n+1) && (2.39)\\
D_2 \quad &: \quad span(\mathbf{M}_{23}\mathbf{V}_3) \subseteq span(\mathbf{M}_{21}\mathbf{V}_1) && (2.40)\\
& \quad\; rank\left[\mathbf{M}_{22}\mathbf{V}_2 \quad \mathbf{M}_{21}\mathbf{V}_1\right] = (2n+1) && (2.41)\\
D_3 \quad &: \quad span(\mathbf{M}_{32}\mathbf{V}_2) \subseteq span(\mathbf{M}_{31}\mathbf{V}_1) && (2.42)\\
& \quad\; rank\left[\mathbf{M}_{33}\mathbf{V}_3 \quad \mathbf{M}_{31}\mathbf{V}_1\right] = (2n+1) && (2.43)
\end{aligned}
$$

Note that within each symbol-extension block there are $(2n+1)$ realizations of $\underline{\xi}$. [44] uses the same framework as in [11], to choose the precoding matrices $\mathbf{V}_1, \mathbf{V}_2, \mathbf{V}_3$ as follows:

$$
\begin{aligned}
\mathbf{V}_1 &= \left[\mathbf{w} \quad \mathbf{T}\mathbf{w} \quad \mathbf{T}^2\mathbf{w}\ldots\mathbf{T}^n\mathbf{w}\right], && (2.44)\\
\mathbf{V}_2 &= \left[\mathbf{R}\mathbf{w} \quad \mathbf{R}\mathbf{T}\mathbf{w}\ldots\mathbf{R}\mathbf{T}^{n-1}\mathbf{w}\right], && (2.45)\\
\mathbf{V}_3 &= \left[\mathbf{S}\mathbf{T}\mathbf{w} \quad \mathbf{S}\mathbf{T}^2\mathbf{w}\ldots\mathbf{S}\mathbf{T}^n\mathbf{w}\right], && (2.46)
\end{aligned}
$$

where $\mathbf{w} = [\,1 \quad 1\ldots1\,]^T$ is a $(2n+1)\times 1$ vector of ones, $\mathbf{T} = \mathbf{M}_{12}\mathbf{M}_{23}\mathbf{M}_{31}\mathbf{M}_{13}^{-1}\mathbf{M}_{32}^{-1}\mathbf{M}_{21}^{-1}$, $\mathbf{R} = \mathbf{M}_{31}\mathbf{M}_{32}^{-1}$ and $\mathbf{S} = \mathbf{M}_{21}\mathbf{M}_{23}^{-1}$.

The equation (2.38)-(2.43) provide a necessary and sufficient condition for alignment. Let's focus on (2.38) for the moment, $[\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{12}\mathbf{V}_2]$ is a $(2n+1)\times(2n+1)$ square matrix, so it is full rank iff its determinant is non-zero. We have

$$\begin{aligned}
[\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{12}\mathbf{V}_2] &= \mathbf{M}_{11}\left[\mathbf{V}_1 \quad \mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{R}\mathbf{V}_1\mathbf{A}\right] && (2.47)\\
&= \mathbf{M}_{11}\left[\mathbf{V}_1 \quad \mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{M}_{31}\mathbf{M}_{32}^{-1}\mathbf{V}_1\mathbf{A}\right] && (2.48)
\end{aligned}$$

where A is a $(n+1)\times n$ compromising of the first n columns of the $(n+1)\times(n+1)$ identity matrix. The matrix $[\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{12}\mathbf{V}_2]$. $\mathbf{M}_{11}$ is an invertible diagonal matrix, so if $\left[\mathbf{V}_1 \quad \mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{M}_{31}\mathbf{M}_{32}^{-1}\mathbf{V}_1\mathbf{A}\right]$ is of full rank, then $[\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{12}\mathbf{V}_2]$ is also of full rank. In other words this matrix should have $(2n+1)$ linearly independent columns, *i.e.,* none of its columns can be written as a linear combination of other columns. Mathematically this condition is stated as follows,

$$\begin{aligned}
\sum_{i=0}^{n} p_i\mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{M}_{31}\mathbf{M}_{32}^{-1}\mathbf{T}^i &\neq \sum_{j=0}^{n} q_j\mathbf{T}^j && (2.49)\\
\Rightarrow \mathbf{M}_{11}^{-1}\mathbf{M}_{12}\mathbf{M}_{31}\mathbf{M}_{32}^{-1}\sum_{i=0}^{n} p_i\mathbf{T}^i &\neq \sum_{j=0}^{n} q_j\mathbf{T}^j && (2.50)
\end{aligned}$$

$$\Rightarrow \mathbf{M}_{11} \neq \mathbf{M}_{12}\mathbf{M}_{31}\mathbf{M}_{32}^{-1}\sum_{i=0}^{n} p_i\mathbf{T}^i\left(\sum_{j=0}^{n} q_j\mathbf{T}^j\right)^{-1} \qquad (2.51)$$

$\forall n$ and $\forall\, p_i, q_j \in \mathbb{F}_p$. Note that condition (2.51) is made only of $\mathbf{M}_{ij}$'s and essentially $\mathbf{M}_{ij}$ is a matrix with $(2n+1)$ realizations of the global coefficient polynomial m_{ij} (network transfer function). $\mathbf{M}_{ij}$'s being diagonal matrices, it is quite easy to translate the condition (2.51) into a condition involving the m_{ij}'s shown below.

$$m_{11}(\underline{\xi}) \not\equiv \frac{m_{12}(\underline{\xi})m_{31}(\underline{\xi})}{m_{32}(\underline{\xi})} \frac{\sum_{i=0}^{n} p_i \left(m_{12}(\underline{\xi})m_{23}(\underline{\xi})m_{31}(\underline{\xi})/m_{21}(\underline{\xi})m_{13}(\underline{\xi})m_{32}(\underline{\xi})\right)^i}{\sum_{j=0}^{n} q_j \left(m_{12}(\underline{\xi})m_{23}(\underline{\xi})m_{31}(\underline{\xi})/m_{21}(\underline{\xi})m_{13}(\underline{\xi})m_{32}(\underline{\xi})\right)^j}, \tag{2.52}$$

In a similar fashion it is possible to derive the feasibility condition for m_{22} and m_{33}. We define the polynomials $a(\underline{\xi})$ and $b(\underline{\xi})$ as

$$a\left(\underline{\xi}\right) = m_{12}\left(\underline{\xi}\right)m_{23}\left(\underline{\xi}\right)m_{31}\left(\underline{\xi}\right), \tag{2.53}$$

$$b\left(\underline{\xi}\right) = m_{21}\left(\underline{\xi}\right)m_{13}\left(\underline{\xi}\right)m_{32}\left(\underline{\xi}\right). \tag{2.54}$$

Assuming the polynomials $m_{ij}(\underline{\xi})$'s to be non-trivial, we can say that the following conditions should hold for alignment to be feasible:

$$m_{11}\left(\underline{\xi}\right) \not\equiv \frac{m_{12}\left(\underline{\xi}\right)m_{31}\left(\underline{\xi}\right)}{m_{32}\left(\underline{\xi}\right)} \frac{\sum_{i=0}^{n} p_i \left(a\left(\underline{\xi}\right)/b\left(\underline{\xi}\right)\right)^i}{\sum_{j=0}^{n} q_j \left(a\left(\underline{\xi}\right)/b\left(\underline{\xi}\right)\right)^j}, \tag{2.55}$$

$$m_{22}\left(\underline{\xi}\right) \not\equiv \frac{m_{21}\left(\underline{\xi}\right)m_{32}\left(\underline{\xi}\right)}{m_{31}\left(\underline{\xi}\right)} \frac{\sum_{i=0}^{n} p_i \left(a\left(\underline{\xi}\right)/b\left(\underline{\xi}\right)\right)^i}{\sum_{j=0}^{n} q_j \left(a\left(\underline{\xi}\right)/b\left(\underline{\xi}\right)\right)^j}, \tag{2.56}$$

$$m_{33}\left(\underline{\xi}\right) \not\equiv \frac{m_{23}\left(\underline{\xi}\right)m_{31}\left(\underline{\xi}\right)}{m_{21}\left(\underline{\xi}\right)} \frac{\sum_{i=0}^{n} p_i \left(a\left(\underline{\xi}\right)/b\left(\underline{\xi}\right)\right)^i}{\sum_{j=0}^{n} q_j \left(a\left(\underline{\xi}\right)/b\left(\underline{\xi}\right)\right)^j}. \tag{2.57}$$

$\forall n$ and $\forall p_i, q_j \in \mathbb{F}_p$. Whenever the above conditions are satisfied, we can rearrange them into a non-zero polynomial, and by Schwartz-Zippel Lemma (Appendix A.1) a random realization of the coefficients would satisfy the conditions (2.38), (2.40) and (2.42) almost surely.

2.5.3 Understanding the Channel Transfer Functions

The channel transfer functions, m_{ij}'s, are the polynomial representation of the global coding coefficient for any message from source S_j to destination D_i., *i.e.,* the destination D_i will see any message originating from source S_j scaled by the coefficient m_{ij}. In a DAG, there might be several paths connecting S_j to D_i, if we consider one such path, then the coding coefficient for this path is just an accumulation (product) of all the local coding coefficients (ξ_i's) corresponding to the edges and nodes that the path follows. The final m_{ij} is a summation of the coding coefficient of all possible paths connect that source S_j to destination D_i. We observe few important characteristics of this network transfer function which might help us later with interpreting the conditions we have.

- The maximum degree of any local coefficient in any m_{ij} is 1. This is evident from the fact that there are no cycle present in a DAG and the local coefficient ξ_i is distinct for every node and edge.

- The m_{ij} as explained earlier is a sum of coefficients of all possible individual paths connecting the S_j to D_i, so we can see the final polynomial to be of sum-of-products. If the structure of the network permits it is also possible to rewrite the same polynomial as a product of its factors. This is analogous to seeing the network as a line graph. The polynomial is irreducible if there is no single edge common to all possible paths that belong to m_{ij}. If there exists an edge which is common to all paths forming the m_{ij}, then it is possible to write the polynomial as a product of two factors, one corresponding to the part before this edge and the other corresponding to the part after this edge.

- Any two distinct m_{ij}'s can have a common factor only if all possible path corresponding to both the m_{ij}'s have at least one common bottleneck edge.

- If two polynomial m_{ij} and m_{kj} have common factors, then the common factors are the polynomial representation of the network from the source S_j up to the final edge where all paths of $S_j - D_i$ and $S_j - D_k$ overlap. In other, $S_j - D_i$ and $S_j - D_k$ share the same set of paths from S_j to the final overlapping edge.

- If two polynomial m_{ij} and m_{ik} have common factors, then the common factors are the polynomial representation of the network from the first edge where all paths of $S_j - D_i$ and $S_k - D_i$ overlap up to the sink D_i . In other, $S_j - D_i$ and $S_k - D_i$ share the same set of paths from the first overlapping edge to D_i.

- If two polynomial m_{ij} and m_{kl} have common factor, the the common factors are the polynomial representation of the network from the first edge where all paths of $S_j - D_i$ and $S_l - D_k$ overlap up to the final edge where all paths of $S_j - D_i$ and $S_l - D_k$ overlap. In other, $S_j - D_i$ and $S_l - D_k$ share the same set of paths from the first overlapping edge up to the final overlapping edge.

2.5.4 The Initial Conjecture

In order to get a better intuition about the relation between these conditions and the network structure, let us consider a subset of these conditions that is obtained by considering some special, small values of the constants p_i, q_j. In particular, the conditions at the left are obtained for $p_0 = 1, q_0 = 1$, and the conditions at the right are obtained for $p_0 = 1, q_1 = 1$ or $p_1 = 1, q_0 = 1$; the rest p_i's, q_j's are set to zero:

$$m_{11}\left(\underline{\xi}\right) \neq \frac{m_{12}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right)}{m_{32}\left(\underline{\xi}\right)}, \qquad m_{11}\left(\underline{\xi}\right) \neq \frac{m_{21}\left(\underline{\xi}\right) m_{13}\left(\underline{\xi}\right)}{m_{23}\left(\underline{\xi}\right)}, \tag{2.58}$$

$$m_{22}\left(\underline{\xi}\right) \neq \frac{m_{21}\left(\underline{\xi}\right) m_{32}\left(\underline{\xi}\right)}{m_{31}\left(\underline{\xi}\right)}, \qquad m_{22}\left(\underline{\xi}\right) \neq \frac{m_{12}\left(\underline{\xi}\right) m_{23}\left(\underline{\xi}\right)}{m_{13}\left(\underline{\xi}\right)}, \tag{2.59}$$

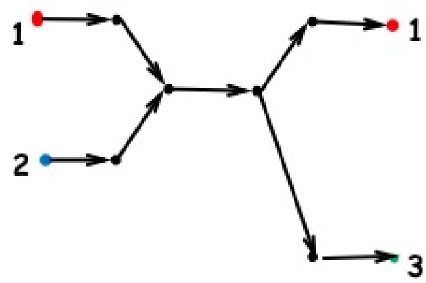
$$m_{33}\left(\underline{\xi}\right) \neq \frac{m_{23}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right)}{m_{21}\left(\underline{\xi}\right)}, \qquad m_{33}\left(\underline{\xi}\right) \neq \frac{m_{32}\left(\underline{\xi}\right) m_{13}\left(\underline{\xi}\right)}{m_{12}\left(\underline{\xi}\right)}. \tag{2.60}$$

The above six conditions require that the co-factor of any off-diagonal term of the 3×3 network transfer matrix $\mathbf{M} = \left[m_{ij}\left(\underline{\xi}\right)\right]$ is not a zero polynomial. Let us consider one such off-diagonal term, *e.g.*, $m_{23}\left(\underline{\xi}\right)$ with its cofactor:

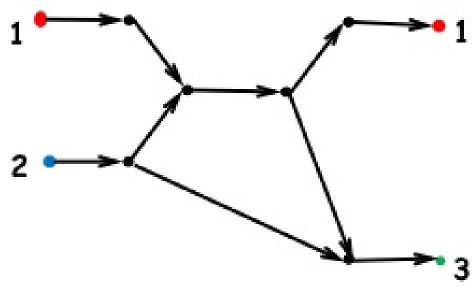
$$\det \begin{bmatrix} m_{11}\left(\underline{\xi}\right) & m_{12}\left(\underline{\xi}\right) \\ m_{31}\left(\underline{\xi}\right) & m_{32}\left(\underline{\xi}\right) \end{bmatrix} = m_{11}\left(\underline{\xi}\right) m_{32}\left(\underline{\xi}\right) - m_{12}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right). \tag{2.61}$$

The condition $m_{11}\left(\underline{\xi}\right) m_{32}\left(\underline{\xi}\right) - m_{12}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right) \not\equiv 0$ essentially states that aligning interference at one receiver (in this case, interference $m_{31}\left(\underline{\xi}\right), m_{32}\left(\underline{\xi}\right)$ from S_1, S_2 at D_3) should not have the undesired side-effect of aligning the signal with the interference at other receivers (in this case, interference $m_{12}\left(\underline{\xi}\right)$ and signal $m_{11}\left(\underline{\xi}\right)$ at D_2).

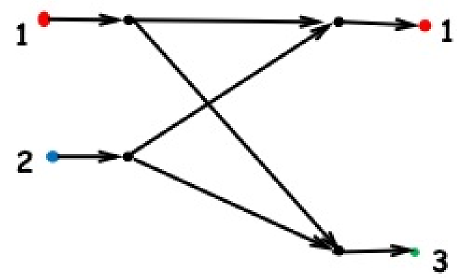
Consider a network with the structure shown in Fig. 2.2a, in this network all paths of the four commodities involved (from S_1 to D_1, S_2 to D_1, S_1 to D_3, and S_2 to D_3) go through a single bottleneck edge. This common edge causes the m_{ij}'s of the four involved commodities to have a common factor. The common factors produces a relation between the m_{ij}'s of the four commodities such that $m_{11}\left(\underline{\xi}\right) m_{32}\left(\underline{\xi}\right) - m_{12}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right) = 0$. In general, if all paths, of the any four commodities of any one of the six condition stated above, go through a bottleneck edge then the corresponding condition will bear an equality. In other words, the conditions state that not all paths for the four commodities involved in a condition should go



(a) Structure with bottleneck



(b) Bypassing the bottleneck



(c) No bottleneck

Figure 2.2: Feasible and Infeasible structure: (a) shows a structure where interference alignment is not feasible because of the the bottleneck edge. (b) shows a structure where there is a path for at least one commodity that doesn't go through the bottle neck edge, interference alignment is feasible in this scenario. (c) shows a structure where there is no bottleneck edge, thereby reducing the spatial dependency, interference alignment is feasible here.

through the same bottleneck edge. Fig. 2.2b shows an example structure where the condition holds true, here one of the commodities (S_2 to D_3) has a path that doesn't go through the bottleneck edge. Fig. 2.2c shows another such structure where the condition holds true, but here the paths are almost independent of each other. Interestingly, the conditions about the non-zero off-diagonal co-factors coincide with the necessary conditions for achieving a rate more than 1/3 per user through any achievable scheme, as shown previously in [45] based on Shannon theoretic arguments for the 3-user wireless interference network setting. It is also observed in simulations that these conditions seem to hold whenever interference alignment is possible. This led us to the conjecture [46] that the complex looking conditions in (2.55)-(2.57) can be reduced to the simple conditions given by (2.58)-(2.60), which can be easily interpreted in terms of the network structure. It is also important to note that although these conditions are always met (with high probability) in wireless channels, they

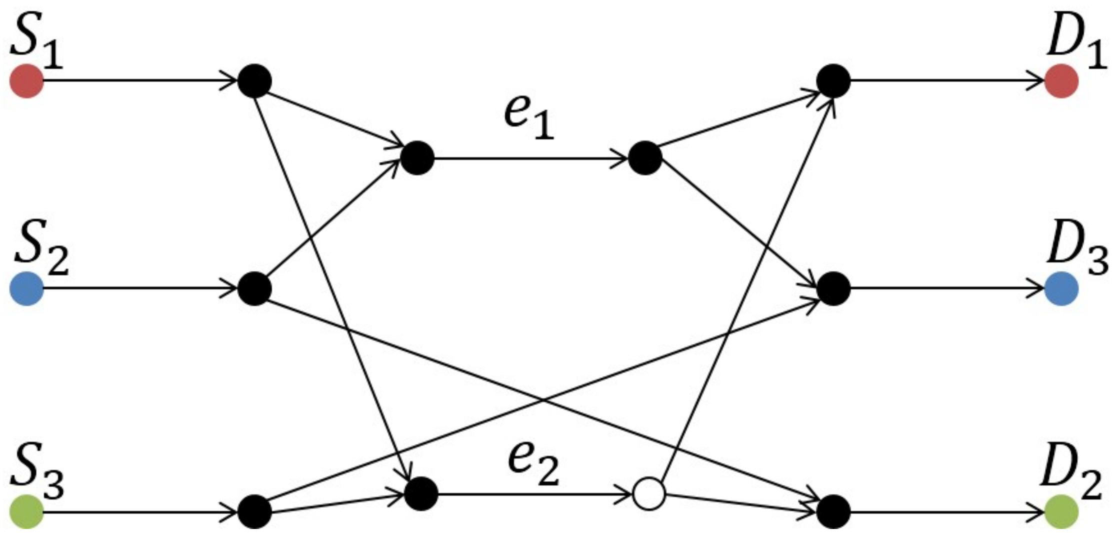


Figure 2.3: Network structure corresponding to a linear combination of the two conditions from the conjecture. Alignment is not feasible in this case too.

may not hold for all DAGs. In general, not all networks can be aligned. Therefore, it is interesting to characterize the feasibility and performance of alignment and their relation to network structure.

2.5.5 Extending the Conjecture

After proposing the conjecture in [46], we observed another class of network structures not covered by the conditions in our conjectures, where network alignment is not feasible. Figure 2.3 illustrates this additional network structure. We observed that this structure corresponds to the condition shown below,

$$m_{11}\left(\underline{\xi}\right) = \frac{m_{12}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right)}{m_{32}\left(\underline{\xi}\right)} + \frac{m_{21}\left(\underline{\xi}\right) m_{13}\left(\underline{\xi}\right)}{m_{23}\left(\underline{\xi}\right)} \tag{2.62}$$

It is easy to see that this condition is a linear combination of the two simple conditions $m_{11}\left(\underline{\xi}\right) = \frac{m_{12}(\underline{\xi})m_{31}(\underline{\xi})}{m_{32}(\underline{\xi})}$ and $m_{11}\left(\underline{\xi}\right) = \frac{m_{21}(\underline{\xi})m_{13}(\underline{\xi})}{m_{23}(\underline{\xi})}$ which corresponds to the structures with bottleneck like the one shown in Figure 2.2a. In the structure, we can see that there are two distinct paths leading from S_1 to D_1, shown with blue and green edges. Each of these paths, independently form a bottleneck structure similar to the one shown in Figure 2.2a.

2.5.6 Proving the Conjecture

In [47, 48], we were able to rigorously prove that the extended conjecture was indeed true. So the necessary and sufficient feasibility condition for precoding-based network alignment can be reduced to the a few sets of conditions as shown in the theorem below,

Theorem 2.1 (The Main Theorem)**.** *Assume that all the senders are connected to all the receivers via directed paths. The three unicast sessions can asymptotically achieve the rate tuple* $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ *through PBNA if and only if the following conditions are satisfied:*

$$m_{11}\left(\underline{\xi}\right) \neq \frac{m_{13}\left(\underline{\xi}\right) m_{21}\left(\underline{\xi}\right)}{m_{23}\left(\underline{\xi}\right)}, \frac{m_{12}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right)}{m_{32}\left(\underline{\xi}\right)}, \frac{m_{13}\left(\underline{\xi}\right) m_{21}\left(\underline{\xi}\right)}{m_{23}\left(\underline{\xi}\right)} + \frac{m_{12}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right)}{m_{32}\left(\underline{\xi}\right)} \tag{2.63}$$

$$m_{22}\left(\underline{\xi}\right) \neq \frac{m_{12}\left(\underline{\xi}\right) m_{23}\left(\underline{\xi}\right)}{m_{13}\left(\underline{\xi}\right)}, \frac{m_{32}\left(\underline{\xi}\right) m_{21}\left(\underline{\xi}\right)}{m_{31}\left(\underline{\xi}\right)}, \frac{m_{12}\left(\underline{\xi}\right) m_{23}\left(\underline{\xi}\right)}{m_{13}\left(\underline{\xi}\right)} + \frac{m_{32}\left(\underline{\xi}\right) m_{21}\left(\underline{\xi}\right)}{m_{31}\left(\underline{\xi}\right)} \tag{2.64}$$

$$m_{33}\left(\underline{\xi}\right) \neq \frac{m_{23}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right)}{m_{21}\left(\underline{\xi}\right)}, \frac{m_{13}\left(\underline{\xi}\right) m_{32}\left(\underline{\xi}\right)}{m_{12}\left(\underline{\xi}\right)}, \frac{m_{23}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right)}{m_{21}\left(\underline{\xi}\right)} + \frac{m_{13}\left(\underline{\xi}\right) m_{32}\left(\underline{\xi}\right)}{m_{12}\left(\underline{\xi}\right)} \tag{2.65}$$

2.5.7 Graph Theoretic Formulation of the Feasibility Conditions

Given two subsets of edges, S and D, a cut-set C between S and D is a subset of edges, the removal of which will disconnect every directed path from S to D. The capacity of cut-set C is defined as the summation of the capacities of the edges contained in C. The minimum cut between S and D is the minimum capacity of all cut-sets between S and D.

Theorem 2.2. $m_{ii}\left(\underline{\xi}\right) = \frac{m_{ij}\left(\underline{\xi}\right) m_{ki}\left(\underline{\xi}\right)}{m_{kj}\left(\underline{\xi}\right)}$ *if and only if the minimum cut between* $\{S_i, S_j\}$ *and* $\{D_i, D_j\}$ *equals one.*

Proof. The transfer matrix between the sources S_i, S_j to the sinks D_i, D_k for the the given network can be written as the 2×2 matrix

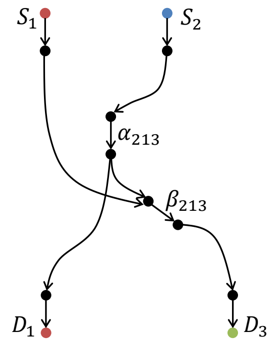
$$\begin{pmatrix} m_{ii}\left(\underline{\xi}\right) & m_{ij}\left(\underline{\xi}\right) \\ m_{ki}\left(\underline{\xi}\right) & m_{kj}\left(\underline{\xi}\right) \end{pmatrix}$$

The determinant of this transfer matrix is given as $m_{ii}\left(\underline{\xi}\right) m_{kj}\left(\underline{\xi}\right) - m_{ij}\left(\underline{\xi}\right) m_{ki}\left(\underline{\xi}\right)$. If the min-cut between $\{S_i, S_j\}$ and $\{D_i, D_j\}$ equals one, then the rank of the transfer matrix is one and so the determinant is zero, *i.e.*,

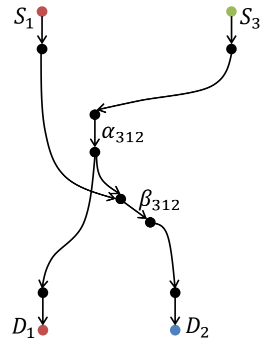
$$m_{ii}\left(\underline{\xi}\right) = \frac{m_{ij}\left(\underline{\xi}\right) m_{ki}\left(\underline{\xi}\right)}{m_{kj}\left(\underline{\xi}\right)}$$

If the min-cut between $\{S_i, S_j\}$ and $\{D_i, D_j\}$ is greater than one, then the transfer matrix has full rank. This implies that the determinant polynomial of the transfer matrix is a non-zero polynomial and thus proving the converse. □

For instance, in Figure 2.2a , the cut-set between $\{S_1, S_2\}$ and $\{D_1, D_3\}$ contains only one edge and thus $m_{11}\left(\underline{\xi}\right) = \frac{m_{12}\left(\underline{\xi}\right) m_{31}\left(\underline{\xi}\right)}{m_{32}\left(\underline{\xi}\right)}$.



(a) α_{213} and β_{213}



(b) α_{312} and β_{312}

Figure 2.4: A graphical illustration of the four edges, α_{213}, β_{213}, α_{312}, and β_{312}.

Let α_{ijk} denote the last bottleneck between S_i and $\{D_j, D_k\}$ in this topological ordering, and β_{ijk} the first bottleneck between $\{S_i, S_j\}$ and D_k. Let $\eta\left(\underline{\xi}\right) = \frac{m_{12}(\underline{\xi})m_{31}(\underline{\xi})m_{23}(\underline{\xi})}{m_{32}(\underline{\xi})m_{21}(\underline{\xi})m_{13}(\underline{\xi})}$. As shown below, the four edges, α_{213}, β_{213}, α_{312}, and β_{312}, are important in defining the networks that realize $\eta\left(\underline{\xi}\right) = 1$. A graphical illustration of the four edges is shown in Fig. 2.4.

Theorem 2.3. *$\eta\left(\underline{\xi}\right) = 1$ if and only if $\alpha_{213} = \alpha_{312}$ and $\beta_{213} = \beta_{312}$.*

In [49], the authors proved the above theorem giving a graph theoretic representation of the condition $\eta\left(\underline{\xi}\right) = 1$.

Given two edges e_1 and e_2, we say that they are parallel with each other if there is no directed paths from e_1 to e_2, or from e_2 to e_1.

Theorem 2.4. *$m_{ii}\left(\underline{\xi}\right) = \frac{m_{ij}(\underline{\xi})m_{ki}(\underline{\xi})}{m_{kj}(\underline{\xi})} + \frac{m_{ji}(\underline{\xi})m_{ik}(\underline{\xi})}{m_{jk}(\underline{\xi})}$ if and only if the following conditions are satisfied*

1. *α_{kij} is a bottleneck between S_i and D_j.*

2. *α_{jik} is a bottleneck between S_i and D_k.*

3. *α_{kij} is parallel to α_{jik}.*

4. *$\{\alpha_{jik}, \alpha_{kij}\}$ forms a cut-set between S_i and D_i.*

Proof. Refer Appendix A.2 for the proof. □

2.6 Optimal Symmetric Rates for Precoding-Based Linear Schemes

For SISO scenarios where all senders are connected to all receivers, there are only three possible rates achievable through any precoding-based network coding schemes.

Definition 2.1. *We classify the networks based on the coupling relations present in the network as follows:*

- *Type I : Networks in which a condition of the form* $m_{ii}\left(\underline{\xi}\right) = \frac{m_{ij}(\underline{\xi})m_{ki}(\underline{\xi})}{m_{kj}(\underline{\xi})}$ *is present.*
- *Type II : Networks in which* $m_{ii}\left(\underline{\xi}\right) \neq \frac{m_{ij}(\underline{\xi})m_{ki}(\underline{\xi})}{m_{kj}(\underline{\xi})}$*, but one of the three mutually exclusive conditions of the form* $m_{ii}\left(\underline{\xi}\right) = \frac{m_{ij}(\underline{\xi})m_{ki}(\underline{\xi})}{m_{kj}(\underline{\xi})} + \frac{m_{ji}(\underline{\xi})m_{ik}(\underline{\xi})}{m_{jk}(\underline{\xi})}$ *is present*
- *Type III : Networks in which none of the above conditions are present.*

In this section, we prove that for SISO scenarios where all the senders are connected to all the receivers via directed paths, there are only three possible symmetric rates achieved by any precoding-based linear schemes. We'll also show that these optimal symmetric rates for Precoding-based linear schemes can be achieved by slightly modifying the PBNA scheme. In order to show this, we first prove that for the networks that violate one of the following three conditions of the form $m_{ii}\left(\underline{\xi}\right) = \frac{m_{ij}(\underline{\xi})m_{ki}(\underline{\xi})}{m_{kj}(\underline{\xi})} + \frac{m_{ji}(\underline{\xi})m_{ik}(\underline{\xi})}{m_{jk}(\underline{\xi})}$, it is not possible to achieve a symmetric rate of more than 2/5 per user, through any precoding-based scheme (the proof follows from [39]). We also show that this outer bound of 2/5 is achievable through our PBNA scheme and thus it is tight.

Consider any precoding-based linear scheme over N channel uses. Let $\tilde{v}_1, \tilde{v}_2, \tilde{v}_3$ be vectors from the spaces span($\mathbf{V}_1$), span($\mathbf{V}_2$), and span($\mathbf{V}_3$), respectively. Consider a *Type II* network, without loss of generality , we assume that the network realizes $m_{11}\left(\underline{\xi}\right) = \frac{m_{12}(\underline{\xi})m_{31}(\underline{\xi})}{m_{32}(\underline{\xi})} + \frac{m_{21}(\underline{\xi})m_{13}(\underline{\xi})}{m_{23}(\underline{\xi})}$. This relation can be equivalently represented in matrix form as

$$\mathbf{M}_{11} = \mathbf{M}_{31}\mathbf{M}_{32}{}^{-1}\mathbf{M}_{12} + \mathbf{M}_{21}\mathbf{M}_{23}{}^{-1}\mathbf{M}_{13} \tag{2.66}$$

Lemma 2.5. *If $\tilde{v}_1$ aligns with $\tilde{v}_3$ at d_2 and with $\tilde{v}_2$ at d_3, then $\tilde{v}_1$ must align in the space spanned by $\tilde{v}_2$ and $\tilde{v}_3$ at d_1.*

Proof. Since $\tilde{v}_1$ aligns with $\tilde{v}_3$ at d_2 and with $\tilde{v}_2$ at d_3, it follows that,

$$d_2: \quad \mathbf{M}_{12}\tilde{v}_1 = a\ \mathbf{M}_{32}\tilde{v}_3 \tag{2.67}$$

$$d_3: \quad \mathbf{M}_{13}\tilde{v}_1 = b\ \mathbf{M}_{23}\tilde{v}_3 \tag{2.68}$$

where a, b are scalars. At d_1, we see the vector $\mathbf{M}_{11}\tilde{v}_1$. Using (2.66), (2.67) and (2.68) we get,

$$\begin{aligned}\mathbf{M}_{11}\tilde{v}_1 &= \mathbf{M}_{31}\mathbf{M}_{32}{}^{-1}\mathbf{M}_{12}\tilde{v}_1 + \mathbf{M}_{21}\mathbf{M}_{23}{}^{-1}\mathbf{M}_{13}\tilde{v}_1 \\ &= a\ \mathbf{M}_{31}\tilde{v}_2 + b\ \mathbf{M}_{21}\tilde{v}_2\end{aligned}$$

This shows that the desired vector at d_1 aligns with the space spanned by the interference. □

Theorem 2.6. *For a Type II network the symmetric rate achievable per user through any precoding-based scheme cannot be more than* 2/5*.*

Proof. Suppose every sender sends d symbols over n dimensions, through any linear precoding scheme. Consider ω_1 , lets use l_{12} and l_{13} to represent the number of dimensions of signal space of d_1 that align with ω_2 at d_3 and ω_3 at d_2 respectively, and V_{12} and V_{13}

to represent their corresponding spaces. From Lemma 2.5, we know that V_{12} and V_{13} must have no intersection, otherwise the intersection part will contain vectors that will align with interference at d_1. Therefore, we must have $l_{12} + l_{13} \leq d$. Now consider ω_2, we already know that there is a l_{13} dimensional space where interference from ω_1 and ω_3 are aligned. So the number of interference dimension is given as $(d + d - l_{13}) = 2d - l_{13}$. The number of desired dimensions at d_2 is d, and this d dimensional desired signal space should remain resolvable from the interference space, so we we have $3d - l_{13} \leq n$. Similarly, consider User 3 to obtain another inequality : $3d - l_{12} \leq n$. Combining these inequalities we get $6d - (l_{13} + l_{12}) \leq n$. But we know $l_{12} + l_{13} \leq d$, so $6d - d \leq 2n \Rightarrow d/n \leq 2/5$, which implies it is not possible to achieve a symmetric rate more than 2/5 per user. $\square$

Corollary 2.7. *For Type II networks, it is possible to achieve a rate of* $2/5$ *per user through through a finite time-slot precoding based network alignment scheme,* i.e., *the outer bound is tight.*

Proof. Without loss of generality, assume the *Type II* networks has a coupling relation $m_{11}\left(\underline{\xi}\right) = \frac{m_{12}(\underline{\xi})m_{31}(\underline{\xi})}{m_{32}(\underline{\xi})} + \frac{m_{21}(\underline{\xi})m_{13}(\underline{\xi})}{m_{23}(\underline{\xi})}$. This scheme can be easily modified to fit the other coupling relations too. Suppose we use a $2n + 1 = 5$ symbol extension, then according to the PBNA scheme we have precoding vectors $\mathbf{V}_1 = (\mathbf{w}\ \mathbf{Tw}\ \mathbf{T}^2\mathbf{w})$, $\mathbf{V}_2 = (\mathbf{w}\ \mathbf{Tw})$ and $\mathbf{V}_3 = (\mathbf{Tw}\ \mathbf{T}^2\mathbf{w})$. The given coupling relation only affects User 1, so the rates at Receiver 2 and 3 will remain unaffected. The matrix equivalent of the coupling relation is given in (2.66), which can be rewritten as,

$$\mathbf{M}_{11} = \mathbf{M}_{31}\mathbf{M}_{32}{}^{-1}\mathbf{M}_{12} + \mathbf{M}_{31}\mathbf{M}_{32}{}^{-1}\mathbf{M}_{12}\mathbf{T} \tag{2.69}$$

At Receiver 1, the desired signal space is given $\mathbf{M}_{11}\mathbf{V}_1$ and the interference space is given by $\mathbf{M}_{31}\mathbf{V}_3$ (Note: The interference from transmitter 2 and 3 are aligned, *i.e.*, $\mathbf{M}_{21}\mathbf{V}_2 =$

$\mathbf{M}_{31}\mathbf{V}_3$). Substituting the alignment equation from Receiver 2 for $\mathbf{V}_3$ we get,

$$\mathbf{M}_{31}\mathbf{V}_3 = \mathbf{M}_{31}{\mathbf{M}_{32}}^{-1}\mathbf{M}_{12}(\mathbf{Tw}\ \mathbf{T}^2\mathbf{w}) \tag{2.70}$$

From (2.69) and (2.70), it can be seen that the second column of the desired signal space ($\mathbf{M}_{11}\mathbf{V}_1$) can be written as a linear combination of the two columns of the interference space. The other two columns of the desired space are linearly independent of the column of interference space. User 1 could use these two dimension to send its signal without interference. In other words, each user would be able to achieve a rate of 2/5 □

Theorem 2.8. *Assume that all the senders are connected to all the receivers via directed paths. The following statements hold:*

1. *The optimal symmetric rate achieved by precoding-based linear schemes for Type I networks is* $1/3$ *per unicast session.*

2. *The optimal symmetric rate achieved by precoding-based linear schemes for Type II networks is* $2/5$ *per unicast session.*

3. *The optimal symmetric rate achieved by precoding-based linear schemes for Type III networks is* $1/2$ *per unicast session.*

Moreover, all of the above optimal symmetric rate is achievable through PBNA schemes.

Proof. *Type I* networks fail to satisfy certain conditions which are information theoretically necessary to achieve any rate more than 1/3 user per session, this was explained in Section 2.5.4. The outer bound for *Type II* networks was derived in Theorem 2.6 and the achievability was shown in Corollary 2.7. *Type III* networks are our main focus , previous sections discussed in detail about schemes and their feasibility for achieving 1/2 rate per user in detail and it is a well known fact that it is not possible to achieve more than 1/2 per user for SISO scenarios in fully connected networks [11]. $\square$

2.7 Coding in the middle of the network

This approach is depicted on the left side of Fig. 2.1 and can include several algorithms. The main idea behind this approach is to choose the coding coefficients at the intermediate nodes such that, the ratios m_{12}/m_{13}, m_{12}/m_{13} and m_{12}/m_{13} are constants, whereas the ratio of the coefficient of the direct channel to the interfering channel is not a constant . One such algorithm is ergodic alignment, originally developed for wireless channels in [41] and proposed for network coding over graphs in [50]. A general approach for coding in the network, is outside the scope of this thesis. However, in Chapter 2.8, where we compare any NA approach to baselines (other known schemes that can achieve 1/2 the rate per session), we provide examples of network alignment where coding is performed in the middle.

This approach is particularly attractive in the network coding setting because it operates in two time slots and thus can lead to practical solutions in terms of the required field size and number of symbols. Moreover, unlike the wireless setting, in the network coding setting the channel conditions are determined by the operations at the intermediate nodes and hence can be controlled. This simplicity comes at the cost of introducing intelligence in the network, and depending on the size of the network this optimization may be difficult.

2.8 Benefit Analysis

In this chapter, we try to understand in which classes of networks alignment is necessary and in which classes it is not (i.e., existing approaches achieve 1/2 rate without alignment). To

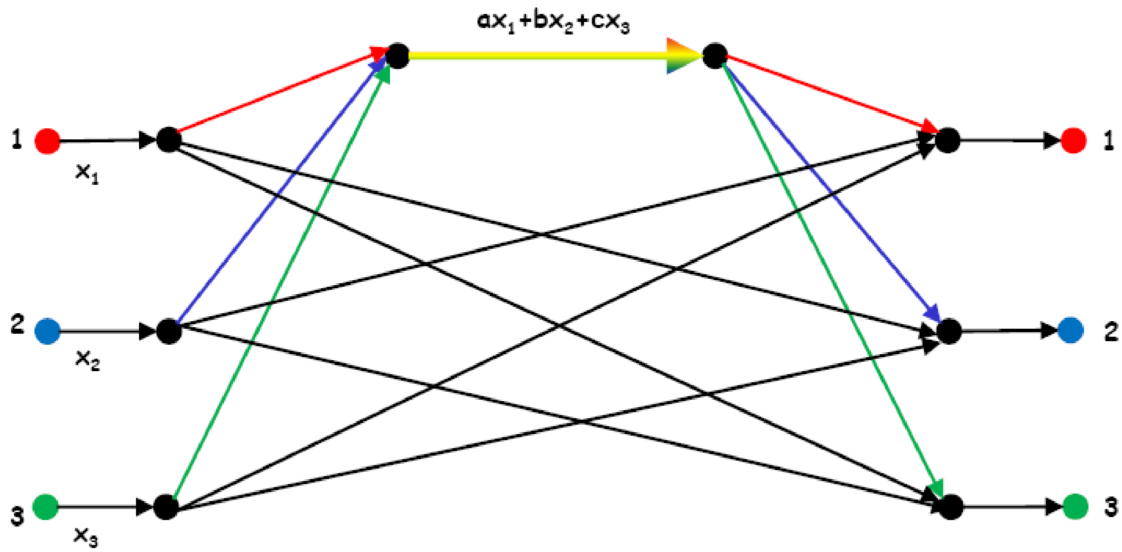
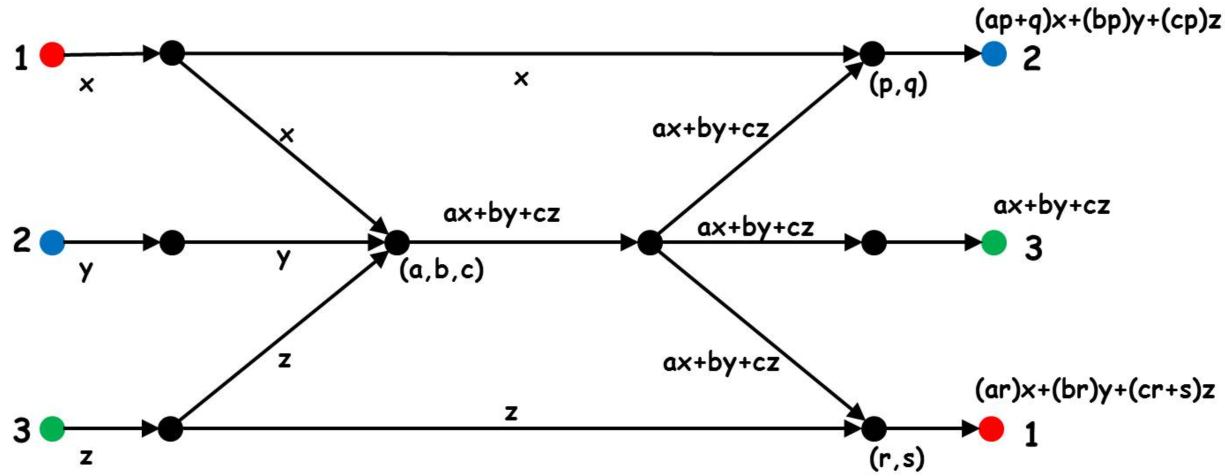


Figure 2.5: Examples for 3 unicast sessions, min-cut 1 where all 3 sessions go through a single bottleneck. The extended butterfly: routing achieves 1/3, alignment achieves 1/2, and network coding achieves rate 1 per session.

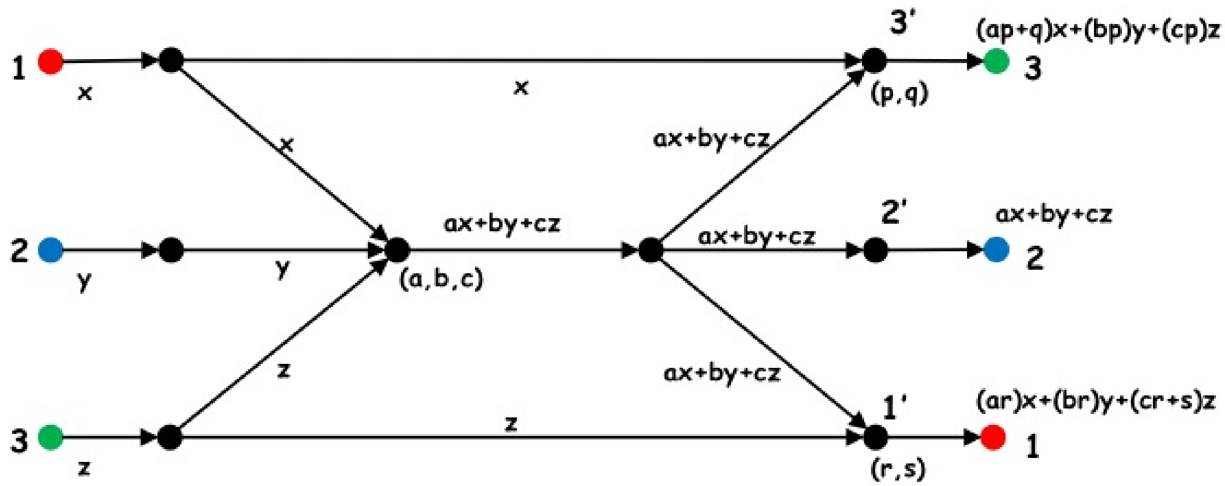
this end, we first consider some illustrative examples, and then we generalize our observations into any network satisfying certain conditions.

2.8.1 $K = 3$ and Min-Cut $= 1$

Let us first discuss some illustrative examples. Consider the canonical example of the extended butterfly, shown in Figure 2.5. In this case, routing can achieve rate of 1/3 since all flows go through the bottleneck link. Network coding achieves rate of 1 per flow if all side links (defined as links between S_i and R_j, $i \neq j$) are available. Let us consider the same example (all flows still go through a bottleneck link) but now only a subset of the side links is present. If only one receiver has side links, then two of the receivers have the same view, and alignment becomes impossible. If two or three receivers have side links, then depending on which links those are, network alignment may be possible. The question we are interested in is: can other approaches also achieve rate of half ?



(a) Alignment is not feasible



(b) Alignment is feasible but a butterfly is also present

Figure 2.6: Examples for 3 unicast sessions (cont.)

Figure 2.6a illustrates a case where two receivers have a side link each, but alignment is not possible; *e.g.,* one can check that the "small" conditions (Eq.(2.58)-(2.60)) are violated. Figure 2.6b illustrates another case where two receivers have a side link each. Notice that compared to the previous example, receivers 2 and 3 have switched place. Alignment is possible in this case and can achieve half the min-cut per session. However, the side links now form a butterfly substructure (between session 1 and 3). Therefore, it is possible to achieve the same rate by time-sharing between the butterfly and session 2. Figure 2.7 illustrates a case where all three receivers have a side link and there is no butterfly (for any

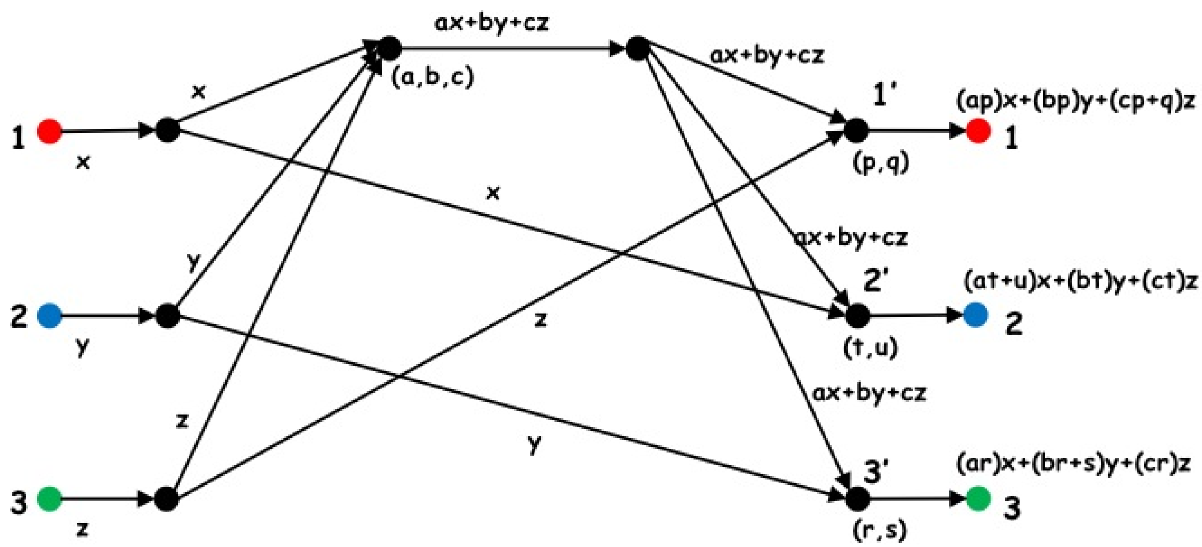


Figure 2.7: Examples for 3 unicast sessions (cont.): Alignment is feasible but coding without alignment can also achieve half the min-cut.

2 sessions) present in the network. Alignment can achieve half the min-cut, which is optimal in this case. However, it is also possible to achieve half the min-cut per session by carefully choosing the coefficients at the nodes $1', 2'$ and $3'$, so as to cancel out one component of the interference for each session. For example, node $1'$ can pick $q = -cp$ so as to cancel z and allow receiver 1 to only see equations in two unknowns (x, y), thus making it possible to solve for its own message x over two time slots.

The above intuition generalizes to a more general statement.

Theorem 2.9. *Consider a DAG with three unicast sessions, each with min-cut of 1. Whenever network alignment can achieve rate of 1/2 per session, there exists an alternative approach that can also achieve rate of 1/2 per session.*

The alternative approaches include: routing, packing butterflies, random linear network coding, or other network coding strategies that do not require the alignment strategy.[1]

[1]Notice that alignment may be achieved by coding in the middle or at the edge of the network. What distinguishes "alignment" from "non-alignment" approaches is that alignment provides the receivers with lesser number of equations than unknowns, but with aligned interference.

Proof. The common routing rate r that can be guaranteed per flow is upper bounded by the sparsity bound S. We consider two cases: $S \leq 1/2$ and $S > 1/2$. Notice that the sparsity bound S in this setup can only take one of the following values $\{1/3, 1/2, 2/3, 1\}$, provided that each session demands a rate equal to its corresponding min-cut ($= 1$).

Case I: Let us consider networks where $S = 1/3 < 1/2$. In these networks, we have $r \leq S = 1/3$. Therefore, routing cannot achieve half the min-cut and alignment needs only be compared against coding alternatives. $S = 1/3$ in this setup means that all paths of the three commodities traverse the same bottleneck edge. The intuition of the examples we discussed earlier still applies with the difference that links should be interpreted as paths from/to the sources/receivers.

In general, a network transfer function $m_{ij}(\underline{\xi})$ is just a sum of several monomials, where each monomial represents a path from source j to sink i. It is possible to express $m_{ij}(\underline{\xi}) = d_{ij}(\underline{\xi}) + c_{ij}(\underline{\xi})$, where $d_{ij}(\underline{\xi})$ is the polynomial resulting from the sub graph G' formed by removing any edge that doesn't belong to a *direct* path connecting source j to its corresponding sink j ($\forall j = 1, 2, 3$) in G; $c_{ij}(\underline{\xi})$ is the polynomial resulting from all the paths that traverse through those edges that were neglected in G'; we refer to the corresponding paths as *cross* or *side* paths. Note that $d_{ii}(\underline{\xi}) \neq 0$ and $c_{ii}(\underline{\xi}) = 0$, $i = 1, 2, 3$. Consider the matrices $D(\underline{\xi}) = [d_{ij}(\underline{\xi})]$ and $C(\underline{\xi}) = [c_{ij}(\underline{\xi})]$, and note that $M(\underline{\xi}) = [m_{ij}(\underline{\xi})] = D(\underline{\xi}) + C(\underline{\xi})$.

Because of the single bottleneck link between all direct paths in this case, $\text{rank}[D(\underline{\xi})] = 1$ and the rate achievable by routing is $1/3$ per session. Also, note that the routing rate doesn't depend on any of the $c_{ij}(\underline{\xi})$'s, but the side paths $c_{ij}(\underline{\xi})$'s do determine the feasibility of alignment. It is quite obvious that if more than one row of $C(\underline{\xi})$ is all zeros, then two or more sinks will be seeing the same perspective and it would be impossible to achieve a rate of more than $1/3$ per session. If one of the rows (say row i') alone is made of all zero elements, then we need to have $c_{i'j}(\underline{\xi}) \not\equiv 0$ and $c_{i'k}(\underline{\xi}) \not\equiv 0$ in order to satisfy the feasibility conditions (Eq.(2.58)-(2.60))

$$m_{jj}\left(\underline{\xi}\right) \not\equiv \frac{m_{kj}\left(\underline{\xi}\right) m_{ji'}\left(\underline{\xi}\right)}{m_{i'k}\left(\underline{\xi}\right)} \quad \forall\ j \neq k \neq i',$$

and this results in forming a butterfly structure between sessions j and k. An example was shown in Fig. 2.6(c): alignment is feasible but there is also a butterfly structure. In Fig. 2.6(b), the side paths do not form a butterfly but they do not satisfy the alignment feasibility conditions either: *i.e.*, $m_{11}(\underline{\xi}) \equiv (m_{12}(\underline{\xi})m_{31}(\underline{\xi}))/m_{32}(\underline{\xi})$ and $m_{22}(\underline{\xi}) \equiv (m_{21}(\underline{\xi})m_{32}(\underline{\xi}))/m_{31}(\underline{\xi})$. If $C(\underline{\xi})$ has a non-zero entry in every row, as in Fig.2.6(d), it is possible to achieve half the min-cut, without alignment, by carefully choosing the coefficients at the node where paths corresponding to the non zero $c_{ij}(\underline{\xi})$'s join with the path corresponding to $d_{ij}(\underline{\xi})$'s.

Case II: Let us consider networks where $S \geq 1/2$. If routing can achieve $r \geq 1/2$, then it's obvious that we don't require any other complex schemes let alone alignment scheme. The cases that require investigation are the ones where $r < 1/2 \leq S$; in these cases, we will describe a scheme that uses network coding (without alignment) at carefully selected nodes in the middle of the network, and guarantees rate of 1/2 for every session. In fact, it achieves this goal by using only the direct paths.

Let $\mathcal{P}_i = \{P_i^{(n)} \ : \ n = 1, 2, \ldots, N_i\}$ represent the set of all direct paths from source i to receiver i. Choose a path for each session, say $P_1^{(\alpha)} \in \mathcal{P}_1$, $P_2^{(\beta)} \in \mathcal{P}_2$ and $P_3^{(\gamma)} \in \mathcal{P}_3$ where $\alpha \in \{1, \ldots, N_1\}$, $\beta \in \{1, \ldots, N_2\}$ and $\gamma \in \{1, \ldots, N_3\}$. We are in the case where the rate achievable by routing is smaller than 1/2; this means that the three paths $P_1^{(\alpha)}$, $P_2^{(\beta)}$, $P_3^{(\gamma)}$ traverse through a bottleneck edge. But we are also in the case where $S > 1/3$; this ensures that there is another path $P' \in \mathcal{P}_1$ or $\mathcal{P}_2$ or $\mathcal{P}_3$ such that there is no bottleneck edge through which all four of the paths ($P_1^{(\alpha)}$, $P_2^{(\beta)}$, $P_3^{(\gamma)}$ and P') will traverse. Without loss of generality we can assume that $P' \in \mathcal{P}_1$. P' , $P_2^{(\beta)}$, $P_3^{(\gamma)}$ should have a bottleneck edge so that routing can't achieve half rate. Any edge that doesn't belong to any of the previously defined four paths can be ignored for the design of the scheme we are about to describe.

The scheme involves sharing the network between the sessions over time. In the first time slot, the network is used by a pair of sessions; during the second time slot the session that was left out in the first time slot gets the network resources to itself. To achieve rate of half the min-cut, we need to make sure that the two sessions, sharing the network in the first time slot, are able to decode their respective messages within that time slot. This is done by carefully choosing the coding coefficients at some nodes. To pick the pair of sessions that would be active in the first time slot we need to look at the first and last edge where all direct paths of any two sessions overlap. Such an overlap is always present because of the bottlenecks described earlier. Let edge e_f be the first edge, where all paths of sessions (i, j) overlap, and edge e_l be the final edge, where sessions (j, k) overlap at the end, then we can choose session (i, k) to be the active pair in the first time slot.

If session 1 is one of the active sessions in the first time slot, then we know that there are two paths $P_1^{(\alpha)}$ and P' joining at some node after e_l. We can choose the coefficients at the tail node of the edge e_l such that the message component of session 1 is eliminated and only the component of the other session is present at the head of the edge e_l. Thus the sink of the other session gets its message without any interference and session 1 can still decode its message by using the component of the other session, from the head of edge e_l, as an antidote at the edge where the two paths of session 1 meet. If session 1 is not among the pair of active sessions in the first time slot, then we can use the same idea to cancel out the interference before delivering the messages to the respective sinks of the sessions. But here, we choose the coefficients at the tail node of edge e_l to cancel out the interference component of session i and the coefficients at the tail node of the edge right before e_l where the paths of session i and session k overlap. This ensures that each session that is active in the first time slot gets its corresponding message delivered within that time slot. This scheme succeeds in

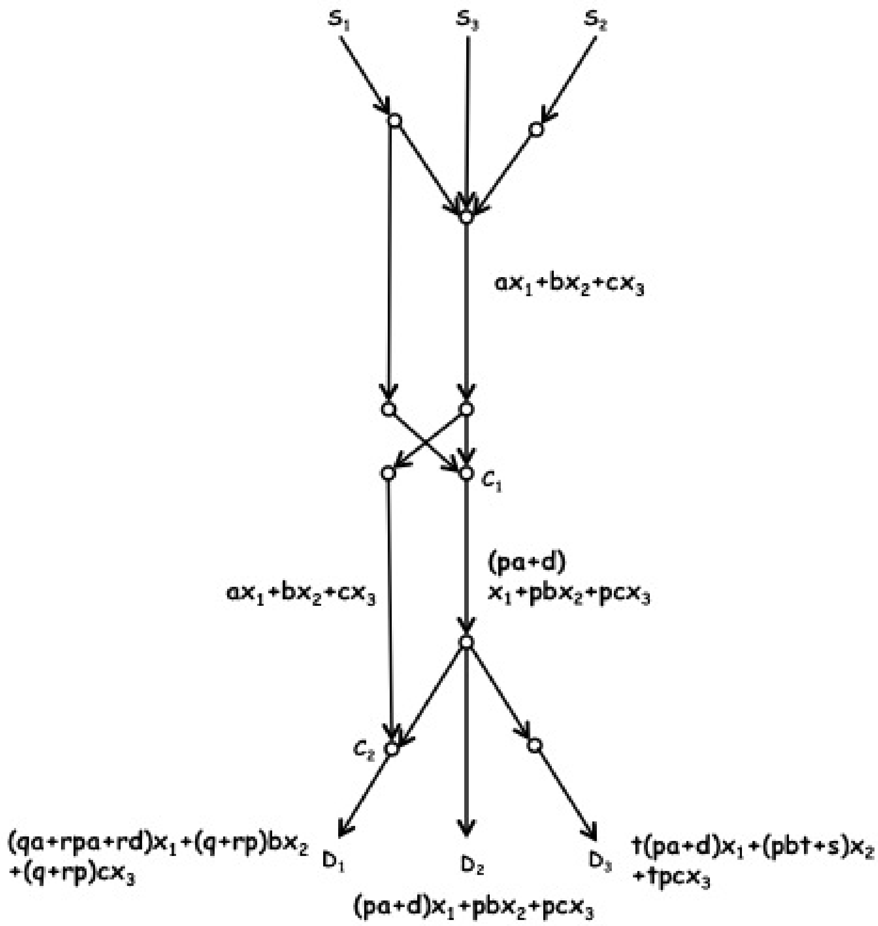


Figure 2.8: Example topology with $r < 1/2$ and $S = 1/2$

achieving rate of half-the-min-cut per session without using any alignment techniques. □

Fig. 2.8 illustrates a simple example where the achievable routing rate $r < 1/2$ and the sparsity bound $S > 1/3$. In this example the value of sparsity bound can be easily determined as $1/2$ and we can see that routing cannot achieve any rate more than $1/3$ because of the overlapping paths. But it is still possible to achieve a rate of $1/2$ by making use of the scheme suggested in the proof above. According to the scheme, we can choose session 1 and session 2 to be active in the first channel use, and during this period we can also choose the coding coefficients node C_1 and C_2, namely, by choosing $d = -pa$ and $q = -rp$, we could get rid of the interfering components at destination D_1 and D_2 respectively. For the second

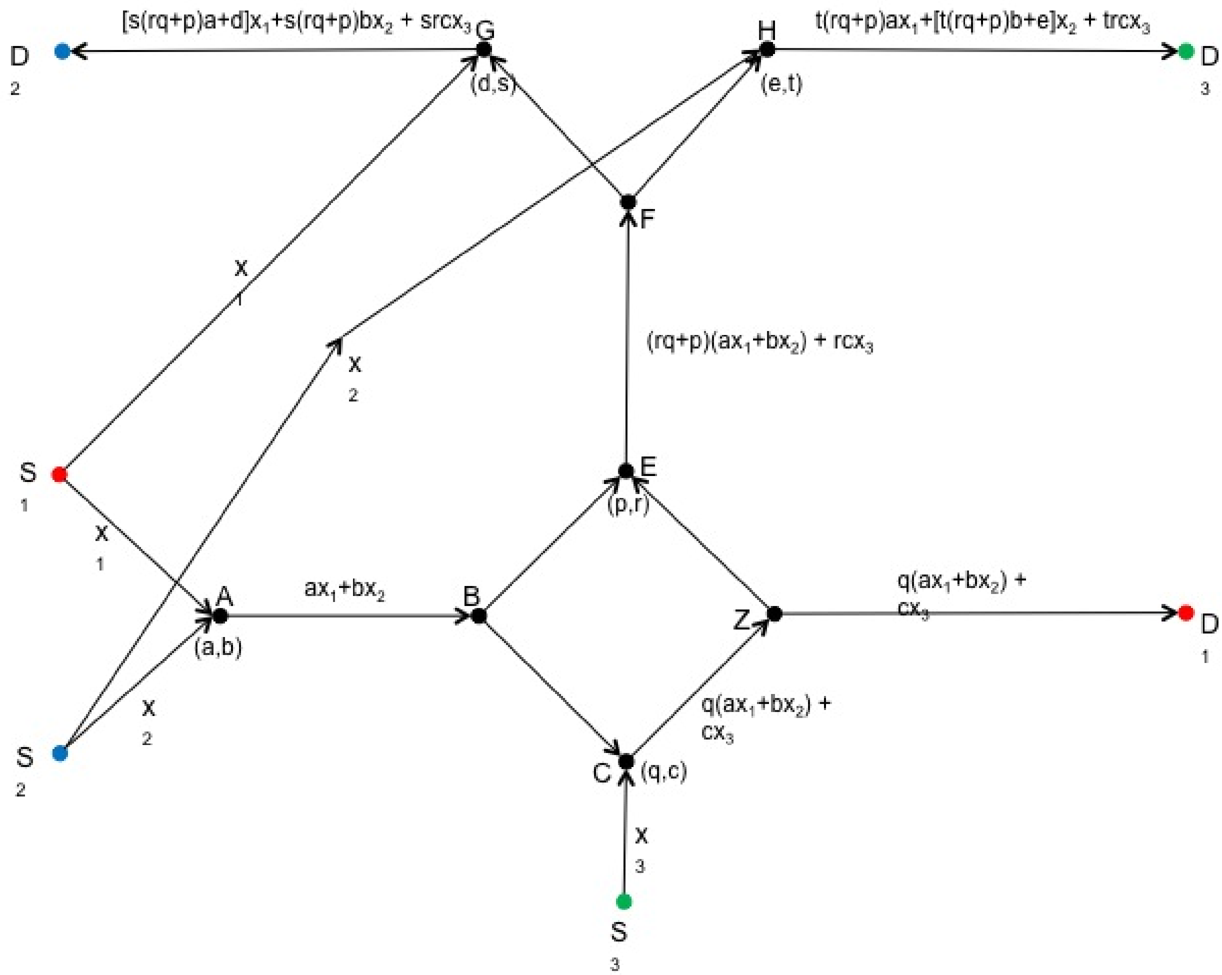


Figure 2.9: Example topology with $r = 1/2$

channel use, we can make session 1 and 2 to be idle, there by giving all the network resources to session 3, thus helping each session to send a single message over two channel uses (rate $= 1/2$).

Fig. 2.9 illustrates a network topology where the desired rate of 1/2 per user is easily achieved with routing alone. Session 1 uses the path $S_1 - A - B - C - Z - D_1$, session 2 uses the path $S_2 - A - B - E - F - G - D_2$ and session 3 uses the path $S_3 - C - Z - E - H - D_3$. At edge AB, session 1 and 2 compete for resource; at edge CZ session 2 and 3 compete for resources; and at edge EF session 2 and 3 compete for resources. The competition for resource at any edge is at maximum between two users only. This is ensured by choosing

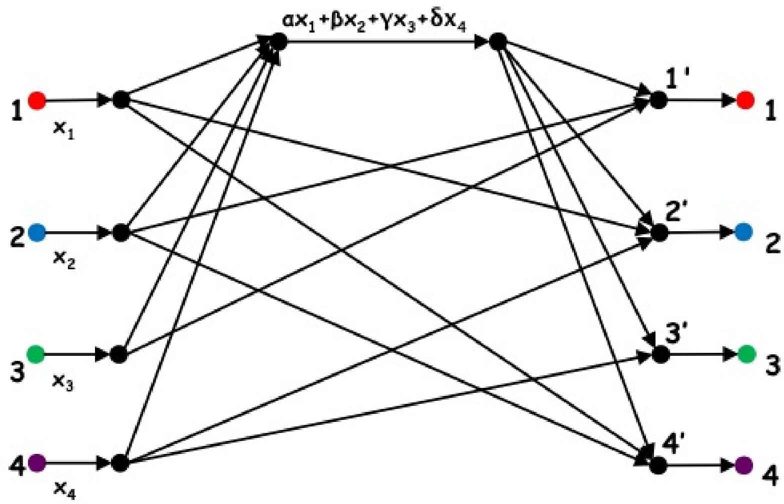


Figure 2.10: Examples that require alignment, $K = 4$, min-cut= 1.

$S_2 - A - B - E - F - G - D_2$ as the path for session 2 when it also had the option to use $S_2 - A - B - C - Z - E - F - G - D_2$. Now by limiting the flow of each session to 1/2, we can make sure that all the session are able to transmit and receive their flows withoiut any problem.

2.8.2 $K > 3$ or Min-Cut > 1

Figures 2.10 to 2.12 show examples where the number of sessions is $K > 3$ or the min-cut is greater than one. In all these cases alignment is required in the following sense: the maximum rate is 1/2 the min-cut, alignment achieves it and no other baseline scheme (packing butterflies or coding without alignment) can achieve it. Indeed, in Fig. 2.10, receiver 3 has 2 equations with 4 unknowns over 2 time slots (Notice that the side links convey the redundant information over 2 time slots). Even if the sink nodes are considered to be present at $1', 2', 3'$ and $4'$, we can observe that over 2 time slots node $3'$ receives only 3 independent equations with 4 unknowns. So it needs an alignment technique to decode its message. This effect is amplified for 5 sessions example in Fig. 2.11, where the nodes $1', 2', 3'$ and $5'$ effectively get 4 equations in 5 unknowns over two time slots. In Fig. 2.12 it can be

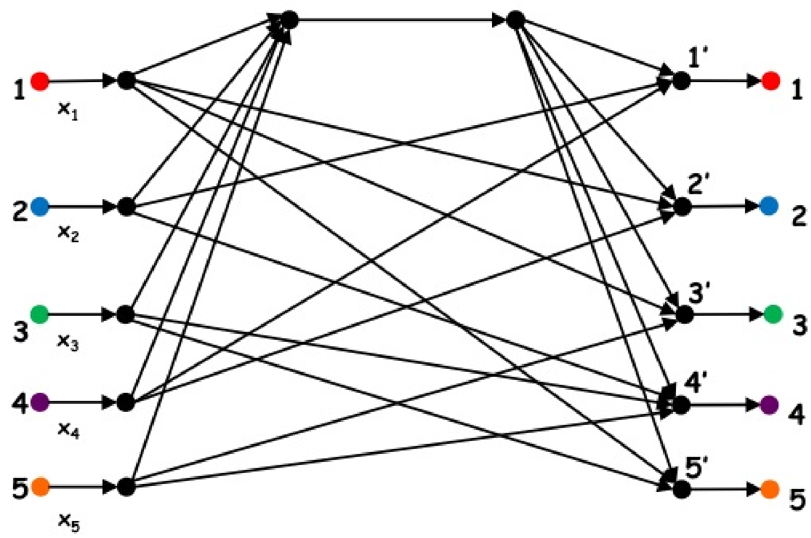


Figure 2.11: Examples that require alignment, $K = 5$, min-cut= 1.

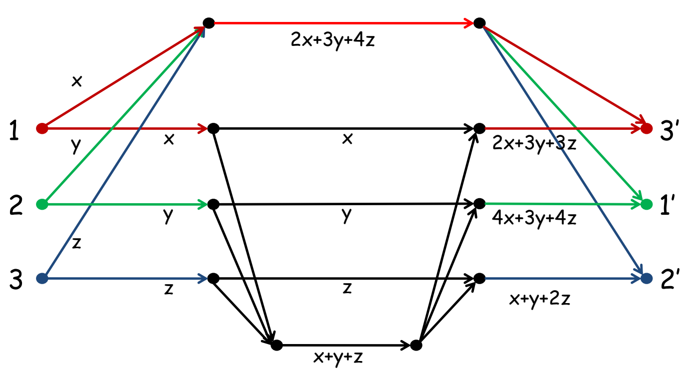


Figure 2.12: Examples that require alignment: $K = 3$, min-cut= 2

observed that the equations carried by the incoming edges of node D_i, $\forall i = 1, 2, 3$, needs to have their interference component aligned in order for the receiver to be able to decode its corresponding message.

2.8.3 PBNA vs. Routing

In this section, we present a comparison of rate achievable through PBNA and the rate achievable through routing. We would like to point out that such a comparison is unfair

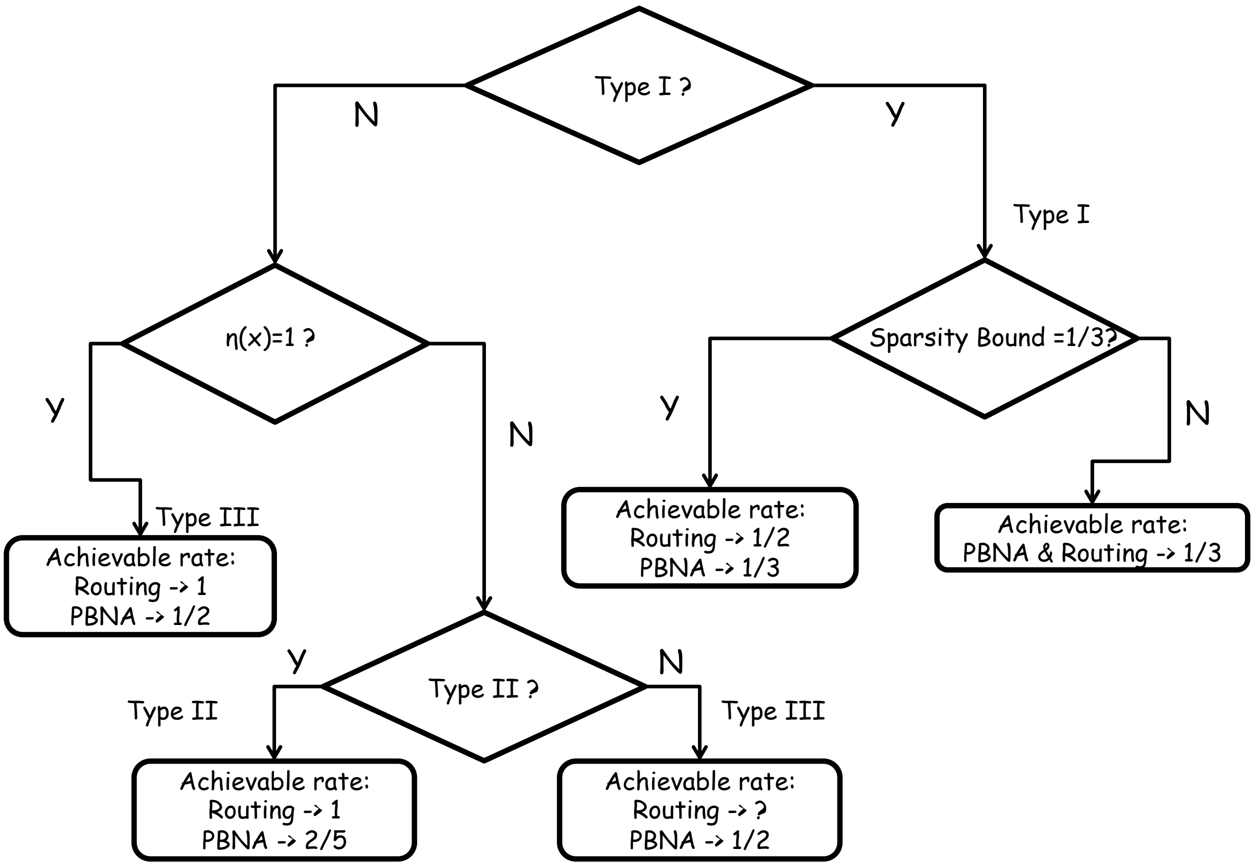


Figure 2.13: A comparison between PBNA and routing in terms of achievable symmetric rate for various types of networks.

to PBNA because of the network setting considered. Routing involves intelligence inside the network, whereas in our setting, the internal nodes have no intelligence and can only perform random linear network coding. Therefore, routing by definition is not included in our network setting. The structure of the network determines the rate achievable by any scheme, depending on the structure one scheme can perform better than the other. In Fig. 2.13, we provide a taxonomy of the network based on its structures and the rates achievable through routing and PBNA.

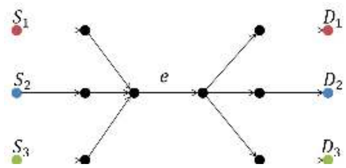
Type III networks are the main focus of this chapter and it is hard to characterize the rate achievable through routing for these networks which make PBNA more interesting as it can systematically achieve a rate of $\frac{1}{2}$ per user under this network type. The following points can be noted from Fig. 2.13:

- *Type I* networks can be further classified into two cases based on the sparsity bound. When the sparsity bound equals $\frac{1}{3}$, both PBNA and routing can only achieve a symmetric rate of $\frac{1}{3}$ per user. An example of such network is shown in Fig. 2.14a. When the sparsity bound is greater than $\frac{1}{3}$, routing can achieve a symmetric rate of $\frac{1}{2}$ per user. However, PBNA can only achieve a symmetric rate of $\frac{1}{3}$ per user, which is the optimal symmetric rate achieved by any precoding-based linear schemes. Fig. 2.14b illustrates such an example.

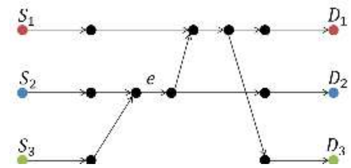
- Type II networks, due to the presence of the coupling relations, $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$ or $p_2(\mathbf{x}) = 1 + \eta(\mathbf{x})$ or $p_3(\mathbf{x}) = 1 + \eta(\mathbf{x})$, will have a network structure where each source has a disjoint path to its corresponding receiver making it possible to achieve a rate of 1 per user with routing. In contrast, PBNA can only achieve a symmetric rate of $\frac{2}{5}$ for these networks. An example of such a network is shown in Fig. 2.14c.

- For Type III networks, PBNA can achieve a symmetric rate of $\frac{1}{2}$. Consider the special case of $\eta(\mathbf{x}) = 1$, it can be shown (see 2.8.4) that these networks can be characterized by the presence of disjoint paths from each source to its corresponding receiver and routing can always achieve a symmetric rate of 1 per user here. An example of such network is shown in Fig. 2.14f. For the less constrained case of $\eta(\mathbf{x}) \neq 1$, however, it is not straightforward to characterize the the performance of routing. We can see that there are networks in which routing can only achieve a symmetric rate of $\frac{1}{3}$ (see Fig. 2.14d); and there are also networks where routing can achieve a symmetric rate of one due to the rich connectivity in the network (see Fig. 2.14e).

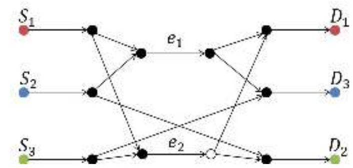
  Type III networks, especially the ones with $\eta(\mathbf{x}) \neq 1$, are our main focus and it is hard to characterize the rate achievable through routing for these networks. This makes the case for PBNA more interesting, as PBNA provides a systematic approach for achieving a rate of $\frac{1}{2}$ per user for these networks.



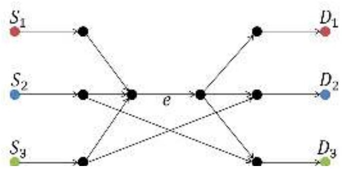
(a) A type I network (Sparsity bound= $\frac{1}{3}$)



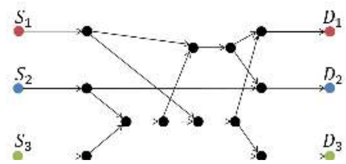
(b) A type I network (Sparsity bound> $\frac{1}{3}$)



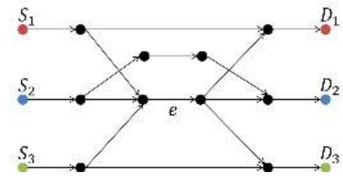
(c) A type II network ($p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{\eta(\mathbf{x})+1}$)



(d) A type III network ($\eta(\mathbf{x}) \neq 1$)



(e) A type III network ($\eta(\mathbf{x}) \neq 1$)



(f) A type III network ($\eta(\mathbf{x}) = 1$)

Figure 2.14: Example networks. (a) shows a Type I network, for which PBNA and routing both achieve symmetric rate $\frac{1}{3}$. (b) shows another Type I network, for which routing can achieve symmetric rate $\frac{1}{2}$, and PBNA can only achieve symmetric rate $\frac{1}{3}$. (c) shows a Type II network, for which routing can achieve symmetric rate one, and PBNA can only achieve symmetric rate $\frac{2}{5}$. (d) shows a Type III network, for which routing can only achieve symmetric rate $\frac{1}{3}$, and PBNA can achieve symmetric rate $\frac{1}{2}$. In (e), we show another Type III network, for which routing can achieve symmetric rate one, and PBNA can only achieve symmetric rate $\frac{1}{2}$. (f) shows a Type III network, for which routing can always achieve symmetric rate one, and PBNA can only achieve symmetric rate $\frac{1}{2}$.

2.8.4 Characterizing the Routing Rate for *Type III* Networks with $\eta(\mathbf{x}) = 1$

In this subsection, we prove that for Type III networks with $\eta(\mathbf{x}) = 1$, routing can always achieve a symmetric rate of one.

We will first define the following polynomials:

$$L(\mathbf{x}) = m_{13}(\mathbf{x})m_{32}(\mathbf{x})m_{21}(\mathbf{x}) \quad R(\mathbf{x}) = m_{12}(\mathbf{x})m_{23}(\mathbf{x})m_{31}(\mathbf{x})$$

Thus, $\eta(\mathbf{x}) = \frac{L(\mathbf{x})}{R(\mathbf{x})}$. Given two distinct edges/nodes e_1, e_2, if there exists a directed path from e_1 to e_2, we say e_1 is upstream of e_2 (or e_1 is downstream of e_2), and denote this relation by $e_1 \prec e_2$. Similarly, $e_1 \not\prec v_2$ implies that there is no directed path from e_1 to e_2.

Given two subsets of nodes $S, D \subseteq V$, let $EC(S; D)$ denote the minimum capacity of all the edge cuts separating S from D. Define the following subsets of edges:

$$\bar{S}_i \triangleq \{e \in E - \{\sigma_i\} \ : \ e \in C_{ij} \cap C_{ik}, j \neq k, j, k \in \{1,2,3\} - \{i\}\}$$
$$\bar{D}_i \triangleq \{e \in E - \{\tau_i\} \ : \ e \in C_{jj} \cap C_{kj}, j \neq k, j, k \in \{1,2,3\} - \{i\}\}$$

The following proposition was stated in [49], which gives a graph theoretic interpretation of the condition $\eta(\mathbf{x}) = 1$.

Proposition 2.10. *$L(\mathbf{x}) \equiv R(\mathbf{x})$ if and only if there exists two distinct integers $i, j \in \{1,2,3\}$ such that $\bar{S}_i \cap \bar{S}_j \neq \emptyset$ and $\bar{D}_i \cap \bar{D}_j \neq \emptyset$.*

Lemma 2.11. *Let i, j be two distinct integers in $\{1,2,3\}$, and $e_2 \in \bar{D}_i \cap \bar{D}_j$. If $\bar{S}_i \cap \bar{S}_j \neq \emptyset$, then there exists $e_1 \in \bar{S}_i \cap \bar{S}_j$ such that $e_1 \prec e_2$ or $e_1 = e_2$.*

Proof. Same as *lemma* 5 in [49] . □

Lemma 2.12. *For a given $i, j, k \in \{1,2,3\}$ and $i \neq j \neq k$, if $\bar{S}_i \cap \bar{S}_j \neq \emptyset$; $\bar{D}_i \cap \bar{D}_j \neq \emptyset$ and $EC(\{s_i, s_j\}; \{d_i, d_k\}) > 1$, then there exists a path P'_{ii} from s_i to d_i such that for each $e' \in P'_{ii}$, $s_j \not\prec e'$, $s_k \not\prec e'$, and $e' \not\prec d_j$, $e' \not\prec d_k$.*

Proof. Without loss of generality, suppose $i = 1$, $j = 2$ and $k = 3$. We can choose two edges $e_1 \in \bar{S}_1 \cap \bar{S}_2$ and $e_2 \in \bar{D}_1 \cap \bar{D}_2$ such that $e_1 \prec e_2$ or $e_1 = e_2$ (from *lemma* 1). Now consider the edge e_1, by definition cutting this edge would cut the flows $s_1 \to d_2$, $s_1 \to d_3$, $s_2 \to d_1$ and $s_2 \to d_3$. Since we also have $EC(\{s_i, s_j\}; \{d_i, d_k\}) > 1$, we can see that there should exist a path P'_{11} such that $e_1 \notin P'_{11}$. Consider any edge $e' \in P_{11}$,

- If this edge e' has d_2 (or d_3) as a *downstream* node, then there will exist a path P_{12}

(or P_{13}) such that $e_1 \notin P_{12}$ (or $e_1 \notin P_{13}$), which contradicts the definition of edge e_1 (or e_2). Thus $e' \not\prec d_2$, $e' \not\prec d_3$.

- Similarly, if edge e' is *downstream* of s_2, it would result in a path P_{21} such that $e_1 \notin P_{21}$, which again will contradict the definition of e_1. Thus $s_2 \not\prec e'$.

- If edge e' is *downstream* of s_3, it would result in a path P_{31}, where $e_1 \notin P_{31}$. But by definition of e_2, $e_2 \in P_{31}$, this in turn would result in paths P'_{12} and P'_{13} that does not go through edge e_1. Thus $s_3 \not\prec e'$.

$\square$

Theorem 2.13. *Assume that all the senders are connected to all the receivers via directed paths. If $\eta(\mathbf{x}) = 1$ and $p_i(\mathbf{x}) \neq 1$ for $1 \leq i \leq 3$, then routing can achieve the rate tuple* $(1, 1, 1)$.

Proof. Without loss of generality, suppose $i = 1$, $j = 2$, $k = 3$ and $\bar{S}_1 \cap \bar{S}_2 \neq \emptyset$; $\bar{D}_1 \cap \bar{D}_2 \neq \emptyset$. From Lemma 2.12, we can see that there exist two disjoint paths, $P_1 \in P_{11}$ and $P_2 \in P_{22}$. Therefore, ω_1 and ω_2 can transmit one unit flow through P_1 and P_2 respectively. Meanwhile, ω_3 can route one unit flow through the rest of the network. This implies that routing can achieve the rate tuple $(1, 1, 1)$. $\square$

2.9 Summary

In this chapter, we consider the problem of network coding for the SISO scenarios with three unicast sessions. We described two general approaches, *i.e.,* coding at the edge or in the middle of the network, and two specific methods of the first approach, *i.e.,* the eigenvector method and the symbol extension method originally introduced in [11,44]. We discussed the feasibility conditions for each of these schemes and their relation to network structure. We show that network topology may introduce algebraic dependence between different transfer functions, which can potentially affect the rate achieved by PBNA. We identify the minimal set of conditions that are realizable in networks. Moreover, we show that each of these conditions has a unique interpretation in terms of network topology. We also derived the optimal rates achievable under any precoding based linear scheme for various network types and showed that these rates are indeed achievable with PBNA (with small modifications in some cases).

We also compared alignment to alternative approaches that can achieve half the rate depending on the network topology. For three unicast sessions with min-cut one, we show a negative result: whenever alignment is possible, alternative approaches can also achieve half the min-cut. However, for more than three sessions and/or for min-cut per session greater than one, we show examples where alignment is necessary. We also provide a classification of the network based on the network structure and compare the rates achievable by PBNA to routing.

This work is limited to three unicast sessions in the SISO scenario (i.e., with min-cut one per session) and following a precoding-based approach (all precoding is performed at the end nodes, while intermediate nodes perform random network coding). This is the simplest, yet highly non-trivial instance of the general problem of network coding across multiple unicasts. Apart from being of interest on its own right, we hope that it can be used as a building block and provide insight into the general problem.

Chapter 3

Networks with Rank Deficient Transfer Matrix

After thoroughly analyzing and successfully characterizing the precoding-based linear rates for the networks with three session in the previous chapter, we move on to more than three sessions ($K > 3$). Alignment techniques become more complex as we increase the number of sessions. In this chapter, we focus on certain sub-classes of network topologies, namely we look at wireless interference channels with a rank deficient transfer matrix. In the wired case, due to the simplicity of the topology considered, it is possible to design the network code to achieve desirable rates. But in the wireless setting, the channel coefficients are inherently random and we have to make use of precoding based scheme, so it makes more sense to study these topologies from the wireless perspective.

3.1 Introduction

Wireless networks are interference limited and understanding the degrees of freedom (DoF) of various interference networks is a significant problem in network information theory. Optimal DoF results are available for several K-user MIMO interference channels using the principle of Interference Alignment [11, 51–53]. When all transmitters and receivers have M nodes with full rank channel matrices, it is known that $\frac{KM}{2}$ DoF are achievable using the CJ08 asymptotic alignment scheme [11], when channel coefficients are time-varying and drawn from a continuous distribution.

The DoF of rank deficient MIMO interference channels have been studied in [54–57]. All these prior works consider individual channels between a transmitter-receiver pair to be rank deficient. Such rank deficient channels are frequently encountered in wireless MIMO networks due to poor scattering and keyhole effects. This chapter considers the overall transfer matrix of the network to be rank deficient, which has not been explored before. Rank deficient transfer matrices are observed typically in wired and wireless networks with constraints in the network topology. For example, such rank deficient transfer matrices could manifest in relay networks, wherein all the intelligence resides only at the transmitters. Rank deficiency in the transfer matrix leads to spatial dependencies between the direct and cross channels, implications of which will be discussed in this work.

The DoF of 2-user SISO interference channel with such rank deficiencies are known trivially, while those for 3-user SISO interference channel follows from [48, 58, 59]. The use of interference alignment for the 3 multiple unicast network coding problem was initially discussed in [58] and [59]. Later, Meng et al. derived the feasibility conditions for asymptotic interference alignment, in [48]. Rank deficiency in X channels was discussed in [60], with the individual channels being rank deficient. Spatial dependencies have also been observed in interference channels with coordinated multipoint (CoMP) transmission and reception, the

DoF of which were explored in [61].

In this chapter, we introduce the problem of characterizing the DoF of K-user SISO interference channels with transfer matrix of rank D ($D < K$). We present a modified version of the asymptotic interference alignment [CJ08] scheme to handle the spatial dependencies that arise due to the rank deficiency. A set of polynomial conditions are derived which are shown to be sufficient for achieving half rate per user using this modified scheme. We analyze the 4-user and 5-user interference channels with rank D, and point out the difficulty in proving the sufficient conditions here. We then study the 6-user interference channel where we show that the sufficient conditions are not satisfied, thereby pointing out the challenges in showing achievability for $K \geq 6$.

3.2 System Model

We consider the K-user SISO interference channel with perfect global channel knowledge. The channel output at the k-th receiver over the t-th channel use is given as,

$$Y_k(t) = \sum_{j=1}^{K} H_{kj}(t) X_j(t) + Z_k(t)$$

where, $k \in \{1, 2, \ldots, K\}$ is the user index, $t \in \mathbb{N}$ is the channel use index, $Y_k(t)$ is the output signal of the k-th receiver, $X_k(t)$ is the input signal of the k-th transmitter, $H_{kj}(t)$ is the channel coefficient from transmitter j to receiver k over the t-th channel use, and $Z_k(t)$ is the AWGN at the k-th receiver. The bold face notations $\mathbf{X}_k, \mathbf{Y}_k$, and $\mathbf{Z}_k$ are used to represent the vector form of their corresponding scalars over multiple channel uses, and the bold face notation $\mathbf{H}_{ij}$ is used to represent the diagonal channel matrix over multiple channel uses. For any given time slot $t \in \mathbb{N}$, the overall transfer matrix is defined as the $K \times K$ matrix of the form $\mathbf{H}(t) = [H_{ij}(t)]\ \forall i, j \in \{1, 2, \ldots, K\}$, and its rank is given by D. Time indices are

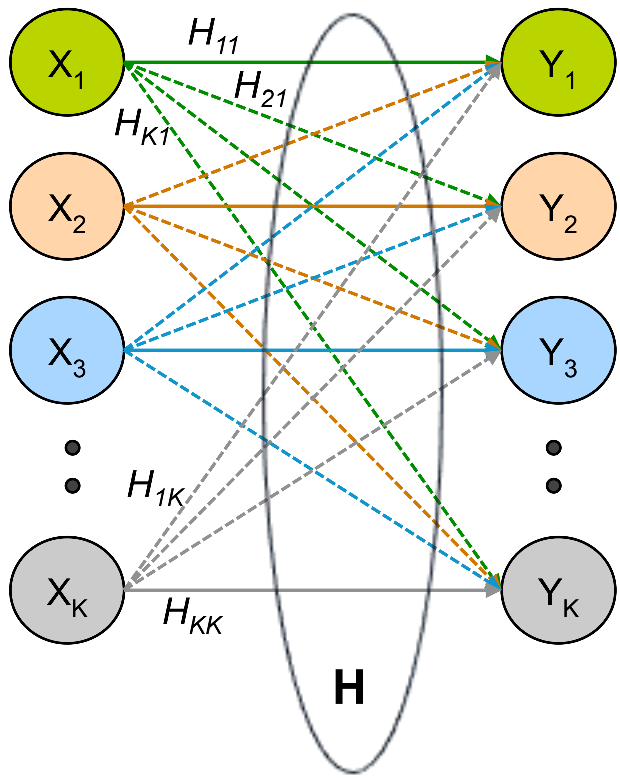


Figure 3.1: System Model

omitted for brevity.

Let $R_k(\rho)$ denote the achievable rate of user k where ρ is the Signal-to-Noise Ratio (SNR). The capacity region $C(\rho)$ of this network is the set of achievable rate tuples $R(\rho) = (R_1(\rho), R_2(\rho), \ldots R_K(\rho))$ such that each user can simultaneously decode its desired message with arbitrarily small error probability. The maximum sum rate of this channel is defined as $R_\Sigma(\rho) = max_{R(\rho)\in C(\rho)} \sum_{k=1}^{K} R_k(\rho)$. The sum DoF is defined as $d_\Sigma = lim_{\rho\rightarrow \inf} \frac{R_\Sigma(\rho)}{log(\rho)}$ and $\frac{d_\Sigma}{K}$ as the normalized DoF per user.

3.3 Overview Of Results

For the K-user SISO interference channel with rank deficient transfer matrix ($D < K$), we show that the outer bound of the sum degrees of freedom is : $d_\Sigma \leq \min\{D, \frac{K}{2}\}$.

The rank deficiency in the transfer matrix creates algebraic dependencies even among the cross channel coefficients making it hard to apply the [CJ08] alignment scheme directly. We introduce a modified version of the [CJ08] scheme to deal with these dependencies, and then derive the sufficient conditions under which there will be no overlap between the desired and interfering signal spaces. This scheme is also used in [62] to show achievability results for individual channel rank deficiency of MIMO interference channels.

Let $\mathbb{S}$ denote the set of channel realizations with rank D.

Theorem 3.1. *Degrees of freedom achievable for the K-user interference channel with rank deficient transfer matrix, can be made arbitrarily close to half per user using the modified alignment scheme, if for each $k \in \{1, \ldots, K\}$: $QH_{kk} - P \neq 0 \ \forall \ P, Q$; where P and Q are multivariate polynomials in the variables $\{H_{ij} : i \neq j\}$ and non-zero under $\mathbb{S}$.*

This theorem signifies that half rate per user is achievable even with algebraic dependencies among the channels, provided the dependencies between the direct and cross channels can not be expressed in the form defined above.

We then check if these conditions hold true for the general K-user case. We discuss a simple approach that uses ergodic alignment [41] ideas to get a subspace of realizations which in turn would help us prove some, but not all, of these sufficient conditions.

We present our analysis for $K = 4, 5$ and explain the difficulties of proving the sufficient conditions from Theorem 3.1. We also study the 6-user channel, and show that the sufficient conditions are not satisfied.

3.4 Mathematical Preliminaries

In the appendix, we use some results in algebraic geometry, so we start by recalling some basic terminology in algebraic geometry. We refer the reader to the book [63] for an excellent introduction.

3.4.1 Varities and Ideals

Let $\mathbb{C}[t_1, t_2, \cdots, t_n]$ and $\mathbb{C}(t_1, t_2, \cdots, t_n)$ denote the set of multivariate polynomials and rational functions, respectively, in the variables $t_1, t_2, \cdots, t_n$. For any polynomials $f_1, f_2, \cdots, f_m \in \mathbb{C}[t_1, t_2, \cdots, t_n]$, the *affine variety* generated by $f_1, f_2, \cdots, f_m$ is defined as set of points at which the polynomials vanish:

$$V(\mathbf{f}) = \{\mathbf{t} \in \mathbb{C}^n : \mathbf{f}(\mathbf{t}) = \mathbf{0}\}.$$

Any subset $I \subseteq \mathbb{C}[t_1, t_2, \cdots, t_n]$ is called an *ideal* if it satisfies the three properties

- $0 \in I$.
- If $f_1, f_2 \in I$, then $f_1 + f_2 \in I$.
- If $f_1 \in I$ and $f_2 \in \mathbb{C}[t_1, t_2, \cdots, t_n]$, then $f_1 f_2 \in I$.

For any set $\mathcal{A} \subseteq \mathbb{C}^n$, the ideal generated by $\mathcal{A}$ is defined as

$$I(\mathcal{A}) = \{f \in \mathbb{C}[t_1, t_2, \cdots, t_n] : f(\mathbf{t}) = 0 \ \forall \mathbf{t} \in \mathcal{A}\}.$$

For any ideal I, the affine variety generated by I is defined as

$$V(I) = \{\mathbf{t} \in \mathbb{C}^n : f(\mathbf{t}) = 0 \ \forall f \in I\}.$$

3.4.2 Algebraic Independence and Jacobian Criterion

Definition 3.1. *The rational functions* $f_1, f_2 \cdots, f_m \in \mathbb{C}(t_1, t_2, \cdots, t_n)$ *are called algebraically dependent (over* $\mathbb{C}$*) if there exists a nonzero polynomial* $F \in \mathbb{C}[s_1, s_2 \cdots, s_m]$ *such that* $F(f_1, f_2, \cdots, f_m) = 0$. *If there exists no such annihilating polynomial* F, *then* $f_1, f_2, \cdots, f_m$ *are algebraically independent.*

Lemma 3.2 (Theorem 3 on page 135 of [64])**.** *The rational functions* $f_1, f_2 \cdots, f_m \in \mathbb{C}(t_1, t_2, \cdots, t_n)$ *are algebraically independent if and only if the Jacobian matrix*

$$\mathbf{J}_f = \left(\frac{\partial f_i}{\partial t_j}\right)_{1\le i\le m, 1\le j\le n} \tag{3.1}$$

has full row rank equal to m.

The Jacobian matrix is a function of the variables $t_1, t_2, \cdots, t_n$, and hence the Jacobian matrix can have different ranks at different points $\mathbf{t} \in \mathbb{C}^n$. The above lemma refers to the *structural rank* of the Jacobian matrix which is equal to m if and only if there exists at least one realization $\mathbf{t} \in \mathbb{C}^n$ where the Jacobian matrix has full row rank.

3.5 Preliminary Analysis

Lemma 3.3. *For the* K*-user SISO interference channel with transfer matrix* $\mathbf{H}$ *of rank* D, *the sum DoF* d_Σ *is bounded from above by* $\min\{D, K/2\}$, *i.e.* $d_\Sigma \le \min\left\{D, \frac{K}{2}\right\}$

Proof. We know that for a generic K-user interference channel the outer bound for the sum DoF is given by $\frac{K}{2}$ [11]. This bound also holds for the rank deficient channel considered in this chapter, giving the outer bound of $\frac{K}{2}$ when the rank $D \ge \frac{K}{2}$. The rank of the transfer matrix, D, acts as the cutset bound, *i.e.*, no more than D independent data streams can be transmitted over this network. Hence we get the final outer bound for the sum DoF as, $d_\Sigma \leq \min\{D, \frac{K}{2}\}$. $\square$

The outer bound depends on the value of D, but in our analysis we will focus on the setting where $D = \lceil \frac{K}{2} \rceil$. The rank $D = \lceil \frac{K}{2} \rceil$ is the most interesting setting because, if we can prove the achievability for this, the result extends easily to all other values of D. A more rigorous discussion about this can be found in Appendix B.1.

Consider the determinant of any $l \times l$ sub matrix of the network transfer matrix $\mathbf{H}$, where $l > D$, it gives a polynomial in H_{ij} which identically equates to 0. This implies that the channel coefficients (H_{ij}'s) are algebraically dependent. In a generic interference channel the cross channels are all algebraically independent, so the precoding matrix used in [CJ08] scheme is almost surely full rank. But in a rank deficient interference channel, especially in the case where the rank is $\lceil \frac{K}{2} \rceil$, the cross channels might be algebraically dependent thus making the precoding matrix rank deficient too. We will modify the [CJ08] scheme to exclude the linearly dependent columns in the precoding matrix, which reduces the number of dimensions of the desired and interfering signal spaces at the receivers. We will explore if there is overlap between the desired and the interfering signal spaces, and consequently the achievability of half rate per user.

3.5.1 The Modified Scheme

Consider the asymptotic interference alignment scheme for K-user Gaussian SISO interference channel (CJ08) as described in [11, 65]. The symbol extended version of the receiver equation is given below,

$$\mathbf{Y}_j = \sum_{i=1}^{K} \mathbf{H}_{ji}\mathbf{X}_i + \mathbf{Z}_i, \forall j \in \{1, \ldots, K\} \tag{3.2}$$

Let us denote the precoding matrix used at each transmitter in the [CJ08] scheme as V_n for some arbitrarily large n ,

$$V_n = \{(T_1)^{\alpha_1}(T_2)^{\alpha_2}\dots(T_N)^{\alpha_N}\mathbf{1} \quad | \\ \sum_{i=1}^{N}\alpha_i \leq n, \quad \alpha_1, \alpha_2, \dots, \alpha_N \in \{0\} \cup Z_+\} \tag{3.3}$$

$$\mathcal{I}_n = \{(T_1)^{\alpha_1}(T_2)^{\alpha_2}\dots(T_N)^{\alpha_N}\mathbf{1} \quad | \\ \sum_{i=1}^{N}\alpha_i \leq n+1, \quad \alpha_1, \alpha_2, \dots, \alpha_N \in \{0\} \cup Z_+\} \tag{3.4}$$

wherein $T_1, \dots, T_N$ are the $N = K(K-1)$ cross channels $\mathbf{H}_{ji}, j \neq i$ and $\mathbf{1}$ refers to the all one column vector. We can impose a lexicographic ordering on the columns V_n and $\mathcal{I}_n$. We will construct a new precoding matrix $\bar{V}_n$ by just removing linearly dependent columns of V_n. We will use $\mathcal{I}_n$ and $\bar{\mathcal{I}}_n$ to denote the original interference space and interference space with the modified scheme, at the receivers respectively. Similar to [65], all transmitters use the same set of beamforming vectors $\bar{V}_n$ and all receivers approximately see the same interfering signal space of $\bar{\mathcal{I}}_n$. It can be noted that $\mathcal{I}_n = V_{n+1}$ and $\bar{\mathcal{I}}_n = \bar{V}_{n+1}$. Since we have removed only the linearly dependent columns from V_n and $\mathcal{I}_n$ to form $\bar{V}_n$ and $\bar{\mathcal{I}}_n$, the column span of the precoding matrices remains the same, i.e. following relations hold

$$\begin{aligned} span(\bar{V}_n) &= span(V_n) \\ span(T_i\bar{V}_n) = span(T_iV_n) &\subseteq span(\mathcal{I}_n) = span(\bar{\mathcal{I}}_n) \end{aligned}$$

In the original construction, number of column vectors was given by $|V_n| = \binom{n+N}{N}$ and $|\mathcal{I}_n| = \binom{n+N+1}{N}$. While we do not precisely know the number of column vectors in $\bar{V}_n$ and $\bar{\mathcal{I}}_n$, we know that $|\bar{V}_n| < |\bar{\mathcal{I}}_n| = |\bar{V}_{n+1}|$. Now we will show that the desired signal space occupies half the dimensions at all receivers, almost surely. To this end, we need to align all the interference at every receiver within one half of the total signal space available, leaving the other half interference free for the desired signals. This will enable the receivers to decode its desired message.

We will use limit infimum in proofs for the following lemmas as limits may not exist in general for divergent series.

Lemma 3.4. *Growth rate of the new precoding vectors asymptotically reaches zero for large n,* i.e.

$$\liminf_{n\to\infty} \frac{|\bar{V}_{n+1}|}{|\bar{V}_n|} = 1 \tag{3.5}$$

Proof. We will prove this by contradiction. Suppose the contrary is true, i.e., there exists a positive number $\epsilon > 0$ such that

$$\liminf_{n\to\infty} \frac{|\bar{V}_{n+1}|}{|\bar{V}_n|} > (1+\epsilon) \tag{3.6}$$

By definition of limit infimum, (3.6) means that there exists a positive integer n_0 such that for all $n > n_0$, the below relation holds.

$$\frac{|\bar{V}_n|}{|\bar{V}_{n_0}|} > (1+\epsilon)^{n-n_0} \tag{3.7}$$

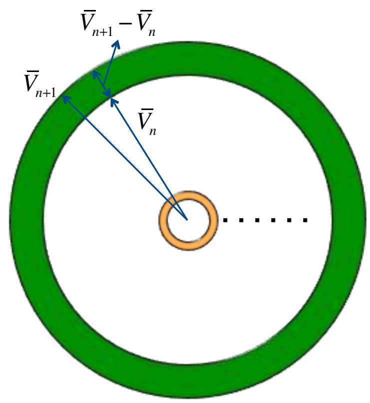


Figure 3.2: Growth rate of precoding vectors

Note that (3.7) represents a recursive relation that holds for all positive integers n, leading to :

$$|\bar{V}_n| > (1+\epsilon)^{n-n_0}|\bar{V}_{n_0}| \tag{3.8}$$

Based on the modified construction of precoding vectors for asymptotic interference alignment scheme, we know that $|\bar{V}_{n+1}| \leq \binom{n+N+1}{N}$. Hence, we have the following :

$$\frac{|\bar{V}_{n+1}|}{|\bar{V}_n|} \leq \frac{\binom{n+N+1}{N}}{(1+\epsilon)^{n-n_0}\bar{V}_{n_0}} \tag{3.9}$$

It can be seen that for large n, the term on the right side goes to zero since it is a ratio of a polynomial over an exponential in n. Note that we have assumed ϵ to be a positive number. However, we know that this cannot be true since $|\bar{V}_n| \leq |\bar{V}_{n+1}|$, leading to a contradiction. Hence the assumption in (3.6) cannot hold, and we have proved the lemma, i.e., growth rate of size of precoding matrix after removing the dependent columns, reaches zero asymptotically for large n. □

Lemma 3.5. *Given that the desired signal space $\bar{V}_n$ does not overlap with the interfering signal space $\bar{\mathcal{I}}_n$, the ratio of desired signal dimensions and total signal dimensions can be made arbitrarily close to $\frac{1}{2}$,* i.e.

$$\frac{|\bar{V}_n|}{|\bar{V}_n| + |\bar{\mathcal{I}}_n|} \approx \frac{1}{2}$$

Proof. We know from Lemma 3.4 that (3.5) holds true. Also, for a sequence x_n, if $a > \liminf x_n$, then there is an infinite subsequence x_{n_k} of x_n such that $a > x_{n_k}$. Using this we can choose a series of n and a value for δ such that

$$1 \leq \frac{|\bar{V}_{n+1}|}{|\bar{V}_n|} < 1 + \delta$$

from which we get

$$\lim_{\delta \to 0} \frac{|\bar{V}_n|}{|\bar{V}_n| + |\bar{V}_{n+1}|} = \frac{1}{2}$$

Hence with appropriate choice of δ, we can make above relation arbitrarily close to $\frac{1}{2}$, *i.e.* the ratio of desired signal dimensions and total signal dimensions reaches $\frac{1}{2}$ for large n. □

Lemma 3.4 and 3.5 imply that for the interference channel with rank deficient transfer matrix, DoF per user can be made arbitrarily close to $\frac{1}{2}$ for large n with the modified scheme, if the desired and interfering signal space do not overlap. This modified scheme has been essential in proving achievability results in [62] for the rank-deficient MIMO interference channels where the individual channels are rank-deficient.

3.5.2 The Overlap

Proof of Theorem 3.1. Let us consider the signal space at receiver 1, $S_1 = [\mathbf{H}_{11}\bar{V}_n \quad \bar{\mathcal{I}}_n]$. Matrix S_1 needs to be full rank so that desired and interference signal spaces have no overlap.

$$span(\mathbf{H}_{11}\bar{V}_n) \cap span(\bar{\mathcal{I}}_n) = \emptyset \tag{3.10}$$

Let us denote number of columns in $\bar{V}_n$ as l_v, and the number of columns in $\bar{\mathcal{I}}_n$ as l_{int} . Note that $l_v = \binom{n+N-1}{N}$ and $l_{int} = \binom{n+N}{N}$ when all cross channels are algebraically independent. Based on modified CJ08 construction scheme, the linear independence condition can be expressed as

$$\sum_{i=0}^{l_v-1} q_i \mathbf{H}_{11} \prod_m \Big(\mathbf{T}_m\Big)^{\alpha_{mi}} \neq \sum_{j=0}^{l_{int}-1} p_j \prod_m \Big(\mathbf{T}_m\Big)^{\alpha_{mj}} \tag{3.11}$$

where $m \in \{1, \ldots, K(K-1)\}$, $\alpha_{mi} \in \{0, 1, \ldots, l_v - 1\}$, $\alpha_{mj} \in \{0, 1, \ldots, l_{int} - 1\}$, and all p_i, q_j are not simultaneously zeros. Rearranging above, we get

$$\mathbf{H}_{11} \neq \frac{\sum_{j=0}^{l_{int}-1} p_j \prod_m \Big(\mathbf{T}_m\Big)^{\alpha_{mj}}}{\sum_{i=0}^{l_v-1} q_i \prod_m \Big(\mathbf{T}_m\Big)^{\alpha_{mi}}} \tag{3.12}$$

Because of the diagonal nature of $\mathbf{H}_{ij}$'s and T_m's, (3.11) can be easily translated to its scalar form. If the conditions in Theorem 3.1 are satisfied, (3.11) will hold almost surely under $\mathbb{S}$, a rigorous proof for this is presented in Appendix B.2, and consequently matrix S_1 will be

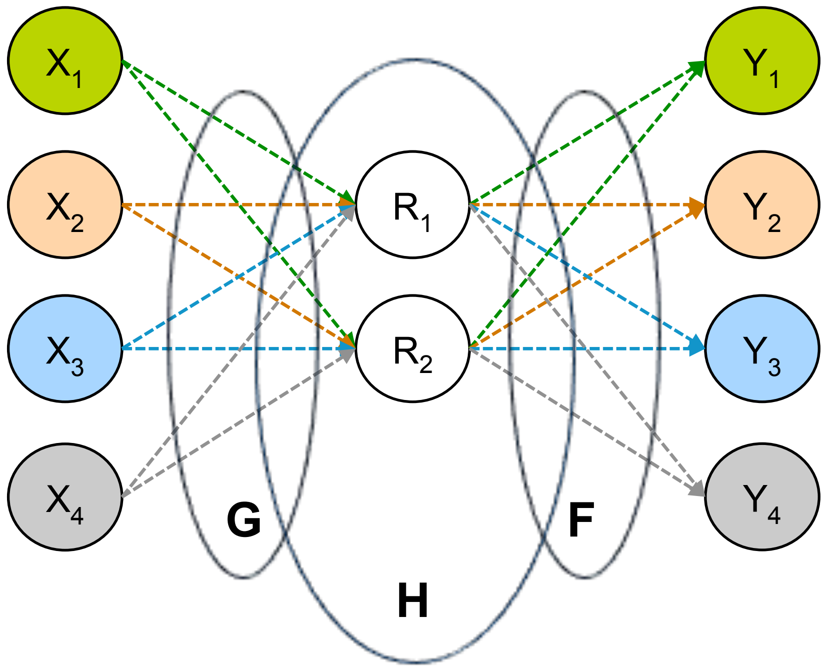


Figure 3.3: Decomposition of the 4×4 transfer matrix of rank 2.

full rank. The same argument can be extended to the other direct channels H_{kk}. This, along with Lemma 3.5, proves the theorem. $\square$

Consider the case where the rank is $\lceil \frac{K}{2} \rceil$, *i.e.* $D = \lceil \frac{K}{2} \rceil$. In order to show that the sum DoF outer bound is tight for this case, all we need to prove is that the direct channel cannot be expressed as a rational function of the cross channels.

$$H_{kk} \neq \frac{P}{Q} \quad \forall\ k = 1, \ldots, K \tag{3.13}$$

where X is the set of all cross channels and P, Q are non-zero multivariate polynomials of the cross channels. In other words, we need to show that any polynomial of the form $QH_{kk} - P$ is not identically equal to 0 under $\mathbb{S}$. We can make use of the Schwartz-Zippel lemma from polynomial identity testing for this purpose, proof of applicability of Schwartz-Zippel lemma under $\mathbb{S}$ is presented in Appendix B.2.

Consider the $K \times K$ transfer matrix $\mathbf{H}$, since the rank of this matrix is $D = \lceil \frac{K}{2} \rceil$, it's rank decomposition is the product of a $K \times D$ matrix and a $D \times K$ matrix.

$$\mathbf{H} = \mathbf{G}_{|K\times D|} * \mathbf{F}_{|D\times K|} \tag{3.14}$$

In one time slot, each receiver will see a linear equation in K variables (messages), each of which is in turn a linear combination of D linear equations. Now consider sending the same set of messages over two consecutive time slots, we will use $\mathbf{H}_1$ to represent the coefficients of the linear equations at the receivers for the first time slot and $\mathbf{H}_2$ for the second time slot. Each receiver would be able to decode its respective message if $\mathbf{H}_1 - \mathbf{H}_2 = \mathbf{I}_{|K\times K|}$, which implies,

$$\begin{aligned}
\mathbf{G}_1 * \mathbf{F}_1 - \mathbf{G}_2 * \mathbf{F}_2 &= \mathbf{I}_{|K\times K|} \\
\begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 \end{bmatrix} \begin{bmatrix} \mathbf{F}_1 \\ -\mathbf{F}_2 \end{bmatrix} &= \mathbf{I}_{|K\times K|} \\
\begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 \end{bmatrix} &= \begin{bmatrix} \mathbf{F}_1 \\ -\mathbf{F}_2 \end{bmatrix}^{-1}
\end{aligned} \tag{3.15}$$

wherein $\mathbf{G}_t, \mathbf{F}_t$ are obtained from rank decomposition of matrix $\mathbf{H}_t, t \in \{1, 2\}$. If we have the freedom to manipulate $\begin{bmatrix} \mathbf{G}_1 & \mathbf{G}_2 \end{bmatrix}$ or $\begin{bmatrix} \mathbf{F}_1 \\ -\mathbf{F}_2 \end{bmatrix}$, then by choosing one as the inverse of the other and by sending the same message over the two time slots, we would be able to zero force the interference over the two slots at each receiver. This gives us a set of realizations where the value of the cross channels would remain the same while the direct channels would vary, similar to ergodic alignment [41].

We define the subset $\mathbb{S}' \subset \mathbb{S}$ as the set of channel realizations where for each $\mathbf{H}$ in the subset there exists a complementary realization $\mathbf{H}'$ such that $\mathbf{H} - \mathbf{H}' = \mathbf{I}$, i.e.,

$$\mathbb{S}' = \{\mathbf{H} \quad | \quad \mathbf{H} \in \mathbb{S},\ \mathbf{H}' \in \mathbb{S},\ \mathbf{H} - \mathbf{H}' = \mathbf{I}\} \tag{3.16}$$

When $P \neq 0$ and $Q \neq 0$ under $\mathbb{S}'$, we get non-zero realizations for $QH_{kk} - P$, thus proving that this polynomial is non-zero under $\mathbb{S}$. The same argument could be made when only one of the two polynomials P or Q is non-zero under $\mathbb{S}'$. The problem occurs when both P and Q are zeros in $\mathbb{S}'$, in which case we can not get non-zero realizations for $QH_{kk} - P$ even under $\mathbb{S}'$, making it hard to say whether $QH_{kk} - P$ is a zero or a non-zero polynomial under $\mathbb{S}$. At this point, it is not clear whether the conditions hold for generic K.

3.6 Achievability

In this section we will first show the hurdles in proving the sufficient conditions for $K = 4$ and $K = 5$. We will also show that the sufficient conditions are not satisfied for $K \geq 6$.

3.6.1 $K = 4$ and $D = 2$

Consider the 4-user rank deficient SISO interference channel with 4 direct channels and 12 cross channels shown below,

$$\mathbf{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} \\ H_{21} & H_{22} & H_{23} & H_{24} \\ H_{31} & H_{32} & H_{33} & H_{34} \\ H_{41} & H_{42} & H_{43} & H_{44} \end{bmatrix}$$

Lemma 3.6. *All* 12 *cross channels of* 4×4 *channel matrix with rank* $D = 2$*, are algebraically independent.*

Proof. We prove this with the help of symbolic toolbox in MATLAB. Please refer Appendix B.3 for further details. $\square$

Let us denote the first direct channel H_{11} as z; the set of all cross channels as $X = \{H_{ij} : i \neq j; \forall i, j \in \{1, 2, 3, 4\}\}$, and $\mathbb{S}$ the set of all channel realizations with rank 2. Consider a 3×3 submatrix of $\mathbf{H}$ containing two direct channels, say H_{11} (denoted as z) and H_{22}. The determinant of any such submatrix is zero (since $D = 2$).

$$\begin{vmatrix} z & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{41} & H_{42} & H_{43} \end{vmatrix} = 0$$

Evaluating the determinant, we get a polynomial in z, H_{22} and the 7 cross channels. Rearranging the polynomial equation, we can express H_{22} as a rational function, $f_2(z, X)$, of z and the 7 cross channels.

$$H_{22} = \frac{H_{21}(H_{12}H_{43} - H_{42}H_{13}) + H_{23}(zH_{42} - H_{41}H_{12})}{zH_{43} - H_{41}H_{13}}$$

The denominator polynomial, $zH_{43} - H_{41}H_{13}$ is non-zero, this is shown in Appendix B.3. The denominator could still evaluate to zero for some realizations in $\mathbb{S}$ and $f_2(z, X)$ will be undefined for these realizations. But we can consider a domain $\mathbb{D}$ under which $f_2(z, X)$ is always defined, *i.e.*, set of all points in $\mathbb{S}$ for which the denominator polynomial is always non-zero. We can also see that the set of points excluded from $\mathbb{S}$ to get $\mathbb{D}$ has measure zero.

Consider the determinant of another 3×3 submatrix comprising of H_{11} and H_{22}.

$$\begin{vmatrix} z & H_{12} & H_{14} \\ H_{21} & f_2(z, X) & H_{24} \\ H_{31} & H_{32} & H_{34} \end{vmatrix} = 0$$

Evaluating the above determinant, we get a multivariate polynomial which is quadratic in z, of the form:

$$A(X)z^2 + B(X)z + C(X) = 0 \tag{3.17}$$

where $A(X), B(X), C(X)$ are all polynomial functions of the 12 cross channels. Also polynomials $A(X), B(X), C(X)$ are non-zero since the cross channels are algebraically independent.

Let us assume that there is a polynomial $Q(X)H_{11} - P(X)$ that always evaluates to zero under $\mathbb{D}$. We already know that $Q(X)$ is a non-zero polynomial, so we can express H_{11} as a rational function of the cross channels, *i.e.*, $z = H_{11} = \frac{P(X)}{Q(X)}$, which is always defined in the domain $\mathbb{D}' \subseteq \mathbb{D}$. Similar to $\mathbb{D}$, we can see that the set of points excluded from $\mathbb{S}$ to get $\mathbb{D}'$ has measure zero. Substituting this rational function for z in (3.17), we get

$$A(X)P(X)^2 + B(X)P(X)Q(X) + C(X)Q(X)^2 = 0 \tag{3.18}$$

The above equation holds if

- The polynomial in (3.18) is non-trivial and always evaluates to zero.
- $z = \frac{P(X)}{Q(X)}$ is a root of the quadratic equation (3.17).

If we suppose that the non-trivial polynomial in (3.18) always evaluates to zero, then this gives a zero polynomial in the 12 cross channels, indicating that the 12 cross channels are algebraically dependent. However this is a contradiction.

If we could show that $z = \frac{P(X)}{Q(X)}$ cannot be a root, we can establish that $Q(X)H_{11} - P(X) \neq 0$. This is hard as we do not exactly know $P(X)$ or $Q(X)$. Using MATLAB, we were able to verify that for rational realizations of the cross channels H_{ij} the roots of the equation (3.17) are not always rational. Even though this helps in showing that the polynomial $Q(X)H_{11} - P(X)$ is almost surely non-zero in the rational space, we will not be able to make the same statement for the general space $\mathcal{S}$.

3.6.2 $K = 5$ and $D = 3$

Consider the 5-user rank deficient SISO interference channel represented by the following matrix,

$$\mathbf{H} = \begin{bmatrix} H_{11} & H_{12} & H_{13} & H_{14} & H_{15} \\ H_{21} & H_{22} & H_{23} & H_{24} & H_{25} \\ H_{31} & H_{32} & H_{33} & H_{34} & H_{35} \\ H_{41} & H_{42} & H_{43} & H_{44} & H_{45} \\ H_{51} & H_{52} & H_{53} & H_{54} & H_{55} \end{bmatrix}$$

the rank of this matrix is D=3,

Lemma 3.7. *All* 20 *cross channels along with any one of the direct channel of the* 5×5 *transfer matrix with rank* $D = 3$ *are algebraically independent.*

Proof. Proof is presented in Appendix B.3. □

Let us denote the first direct channel as $z = H_{11}$, and the fifth direct channel as $Z_{AI} = H_{55}$ (that is algebraically independent of other channels). Set of cross channels is denoted by X, similar to that in $K = 4$ setting.

Consider a 4×4 submatrix of H containing 3 direct channels, say H_{11}, H_{22}, H_{55}, wherein H_{55} is considered to be algebraically independent of all the cross channels. The determinant of any such sub-matrix is zero, since $D = 3$.

$$\begin{vmatrix} z & H_{12} & H_{14} & H_{15} \\ H_{21} & H_{22} & H_{24} & H_{25} \\ H_{31} & H_{32} & H_{34} & H_{35} \\ H_{51} & H_{52} & H_{54} & Z_{AI} \end{vmatrix} = 0$$

Evaluating the determinant, we can express H_{22} as a rational function of z, fifth direct channel Z_{AI} and 20 cross channels.

$$H_{22} = f_2(z, X, Z_{AI}) \tag{3.19}$$

The denominator polynomial of this rational function can be shown to be non-zero. This is due to the algebraic independence of all cross channels and Z_{AI}, as discussed in Appendix C2. The denominator could still evaluate to zero for some realizations in $\mathbb{S}$ and $f_2(z, X, Z_{AI})$ will be undefined for these realizations. But we can consider a domain $\mathbb{D}$ under which $f_2(z, X, Z_{AI})$ is always defined, *i.e.*, set of all points in $\mathbb{S}$ for which the denominator polynomial is always non-zero.

Now, let us consider determinant of another 4×4 submatrix comprising of H_{11}, H_{22} and H_{55}.

$$\begin{vmatrix} z & H_{12} & H_{13} & H_{15} \\ H_{21} & f_2(z, X, Z_{AI}) & H_{23} & H_{25} \\ H_{41} & H_{42} & H_{43} & H_{45} \\ H_{51} & H_{52} & H_{53} & Z_{AI} \end{vmatrix} = 0$$

Evaluating above determinant, we get a multivariate polynomial which is quadratic in z of the form

$$A(X, Z_{AI})z^2 + B(X, Z_{AI})z + C(X, Z_{AI}) = 0 \tag{3.20}$$

where $A(X, Z_{AI}), B(X, Z_{AI}), C(X, Z_{AI})$ are all polynomial functions of the 20 cross channels and 1 direct channel H_{55}. Also, $A(X, Z_{AI}), B(X, Z_{AI}), C(X, Z_{AI})$ are non-zero since X, Z_{AI} are algebraically independent.

Let us assume that there is a polynomial $Q(X)H_{11} - P(X)$ that always evaluates to zero under $\mathbb{D}$. We already know that $Q(X)$ is a non-zero polynomial, so we can express H_{11} as a rational function of the cross channels, *i.e.*, $z = H_{11} = \frac{P(X)}{Q(X)}$, which is always defined in the domain $\mathbb{D}' \subseteq \mathbb{D}$. Similar to $\mathbb{D}$, we can see that the set of points excluded from $\mathbb{S}$ to get $\mathbb{D}'$ has measure zero. Substituting this rational function for z in (3.20), we get

$$A(X, Z_{AI})\frac{p(X)^2}{q(X)^2} + B(X, Z_{AI})\frac{p(X)}{q(X)} + C(X, Z_{AI}) = 0 \tag{3.21}$$

Above equation holds if

- The polynomial in (3.21) is non-trivial and always evaluates to zero.
- $z = \frac{P(X)}{Q(X)}$ is a root of the quadratic equation (3.20).

If the non-trivial polynomial in (3.21) always evaluates to zero, then it gives a zero polynomial in the 20 cross channels and 1 direct channel, indicating that the 21 channels are algebraically dependent. However this is a contradiction.

If we could show that $z = \frac{P(X)}{Q(X)}$ cannot be a root, we can establish that $Q(X)H_{11} - P(X) \neq 0$. This is hard as we do not exactly know $P(X)$ or $Q(X)$. Using MATLAB, we could verify that for rational realizations of the 20 cross channels H_{ij} and 1 direct channel, the roots of the equation (3.20) are not always rational. Even though this helps in showing that the polynomial $Q(X)H_{11} - P(X)$ is almost surely non-zero in the rational space, we will not be able to make the same statement for the general space $\mathcal{S}$.

This analysis provides insights into why it is hard to prove the sufficient conditions, even for the simple 4-user and 5-user channels.

3.6.3 Challenges with Higher Number of Users

The 4-user and 5-user channels have algebraically independent cross channels. But as we increase the number of users to 6, we can see that the cross channels are no longer algebraically independent. To see how this might affect us, consider the 6-user interference channel with rank deficient transfer matrix $\mathbf{H}$. Similar to the analysis in section 3.6.1, consider a 4×4 submatrix of $\mathbf{H}$ containing only 2 direct channels, say $H_{11} = z$, and H_{22}, the determinant of any such sub matrix is zero.

$$\begin{vmatrix} z & H_{12} & H_{15} & H_{16} \\ H_{21} & H_{22} & H_{25} & H_{26} \\ H_{31} & H_{32} & H_{35} & H_{36} \\ H_{41} & H_{42} & H_{45} & H_{46} \end{vmatrix} = 0 \tag{3.22}$$

Evaluating this determinant, we can express H_{22} as a rational function of H_{11} and the 12 cross channels, $H_{22} = f_2(z, X)$, and this rational function is always defined in a domain $\mathbb{D}$.

Consider the determinant of another 4×4 submatrix containing of H_{11} and $H_{22} = f_2(z, X)$.

$$\begin{vmatrix} z & H_{12} & H_{13} & H_{14} \\ H_{21} & f_2(z, X) & H_{23} & H_{24} \\ H_{51} & H_{52} & H_{53} & H_{54} \\ H_{61} & H_{62} & H_{63} & H_{64} \end{vmatrix} = 0 \tag{3.23}$$

Evaluating this determinant, we can get a multivariate polynomial which is quadratic in z of the form

$$A_1(X)z^2 + B_1(X)z + C_1(X) = 0 \tag{3.24}$$

where $A(X), B(X), C(X)$ are all polynomial functions of the cross channels. We can do the same for H_{33}, by considering two different sub matrices containing H_{11} and H_{33}, and derive another multivariate polynomial which is quadratic in z after modifying the domain $\mathbb{D}$ to include the rational function $H_{33} = f_3(z, X)$.

$$A_2(X)z^2 + B_2(X)z + C_2(X) = 0 \tag{3.25}$$

In the 4-user case, doing this with H_{33} would result in the same polynomial as (3.17). But in the 6-user case considered here, we can see that (3.24) will have certain cross channel coefficients, namely H_{24} and H_{42}, which are not present in (3.25), and (3.25) will have certain cross channel coefficients, namely H_{34} and H_{43}, which are not present in (3.24). By linearly combining (3.24) and (3.25) after scaling them appropriately, we can eliminate the z^2 terms and solve for z as a function of the cross channels.

$$\begin{aligned}&(A_2(X)B_1(X) - A_1(X)B_2(X))z - \\ &(A_1(X)C_2(X) - A_2(X)C_1(X)) = 0\end{aligned} \tag{3.26}$$

The above equation shows that for a 6-user interference channel with rank $D = 3$, there exists a polynomial $Q(X)H_{11} - P(X) = 0$. But, both the $P(X)$ part and $Q(X)$ part of (3.26) have to be non-zero polynomials in order for (3.26) to be relevant, as under the modified alignment scheme it is not possible for either of them to be zero polynomials. That being said, by using MATLAB we can numerically confirm that these polynomials are non-zeros and thus (3.26) is relevant. In other words the desired and interfering signal spaces will overlap at the receivers if we try to use the modified scheme for the 6-user channel.

Even though this is not enough to say that the DoF outer bounds from Lemma 3.3 are not achievable for higher number of users, it shows the complications that arise when we increase the number of users in the interference channel with rank deficient transfer matrix.

3.7 Summary

We introduced the problem of characterizing the DoF of the K-user interference channel with rank a deficient transfer matrix. We presented a modified asymptotic alignment scheme to handle the algebraic dependencies, and discussed the sufficient conditions to achieve half rate per user. We illustrated the difficulties of proving the sufficient conditions for the simpler cases of $K = 4, 5$ and showed that the sufficient conditions are not met for $K \geq 6$. In conclusion, finding the optimal DoF of the general K-user interference channel with rank deficient transfer matrix remains open and presents a considerable challenge. The contents of this chapter appeared in a paper that we published in ISIT 2014 [66].

Chapter 4

Coded Caching

In this chapter, we look at content distribution networks with caches at the enduser. The cache essentially provides side information, and makes the problem reduce to index coding instances [76].

4.1 Introduction

Today's Internet traffic is dominated by content distribution services like live-streaming and video-on-demand. These services have been the driving force behind the explosive traffic growth seen in recent times. Popular services like Netflix, YouTube, etc, exhibit two important features: (i) the user demands are predictable based on their statistical history [67] and (ii) they exhibit strong temporal variability, resulting in highly congested peak hours and underutilized off-peak hours.

A common approach is to take advantage of the memories distributed across the network (at end users and/or inside the network) to store some of the popular contents. This process, termed *caching*, can be done during off-peak hours, so that during peak hours user requests can be served from these locally available memories without having to burden the network. Design and analysis of caching techniques for various kinds of networks have been researched extensively in the past [68–72] along with the impact of user demand statistics (file popularities) on the performance of caching [67, 73, 74]. This body of prior work only considered uncoded caching. It is easy to see that filling the cache with the most popular files (LFU) is indeed the optimal strategy for a system with only one user [75], but as we move on to systems with requests from multiple users, new challenges and opportunities arise.

A *coded caching* strategy was proposed by Maddah-Ali *et al.* in [76] for a system of uniform user demands and centralized content placement scheme and was extended to decentralized approach [77] and non-uniform Zipf based user demands in [78]. In a recent work [79], Ji *et al.* independently and in parallel proved order optimality for a caching policy that is uniform or uniform over a subset of the files based on the Zipf parameter for a Zipf distributed file popularity distribution. The coded caching problem typically consists of two phases, the *placement* phase where the contents are placed in local memory based on the statistics of user demands (file popularities) and the *delivery* phase where the remaining content, which is not available locally, is delivered after the demands of the users have been revealed.

In this chapter, we focus mainly on the delivery phase. Our main contribution is the design and evaluation of the *Heterogenous Coded Delivery (HCD)* scheme that improves upon the current state-of-the-art in coded caching and significantly reduces the load on the server during the delivery phase. The structure of the rest of this chapter is as follows. In Section 4.2, we formulate the problem. In Section 4.3, we discuss the intuition and background of coded caching. In Section 4.4, we present the proposed heterogenous coded delivery scheme and explain its working in detail. In Section 4.5, we present an evaluation of the proposed scheme. Section 4.7 concludes the chapter.

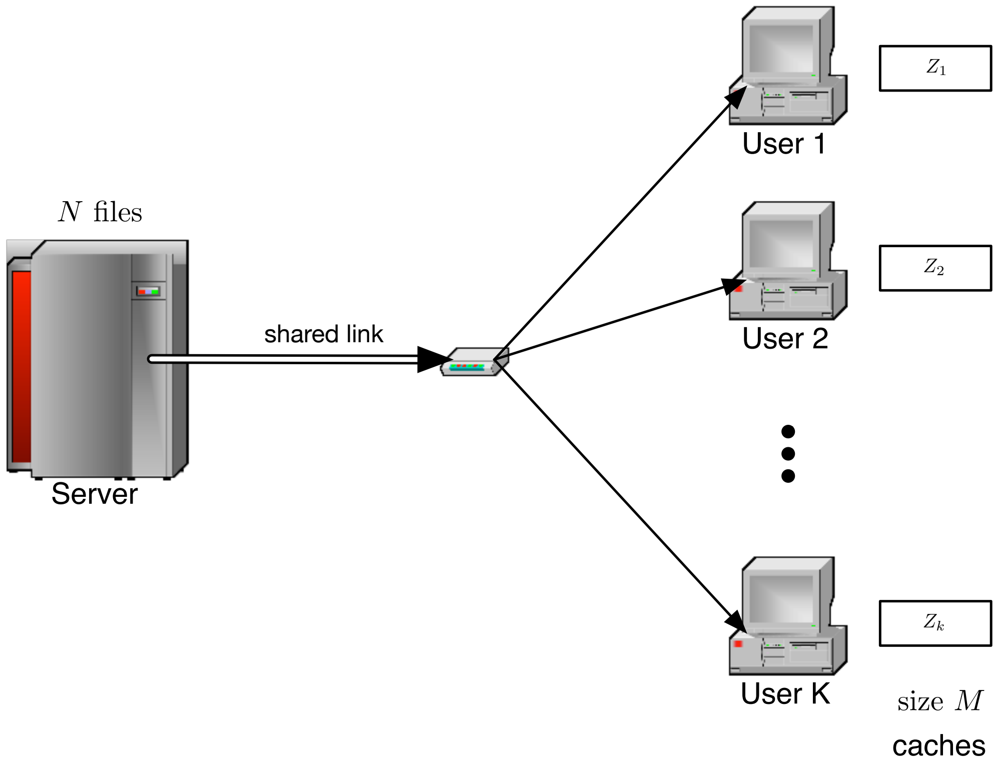


Figure 4.1: Problem Setup for Coded Caching.

4.2 Model and Assumptions

4.2.1 Problem Setting

We consider a system consisting of a server connected through a shared, error-free link to K users as illustrated in Fig. 4.1. The server has access to a database of N files $W_1, \ldots, W_N$ each of size F bits. Without loss of generality, we assume all files to be of the same size[1]. Each user k has an isolated cache memory Z_k of size MF bits (MB packets) for some real number $M \in [0, N]$. The popularity of a file W_n is the probability that this file is requested by a user. The file popularity distribution in the server is $\mathbf{p} = [p_n]_{n=1}^N$, where $\sum_{n=1}^N p_n = 1$. W.l.o.g. we can assume $p_1 \geq p_2 \geq \cdots \geq p_N$. The system operates in two phases: a *placement phase* and a *delivery phase*.

[1]Files of different sizes can be split into smaller files of the same size.

In the placement phase, the users are given access to the entire database $W_1, \ldots, W_N$ of files. Each user k is then able to fill the content of its cache Z_k using the database. The users follow a caching policy $\mathbf{q} = [q_n]_{n=1}^N$, where q_n denotes the fraction of the cache space in each user that will be allocated to the file W_n and $\sum_{n=1}^N q_n = 1$. The caching can either be done in a centralized manner, where the server besides deciding the caching policy $\mathbf{q}$ also decides what parts of each file is stored in each user's cache; or in a decentralized/distributed approach. In the decentralized approach, the server has no control over which parts of the file goes into each user's cache, it can only control what fraction of each file is cached (*i.e.,* the caching policy $\mathbf{q}$). The users randomly cache some portion (the size of this portion alone is dictated by the server) of each file in their corresponding cache. The server is assumed to have complete knowledge of Z_k, the content of the cache of user k. By knowing the content of cache Z_k, the server knows which bits/packets of each file are stored in the cache of user k.

In the delivery phase, only the server has access to the database. Each user k requests one of the files W_{d_k} in the database, where d_k represents the index of the file requested by user k. The vector $(d_1, \ldots, d_k)$ is a vector of indices of the files requested simultaneously by all K users ordered accordingly. The file requests are independently and identically distributed across all the users and follow the popularity distribution $\mathbf{p}$. The probability that a user requests file n is p_n. The server is informed of these requests and proceeds by transmitting a message $X_{(d_1,\ldots,d_k)}$ of size $R_{(d_1,\ldots,d_k)}F$ bits over the shared link for some fixed real number $R_{(d_1,\ldots,d_k)}$. $R_{(d_1,\ldots,d_k)}$ is referred to as the *load* in this chapter and is a measure of the length of the message. Using its cache content Z_k and the message $X_{(d_1,\ldots,d_k)}$ received over the shared link, each user k aims to reconstruct its requested file W_{d_k}.

4.2.2 Problem Statement

Problem Statement. Given the cache content Z_k for each user k and the exact demand vector $(d_1, d_2, \ldots, d_K)$, what is the optimal length $R^*_{(d_1,\ldots,d_k)}F$ of message $X_{(d_1,\ldots,d_k)}$ that the server should transmit to satisfy the given demand?

In general, this is an optimization of the delivery scheme and assumes the content of the caches to be given. As shown in the next subsection, this optimization is in general NP-hard. Here we design a practical delivery scheme for the given cache state that helps to reduce the length of the message $X_{(d_1,\ldots,d_k)}$ compared to the current state of the art scheme in [78].

The expected load is defined as the expectation of the loads over all possible demand vectors, *i.e.*, $\bar{R}(\mathbf{p}) = \sum_{(d_1,\ldots,d_K)} R_{(d_1,\ldots,d_K)} p_{d_1} p_{d_2} \cdots p_{d_K}$, and will be used to evaluate how the delivery schemes perform over all demands.

4.2.3 Formulation

In general, the process of various users caching different parts of a given file results in splitting a file into several nonoverlaping subfiles, such that each subfile is present in the caches of a distinct subset of users. Consider a file in the server, say A, there are K users in the system and during the placement phase, based on the caching policy, different parts of this file A might get placed at the cache of each user. Grouping the bits of this file based on the set of users they are cached at, the file A can be split into several subfiles $A_{\mathcal{S}}$, where each subfile $A_{\mathcal{S}}$ denotes the group of bits that are stored only in the cache of the specific set of users given by the corresponding $\mathcal{S} \subseteq \{1, \ldots, K\}$. For example, let us say there are $K = 2$ users, then the file A can be possibly split into $A_{\{\}}, A_{\{1\}}.A_{\{2\}}$ and $A_{\{1,2\}}$. Here $A_{\{\}}$ represents all the bits that are cached at neither of the two users, $A_{\{1\}}(A_{\{2\}})$ represents the bits that are cached only at user 1(2) and $A_{\{1,2\}}$ represents the bits that are cached at both users 1 and

2. Note that these subfiles are non-overlapping, *i.e.,* if a bit of the file is present in one subfile it cannot be present in any other subfile. Also note that not all subfiles have to be populated, in the example above if not even a single bit of file A is cached at both user 1 and 2 then $A_{\{1,2\}}$ is unpopulated and hence can be discarded. The cache Z_k of the user k can be thought of as a collection of such subfiles that are cached at this user k. The subfiles can in turn be classified into types based on the number of users they are cached at. We say *a subfile is of type* t if exactly t users have cached the bits of this subfile. It is easy to see that there are $K+1$ types of subfiles and there are $\binom{K}{t}$ subfiles for each type t.

In a centralized approach, the server has complete control over which bit of each file gets placed in the cache of each user. This basically means that the server can determine how a file is split in the subfiles defined above. In the decentralized approach, since the users randomly choose the bits they will cache, the splitting is also randomized.

Now consider a demand vector $(d_1, d_2, \ldots, d_K)$, which indicates that user 1 demands file W_{d_1}, user 2 requests file W_{d_2}, and so on. To simplify notation, let V_k denote the file requested by user k, *i.e.,* $V_k = W_{d_k}$ and $V_{k,\mathcal{S}}$ denote the subfiles corresponding to the file W_{d_k} that are requested by user k and are only available in the cache of users in $\mathcal{S}$. The notation $\mathcal{S}$ is used to refer to some ordered set of users, $\mathcal{S} \subseteq \{1, 2, \ldots, K\}$. For a given demand, based on the cache content across all users, the server creates subfiles of the form $V_{k,\mathcal{S}}$, for each user k, which will be used for transmission. The user k already knows some subfiles of the file V_k, as these subfiles are already present in its cache Z_k; so the server only needs to transmit the subfiles that are not present in its cache Z_k to satisfy user k's demand, *i.e.,* the server needs to send all populated subfiles of the form $V_{k,\mathcal{S}\setminus\{k\}}$ $\forall\mathcal{S}$, for that particular user k. Note that even if two users, i and j, request the same subfile, say $A_{\mathcal{S}}$, we will use separate notations $V_{i,\mathcal{S}}$ and $V_{j,\mathcal{S}}$ to denote the subfile requested by the respective users.

For a given caching and demand vector, finding the delivery scheme that minimizes the code length of the message is equivalent to solving an index coding problem [80, 81] whose side information graph is determined by the caching configuration. The side information graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{E}$ the set of (directed) edges, can be constructed as follows

- Each bit requested by each user is represented as a vertex. More specificially, every vertex $v \in \mathcal{V}$ corresponds to a bit in a subfile of the form $V_{k,\mathcal{S}\setminus\{k\}}$ that is requested by user k and is cached only at all the users in the set $\mathcal{S} \setminus \{k\}$. If two users request the same bit of a file, then that bit is represented by two distinct vertices. Note that the notation $V_{k,\mathcal{S}\setminus\{k\}}$, when used in context of the side information graph, denotes the group of vertices that represent the bits in the subfile $V_{k,\mathcal{S}\setminus\{k\}}$.
- There exists an edge $(u, v) \in \mathcal{E}$ iff the cache of the user requesting the bit represented by u contains the bit represented by v.

It is well-known that the index coding problem is, in general, NP-hard [82]. One can use the chromatic number based approach to get a sub-optimal solution [79] by constructing an undirected graph $G_a = (\mathcal{V}, \mathcal{E}_a)$ similar to G. The vertices of G_a are the same as the vertices of G and there exists an undirected edge $(u, v) \in \mathcal{E}_a$ between two vertices u and v iff the cache of the user requesting the bit represented by u contains the bit represented by v and the cache of the user requesting the bit represented by v also contains the bit represented by u. The chromatic number solution for the complement graph of G_a, which is equivalent to the minimum clique cover of the graph G_a, will give a sub-optimal approach for the delivery scheme in our problem, but finding the chromatic number is also known to be NP-hard, in general.

4.3 Background on Coded Caching

In this section, we present background on coded caching and a rather detailed overview of prior work, including observations and insights that inspired our new scheme.

4.3.1 Uniform Demands

We start from the intuition of the caching policy and the delivery scheme for the uniform demands scenario, considered in [76]. The N files in the database of the server are assumed to have equal popularity here, *i.e.,* the probability that a given file is requested by a user is $1/N$. Due to the uniform popularity, it only makes sense that an equal portion of each file be cached at the user caches and since each user has a cache of size M files, some M/N portion of each file should be cached at every user. In other words, we choose a caching policy where $q_n = M/N$ for all n. Also note that, [76] considers a centralized placement scheme, where the server can determine which bits of each file gets cached at each user.

Let us consider a file A. The caching policy dictates that each user caches MF/N bits of this file. In order to determine which bits get placed at which user's cache, the server splits this file A into $\binom{K}{t}$ subfiles of type t each labeled as $A_{\mathcal{S}}$, for some $\mathcal{S}$. The subset $\mathcal{S}$ comprises of t users and it is easy to see that there are $\binom{K}{t}$ such subsets for a system with K users. Since the content of the subfiles are nonoverlapping, each subfile will contain $F/\binom{K}{t}$ bits of the file A. A user k must cache all the subfiles $A_{\mathcal{S}}$, where $k \in \mathcal{S}$. There are $\binom{K-1}{t-1}$ such subfiles for each user and the total bits in them should add up to MF/N bits for each user.

$$\frac{\binom{K-1}{t-1}F}{\binom{K}{t}} = \frac{MF}{N} \Rightarrow t = \frac{M}{N}K$$

Thus the server will split each file into subfiles of type $t = MK/N$ in order to help the users determine which bits they will be storing in their respective caches.

Designing an optimal delivery scheme for a given demand vector is equivalent to solving an index coding problem. Due to the NP-hardness of this problem, practical solutions consider suboptimal solutions. The centralized placement scheme, described above, coupled with the uniform caching policy introduces a symmetry in the system that makes it easier to find the minimum clique cover (chromatic number) solution for any given demand vector. The undirected graph $G_a = (\mathcal{V}, \mathcal{E}_a)$ can be constructed for a given demand vector as explained in section 4.2.3 every user, there are $\binom{K-1}{t}$, subfiles each with $F/\binom{K}{t}$ bits that are not cached at that particular user. So the server needs to transmit a total of $K\binom{K-1}{t}F/\binom{K}{t}$ bits to satisfy the demand, *i.e.,* $|V| = K\binom{K-1}{t}F/\binom{K}{t} = K(1 - M/N)F$. An uncoded delivery scheme would require the server to send all those bits one by one (in the worst case scenario, when each user demands a different file) to satisfy the demand. The minimum clique cover provides a coding mechanism which would significantly reduce the number of bits that the server needs to transmit. Consider a subset $\mathcal{S}$ of $t+1$ users, for each $k \in \mathcal{S}$ consider a node in G_a that represents a bit in the subfile $V_{k,\mathcal{S}\setminus\{k\}}$. These $t+1$ nodes can form a clique of size $t+1$. By coding (XORing) together all the $t+1$ bits represented by these nodes and transmitting them to all users in $\mathcal{S}$, each user in $\mathcal{S}$ will be able to decode its desired bits using the information stored in its own cache. Note that $F/\binom{K}{t}$ cliques are required to cover all the bits in a subfile and each clique covers a bit in $t+1$ subfiles. Each clique here is a maximal clique and subsequently it is not hard to see that the cover indeed uses a minimum number of cliques. This whole process of clique cover based coding can be thought of bitwise coding (XOR) the subfiles of the form $V_{k,\mathcal{S}\setminus\{k\}}$ $\forall\ k \in \mathcal{S}$ and repeating this for all $\mathcal{S}$ of size $t+1$. In the end there will be $F/\binom{K}{t}$ cliques for each subset $\mathcal{S}$ of size $t+1$ and $\binom{K}{t+1}$ such subsets, so the server only needs to send a message of size $\binom{K}{t+1}F/\binom{K}{t}$ bits to satisfy all the demands. The transmitted message length $R_{(d_1,\dots,d_k)}F = \binom{K}{t+1}F/\binom{K}{t}$ remains the same across all demand vectors because of the symmetry and the uniformity in the placement and delivery schemes. For easier notation, let us denote this load $R_{(d_1,\ldots,d_k)}$ as just R.

$$R = \binom{K}{t+1} \frac{1}{\binom{K}{t}} = \frac{K-t}{t+1} = \frac{K(1-M/N)}{1+KM/N} \tag{4.1}$$

In [76], R is shown to be information theoretically optimal.

In the decentralized approach for content placement, the server has no control over which bits are cached at each user, so it is not possible to look at the files as a collection of subfiles of a single type. The users randomly select MF/N bits to store in their cache and based on this random placement of bits, each file can be seen as a collection of subfiles of various types. An algorithm was provided in [77] for the delivery in the decentralized setup, which was still based on the clique cover approach, but the clique cover solution yielded by this algorithm is not necessarily the minimum.

Insight. More importantly, the algorithm in [77] was restricted to form cliques only between nodes (i.e., code the corresponding bits together) in the subfiles of the same type, and thus missed coding opportunities We build upon this observation and design a new delivery scheme, which considers more coding opportunities by forming cliques between nodes not only of the same, but also of different type.

4.3.2 Non-Uniform Demans

Although uniform demands facilitate analysis, file popularities are far from uniform in practice; in fact, they could vary several orders of magnitude. In the uniform case, each file has the same probability of being requested by a user, thus it is natural to allocate equal cache to each file at every user. In the non-uniform case, the least popular file almost never get requested by users. Therefore, cache should not be allocated equally to highly popular files and least popular files. An intuitive way to share the cache is to make the caching policy $\mathbf{q}$

follow the popularity distribution $\mathbf{p}$.

The work in [78] considers Zipf-distributed file popularities and they propose a caching policy that groups files together. The files are divided into groups based on the closeness in their popularities and the caching policy q is designed such that all the files in the same group will have the same amount of cache space, q_n, which is determined by dividing the cumulative popularity of all the files in the corresponding group by the total number of files in that group. However, files in different groups will have different q_n (cache space allocation). They also explicitly state that the grouping can be optimized to minimize the expected load.

The work in [79] considers Zipf file popularities, but proposes a simpler and different caching policy than the one in [78]. [79] divides the files into just two groups. The files in one group are not allocated any cache space at all, *i.e.,* $q_n = 0$ for all the files in this group, whereas the files in the other group get to divide the entire cache space equally among themselves ($q_n = M/group\ size$, for all files in this group). Both [78] and [79] assume a decentralized placement scheme, because a centralized approach would require a lot of work from the server, which is not as practical as the decentralized approach.

Unlike the uniform case, the load $R_{(d_1,\dots,d_k)}$ varies across all demand vectors, thus it is more meaningful to consider the expected load $\bar{R}(\mathbf{p})$ instead. The delivery scheme used in [78] is essentially the same as the one in [77], as discussed briefly in section 4.3.1. The key idea behind the scheme in [78] is to code (bitwise XOR) together all the subfiles of the form $V_{k,\mathcal{S}\setminus\{k\}}\ \forall\ k \in \mathcal{S}$ and repeating this coding procedure for all valid $\mathcal{S}$. If the size of these subfiles is not the same, which is usually the case when employing non-uniform caching policy and/or decentralized placement scheme, the scheme just pads them with zeros to make all their sizes the same.

Considering a graph G_a constructed based on the cache content of the users, for each $\mathcal{S}$, the algorithm tries to cover all the nodes in the groups of the form $V_{k,\mathcal{S}\setminus\{k\}}\ \forall k \in \mathcal{S}$, by trying to form cliques of maximum size $|\mathcal{S} \setminus \{k\}|$ between the nodes across the groups $V_{k,\mathcal{S}\setminus\{k\}}$ $\forall k \in \mathcal{S}$. If it cannot find any uncovered node within the groups $V_{k,\mathcal{S}\setminus\{k\}}$ $\forall k \in \mathcal{S}$ to form a clique of size $|\mathcal{S} \setminus \{k\}|$, then the algorithm simply chooses to form a clique of lower size with the available nodes.

Insight. In the situation just described above, the algorithm unnecessarily restricts itself from considering all coding opportunities: it could form a bigger clique with nodes from a different group of the form $V_{k,\mathcal{S}'\setminus\{k\}}$, where $\mathcal{S} \subset \mathcal{S}'$. This is the key point we will exploit in the improved algorithm that we present in the next section. The delivery scheme in [79] makes use of the minimum clique cover solution, but finding the minimum clique cover is NP-hard and [79] does not provide any practical algorithm to do that efficiently. Both [78] and [79], prove that their respective placement and delivery scheme combinations are indeed order optimal. In particular, [78], chooses a specific grouping, where files with popularity differing by at most a factor of two are grouped together, to prove the order optimality of their approach.

4.4 Heterogenous Coded Delivery Scheme

In this section, we propose the Heterogenous Coded Delivery scheme (HCD). First, we define the algorithm and we discuss the core ideas and intuition on how this scheme achieves better performance than the state-of-the-art in [77] and [78]. Then, we walk through the details of this new scheme through an illustrative example.

4.4.1 The Scheme

The pseudocode for the HCD scheme is presented in Algorithm 1. HCD can be used with both centralized and decentralized placement approaches. The new scheme, similar to the

Algorithm 1 Heterogenous Coded Delivery (HCD)

procedure DELIVERY$(d_1, \ldots, d_K)$
    for $t = 1, 2, \ldots, K-1, K$ **do**
        for $\mathcal{S} \subseteq [K] : |\mathcal{S}| = t$ **do**
            $binsize = \max_{k \in \mathcal{S}} V_{k,\mathcal{S}\setminus\{k\}}$
            for $k \in \mathcal{S}$ **do**
                if $|V_{k,\mathcal{S}\setminus\{k\}}| < binsize$ **then**
                    Move $(binsize - |V_{k,\mathcal{S}\setminus\{k\}}|)$ bits from non-empty bins $V_{k,\mathcal{S}'\setminus\{k\}} : \mathcal{S}' \supset \mathcal{S}$ to *temp*
                    Create new subfile $V'_{k,\mathcal{S}\setminus\{k\}}$
                    $V'_{k,\mathcal{S}\setminus\{k\}} \leftarrow V_{k,\mathcal{S}\setminus\{k\}} + temp$
                    $coded \leftarrow coded \oplus V'_{k,\mathcal{S}\setminus\{k\}}$
                else
                    $coded \leftarrow coded \oplus V_{k,\mathcal{S}\setminus\{k\}}$
                end if
            end for
            server transmits subfile *coded*
        end for
    end for
end procedure

ones in [77, 78], is still based on finding a clique cover for the graph G_a. But HCD exploits the possibilities of forming cliques with nodes from subfiles of higher types, which could potentially reduce the number of cliques required for the cover. The users have already populated the cache based on some caching policy and placement scheme. Once the server is informed of the user requests, *i.e.,* the demand vector $(d_1, \ldots, d_K)$, it would be able to format the subfiles V_k that are required for transmission.

The procedure DELIVERY is called for the given demand vector to determine the message $X_{(d_1,\ldots,d_k)}$ that needs to be transmitted to satisfy the demands. The scheme requires to iterate over all possible subsets of users, $\mathcal{S}$, and the two outermost loops in the algorithm help to do that. Note that in Algorithm 1, we use the notation $[K]$ to refer to the set $\{1, 2, \ldots, K\}$, the operator $+$ refers to concatenation and $\oplus$ refers to bitwise XOR operation. The innermost loop helps to iterate through each user within a given subset $\mathcal{S}$. The steps within the inner most loop are the core of the algorithm and the difference between our algorithm and those presented in [77, 78]. Recall that the algorithms in [77, 78] code together all the subfiles of the form $V_{k,\mathcal{S}\setminus\{k\}}$ $\forall$ $k \in \mathcal{S}$, after appending zeros to make them all of the same size (equal number of bits). In our algorithm, instead of appending zeros right away, we first try to borrow bits from the *immediate* higher type subfiles of the form $V_{k,\mathcal{S}'\setminus\{k\}}$, where $\mathcal{S}' \supset \mathcal{S}$. After exhausting all the options for borrowing, we append zeros. In the algorithm, this borrowing process is accompanied by a step involving the creation of new subfile $V'_{k,\mathcal{S}\setminus\{k\}}$. We do this to avoid conflict with the definition of $V_{k,\mathcal{S}'\setminus\{k\}}$. Note that there are bits in the new subfile that, although present in the caches of users in $\mathcal{S} \setminus \{k\}$, are no longer cached *only* at these user in $\mathcal{S} \setminus \{k\}$. Since $\mathcal{S}'$ can be any superset of $\mathcal{S}$, all the bits in the new subfile $V'_{k,\mathcal{S}\setminus\{k\}}$ are cached at all the users in $\mathcal{S} \setminus \{k\}$ and so they would still be able to decode their respective bits using the information present in their cache. Also note that the borrowing process actually involves moving bits from the original subfile, not just copying them. It is important to first code and transmit the subfiles of lower type before moving on to higher types, because a subfile can only borrow bits from the corresponding subfiles of higher types.

In graph theoretic terms, we consider a graph G_a constructed based on the cache content of the users, as in [77, 78]. For each $\mathcal{S}$ the algorithm tries to cover all the nodes in the groups of the form $V_{k,\mathcal{S}\setminus\{k\}}$ $\forall k \in \mathcal{S}$, by trying to form cliques of maximum size $|\mathcal{S} \setminus \{k\}|$ between the nodes across the groups $V_{k,\mathcal{S}\setminus\{k\}}$ $\forall k \in \mathcal{S}$. Our algorithm differs in the fact that, if it cannot find any uncovered node within the groups $V_{k,\mathcal{S}\setminus\{k\}}$ $\forall k \in \mathcal{S}$ to form a clique of size $|\mathcal{S} \setminus \{k\}|$, instead of just settling with forming a clique of lower size, our algorithm tries to cover the nodes from the corresponding higher type groups $V_{k,\mathcal{S}'\setminus\{k\}}$, where $\mathcal{S}' \supset \mathcal{S}$. We would like to point out that, since our algorithm tries to cover nodes from higher type groups whenever possible, even in the worst-case scenario the performance of our algorithm will be at least as good as the ones in [77, 78].

Observe that we do not try to optimize the borrowing step, *i.e.,* we do not try to determine the subfiles to borrow bits from such that final clique is minimized. The optimization of this

W_1	$p_1 = 0.3$
W_2	$p_2 = 0.2$
W_3	$p_3 = 0.2$
W_4	$p_4 = 0.2$
W_5	$p_5 = 0.1$

Figure 4.2: File Server with $N = 5$ and popularity distribution $\mathbf{p}$.

step is the core of the minimum clique cover and so could be NP-hard. Instead, we choose to go with a simpler and more practical approach wherein a subfile $V_{k,\mathcal{S}\setminus\{k\}}$, if required, will first borrow from a valid *immediate* higher type subfile of the form $V_{k,\mathcal{S}'\setminus\{k\}}$, where $\mathcal{S}' \supset \mathcal{S}$. For example, consider an iteration where we are trying to code together the subfiles $V_{1,\{2\}}$, which has 5 bits, and $V_{2,\{1\}}$, which only has 1 bit, and the other subfiles related to user 2 present in the system are $V_{2,\{1,3,4\}}, V_{2,\{1,4,5\}}$ and $V_{2,\{1,3,4,5\}}$ with 2 bits each. The algorithm will first try to borrow from the immediate higher type subfiles, which in this case are $V_{2,\{1,3,4\}}$ and $V_{2,\{1,4,5\}}$. The next higher type subfile $V_{2,\{1,3,4,5\}}$ will only be considered if $V_{2,\{1,3,4\}}$ and $V_{2,\{1,4,5\}}$ do not have enough bits to borrow from.

4.4.2 Example

We now present a detailed walkthrough of our algorithm through an illustrative example. This will help highlight the subtleties in the proposed scheme.

Example 4.1. *Consider a system with a server consisting of $N = 5$ files, each of size F bits, $K = 5$ users, each with a cache of size $M = 2$ files and popularity distribution p as shown in fig. 4.2. The file popularity distribution takes three distinct values: $p_1 = 0.3$, $p_2 = p_3 = p_4 = 0.2$ and $p_5 = 0.1$.*

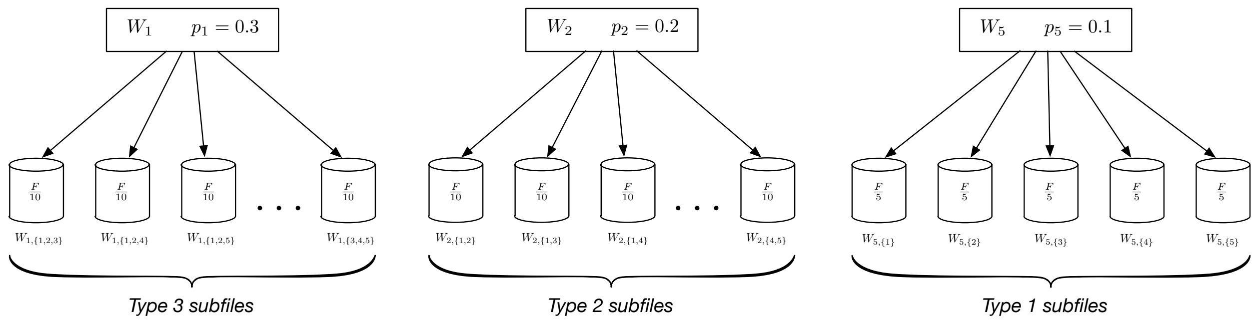


Figure 4.3: Placement Phase

For this example, we will consider a caching policy that follows exactly the popularity distribution, *i.e.,* $\mathbf{q} = \mathbf{p}$, and a centralized placement scheme. We choose the centralized placement scheme as it helps to highlight the key difference between our algorithm and the one in [77, 78]. The centralized scheme for the non-uniform caching policy is quite similar to the one described in section 4.3.1. The server splits a file W_i into $\binom{K}{t_i}$ subfiles of type $t_i = p_i MK$, each getting $F/\binom{K}{t_i}$ bits as shown in fig. 4.3. The server splits the file W_1 into $\binom{5}{3} = 10$ subfiles of type $t_1 = p_1 MK = 3$, namely $W_{1,\{1,2,3\}}$, $W_{1,\{1,2,4\}}$, $W_{1,\{1,2,5\}}$, $W_{1,\{1,3,4\}}$, $W_{1,\{1,3,5\}}$, $W_{1,\{1,4,5\}}$, $W_{1,\{2,3,4\}}$, $W_{1,\{2,3,5\}}$, $W_{1,\{2,4,5\}}$ and $W_{1,\{3,4,5\}}$, each with $F/10$ bits. Similarly, each of the files (W_2, W_3, W_4) with popularity 0.2, will be split into $\binom{5}{2} = 10$ subfiles of type 2 ($t_2 = t_3 = t_4 = 0.2MK = 2$), each with $F/10$ bits. For instance, the file W_2 will be split into the following subfiles: $W_{2,\{1,2\}}$, $W_{2,\{1,3\}}$, $W_{2,\{1,4\}}$, $W_{2,\{1,5\}}$, $W_{2,\{2,3\}}$, $W_{2,\{2,4\}}$, $W_{2,\{2,5\}}$, $W_{2,\{3,4\}}$, $W_{2,\{3,5\}}$ and $W_{2,\{4,5\}}$. Finally, the file W_5 will be split into $\binom{5}{1} = 5$ subfiles of type $t_5 = 0.1MK = 1$, each with $F/5$ bits, namely $W_{5,\{1\}}, W_{5,\{2\}}, W_{5,\{3\}}, W_{5,\{4\}}$ and $W_{5,\{5\}}$. A user i will store in its cache all the subfiles of the form $W_{k,\{\mathcal{S}:i\in\mathcal{S}\}}$ $\forall\ k$, *i.e.,* it will store all the subfiles with a label of the form $W_{k,\mathcal{S}}$, where $i \in \mathcal{S}$. Observe that now each user will have cached $p_i MF$ bits of the file W_i and since $\sum p_i = 1$, the users end up with their caches fully occupied.

We will first consider the demand vector $(W_1, W_2, W_3, W_4, W_5)$ to see the improvements this new scheme offers. After the server is informed about the demand vector, it would be able to create and populate the subfiles $V_{k,\mathcal{S}}$ using the corresponding $W_{d_k,\mathcal{S}}$. User 1 demands the file

Table 4.1

User i	Subfiles requested but not cached at user i	Size of each bin
1	$V_{1,\{2,3,4\}}, V_{1,\{2,3,5\}}, V_{1,\{2,4,5\}}, V_{1,\{3,4,5\}}$	$F/10$
2	$V_{2,\{1,3\}}, V_{2,\{1,4\}}, V_{2,\{1,5\}}, V_{2,\{3,4\}}, V_{2,\{3,5\}}, V_{2,\{4,5\}}$	$F/10$
3	$V_{3,\{1,2\}}, V_{3,\{1,4\}}, V_{3,\{1,5\}}, V_{3,\{2,4\}}, V_{3,\{2,5\}}, V_{3,\{4,5\}}$	$F/10$
4	$V_{4,\{1,2\}}, V_{4,\{1,3\}}, V_{4,\{1,5\}}, V_{4,\{2,3\}}, V_{4,\{2,5\}}, V_{4,\{3,5\}}$	$F/10$
5	$V_{5,\{1\}}, V_{5,\{2\}}, V_{5,\{3\}}, V_{5,\{4\}}$	$F/5$

W_1. It already has the subfiles $V_{1,\{1,2,3\}}$, $V_{1,\{1,2,4\}}$, $V_{1,\{1,2,5\}}$, $V_{1,\{1,3,4\}}$, $V_{1,\{1,3,5\}}$ and $V_{1,\{1,4,5\}}$ in its cache, so the server only needs to send the remaining subfiles $V_{1,\{2,3,4\}}$, $V_{1,\{2,3,5\}}$, $V_{1,\{2,4,5\}}$ and $W_{1,\{3,4,5\}}$. Table 4.1 shows the subfiles, along with the number of bits in each, that the server has to send for each user.

The algorithm goes through all the subfiles in Table 4.1, starting with the lower type ones. In this example the lowest type subfiles are $V_{5,\{1\}}, V_{5,\{2\}}, V_{5,\{3\}}$ and $V_{5,\{4\}}$, each containing $F/5$ bits of the file V_i that is not cached at user 5. If there is a subfile $V_{1,\{5\}}$ containing bits demanded by user 1 and cached only at user 5, we will be able to code (bitwise XOR) $V_{5,\{1\}}$ with $V_{1,\{5\}}$. But such a subfile is not present, so we create a new subfile $V'_{1,\{5\}}$ and populate it with bits borrowed from $V_{1,\{2,3,5\}}$, $V_{1,\{2,4,5\}}$ and $V_{1,\{3,4,5\}}$, which we refer to as the donor subfiles. $V_{1,\{5\}}$ has $F/5$, so we borrow $F/15$ bits from each of the three donor files reducing their size to $F/30$ bits. Note that the bits in these donor subfiles are all demanded by user 1 and are cached at user 5 (and also at few other users), so by transmitting the $F/5$ bits obtained by coding $V_{5,\{1\}}$ with $V'_{1,\{5\}}$ user 1 would still be able to recover its requested bits and at the same time user 5 would also be able to receive and recover some of its requested bits. This process is repeated for all the remaining subfiles type 1 as shown in table 4.2 and fig. 4.4.

Moving on to the type 2 subfiles, we can see that there are 18 subfiles under this type. Originally, all of these subfiles had $F/10$ bits each, but the borrowing steps shown in table 4.2 has reduced the size of some of them to $F/30$ bits. Consider the subfiles $V_{2,\{3,4\}}$, $V_{3,\{2,4\}}$ and

Table 4.2

Subfiles	Size	Donor subfiles : Bits left (before)	Bits taken	Bits left (after)
$V'_{1,\{5\}}$	$\frac{F}{5}$	$V_{1,\{2,3,5\}} : \frac{F}{10}$ $V_{1,\{2,4,5\}} : \frac{F}{10}$ $V_{1,\{3,4,5\}} : \frac{F}{10}$	$\frac{F}{15}$ $\frac{F}{15}$ $\frac{F}{15}$	$\frac{F}{30}$ $\frac{F}{30}$ $\frac{F}{30}$
$V'_{2,\{5\}}$	$\frac{F}{5}$	$V_{2,\{1,5\}} : \frac{F}{10}$ $V_{2,\{3,5\}} : \frac{F}{10}$ $V_{2,\{4,5\}} : \frac{F}{10}$	$\frac{F}{15}$ $\frac{F}{15}$ $\frac{F}{15}$	$\frac{F}{30}$ $\frac{F}{30}$ $\frac{F}{30}$
$V'_{1,\{5\}}$	$\frac{F}{5}$	$V_{3,\{1,5\}} : \frac{F}{10}$ $V_{3,\{2,5\}} : \frac{F}{10}$ $V_{3,\{4,5\}} : \frac{F}{10}$	$\frac{F}{15}$ $\frac{F}{15}$ $\frac{F}{15}$	$\frac{F}{30}$ $\frac{F}{30}$ $\frac{F}{30}$
$V'_{1,\{5\}}$	$\frac{F}{5}$	$V_{4,\{1,5\}} : \frac{F}{10}$ $V_{4,\{2,5\}} : \frac{F}{10}$ $V_{4,\{3,5\}} : \frac{F}{10}$	$\frac{F}{15}$ $\frac{F}{15}$ $\frac{F}{15}$	$\frac{F}{30}$ $\frac{F}{30}$ $\frac{F}{30}$

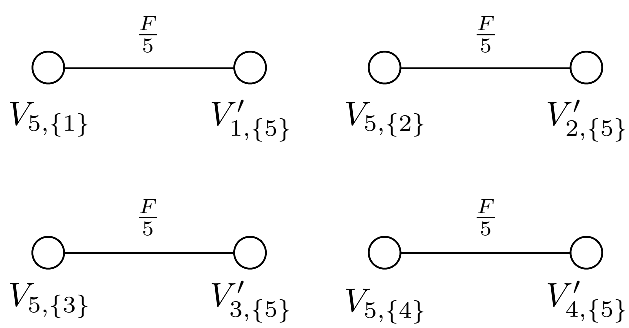


Figure 4.4: Cliques of size 2

$V_{4,\{2,3\}}$, none of them were used in table 4.2 and so these subfiles can be coded together and transmitted as a single subfile of size $F/10$ bits.

Next, consider the subfile $V_{2,\{1,5\}}$, it could be potentially coded together with bins $V_{1,\{2,5\}}$ and $V_{5,\{1,2\}}$ if they were present. We can create new subfile $V'_{1,\{2,5\}}$ and fill it up with $F/30$ bits from donors $V_{1,\{2,3,5\}}$ and $V_{1,\{2,4,5\}}$, but there is no point in creating a subfile $V_{5,\{1,2\}}$ as it does not have any donor subfiles of higher types to borrow from. $V_{3,\{1,5\}}$ and $V_{4,\{1,5\}}$ encounter a similar situation and the entire process for those subfiles can be seen in table 4.3

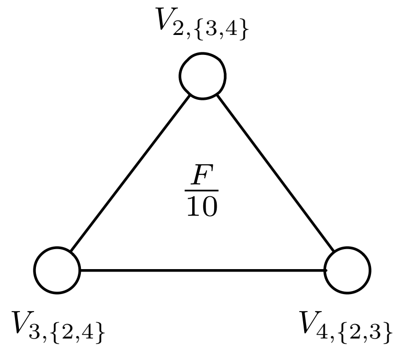


Figure 4.5: Cliques of size 3, no bits borrowed.

Table 4.3

Subfiles	Size	Donor subfiles : Bits left (before)	Bits taken	Bits left (after)
$V'_{1,\{2,5\}}$	$\frac{F}{30}$	$V_{1,\{2,3,5\}} : \frac{F}{30}$	$\frac{F}{60}$	$\frac{F}{60}$
		$V_{1,\{2,4,5\}} : \frac{F}{30}$	$\frac{F}{60}$	$\frac{F}{60}$
$V'_{1,\{3,4\}}$	$\frac{F}{30}$	$V_{1,\{2,3,5\}} : \frac{F}{60}$	$\frac{F}{60}$	0
		$V_{1,\{3,4,5\}} : \frac{F}{30}$	$\frac{F}{60}$	$\frac{F}{60}$
$V'_{1,\{4,5\}}$	$\frac{F}{30}$	$V_{1,\{2,4,5\}} : \frac{F}{60}$	$\frac{F}{60}$	0
		$V_{1,\{3,4,5\}} : \frac{F}{60}$	$\frac{F}{60}$	0

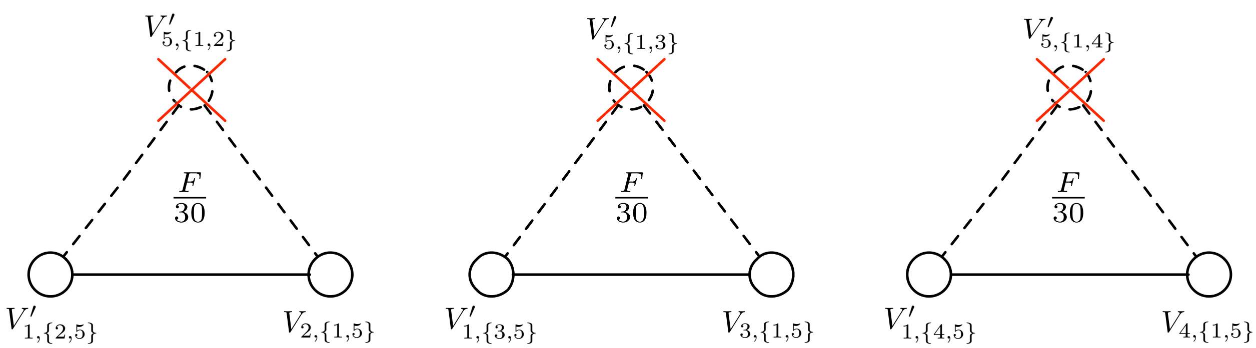


Figure 4.6: Cliques of size 3 that are reduced to cliques of size 2, bits borrowed.

and fig. 4.6.

Consider subfile $V_{2,\{3,5\}}$, it could be potentially coded with $V_{3,\{2,5\}}$ and $V_{5,\{2,3\}}$, one of which is already present. As in the previous case, it does not make sense to create a new subfile $V'_{5,\{2,3\}}$ as it does not have any higher type donor subfiles to borrow from. Both $V_{2,\{3,5\}}$ and $V_{3,\{2,5\}}$ have $F/30$ bits as seen in table 4.2 each, so we could code these two together and

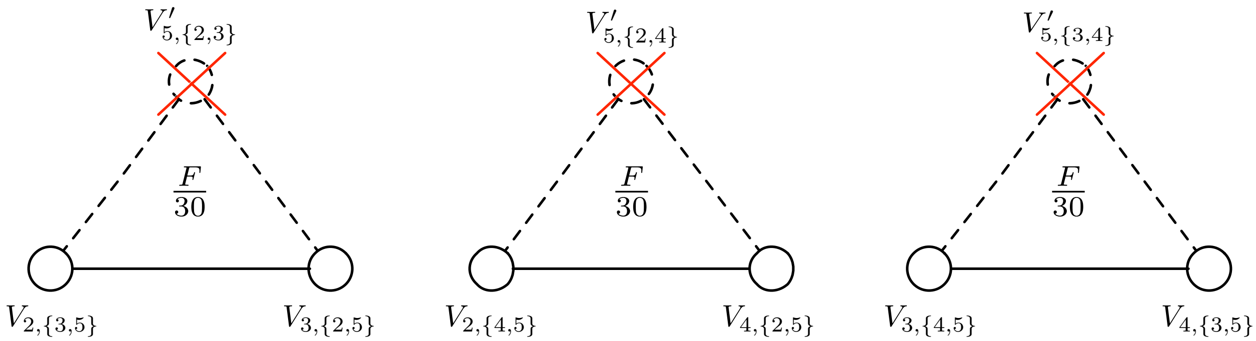


Figure 4.7: Cliques of size 3 that are reduced to cliques of size 2, no bits borrowed.

Table 4.4

Subfiles	Size	Donor subfiles : Bits left (before)	Bits taken from each bin	Bits left (after)
$V'_{1,\{2,3\}}$	$\frac{F}{10}$	$V_{1,\{2,3,4\}}$: $\frac{F}{10}$	$\frac{F}{10}$	0
		$V_{1,\{2,3,5\}}$: 0	0	0
$V'_{1,\{2,4\}}$	$\frac{F}{10}$	$V_{1,\{2,3,4\}}$: 0	0	0
		$V_{1,\{2,4,5\}}$: 0	0	0
$V'_{1,\{3,4\}}$	$\frac{F}{10}$	$V_{1,\{2,3,4\}}$: 0	0	0
		$V_{1,\{3,4,5\}}$: 0	0	0

transmit a single subfile of size $F/30$ bits. This, along with a similar process for $V_{2,\{4,5\}}$, $V_{4,\{2,5\}}$, $V_{3,\{4,5\}}$ and $V_{4,\{3,5\}}$, are shown in fig. 4.7.

There are still a few more subfiles of type 2 that have not been transmitted yet, namely $V_{2,\{1,3\}}$, $V_{3,\{1,2\}}$, $V_{2,\{1,4\}}$, $V_{4,\{1,2\}}$, $V_{3,\{1,4\}}$ and $V_{4,\{1,3\}}$. None of them were used as a donor so far and so each of them still has $F/10$ bits. $V_{2,\{1,3\}}$ and $V_{3,\{1,2\}}$ can be coded together with $V_{1,\{2,3\}}$, which is not present. So we create a new subfile $V'_{1,\{2,3\}}$ and try to fill it up with bits from donor subfile. We can see from Table 4.4 that there is only one non-empty donor subfile that we could borrow from and we will borrow all $F/10$ bits from it. For the other subfiles, we are left with no donor subfiles to borrow from and hence it does not make sense to create new subfiles. The process is illustrated in table 4.4 and fig. 4.8.

In total, we have transmitted 4 coded subfiles of size $F/5$ bits (fig. 4.4), 4 coded subfiles of size

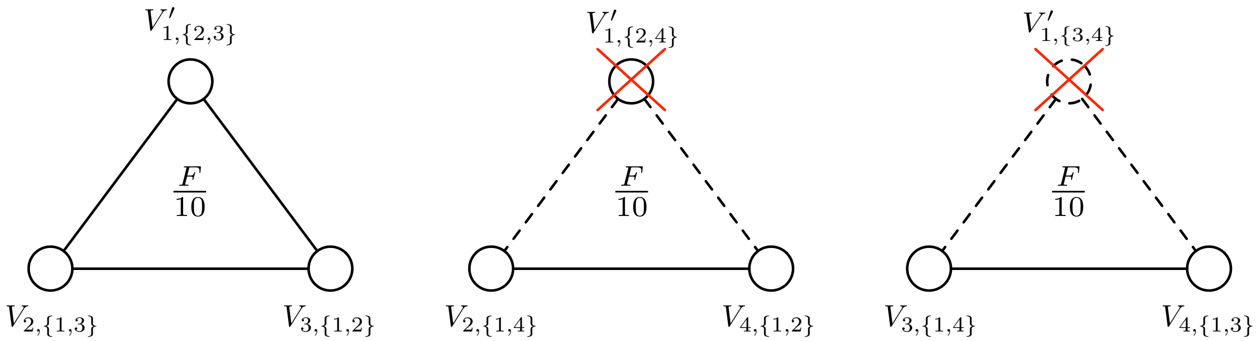


Figure 4.8: Cliques of size 3, some are reduced to size 2.

$F/10$ bits (figs. 4.5 and 4.8) and 6 coded subfiles of size $F/30$ bits. Therefore, the total length of the message that was transmitted to satisfy the demands vector $(W_1, W_2, W_3, W_4, W_5)$ is

$$4 \times \frac{F}{5} + 6 \times \frac{F}{30} + 4 \times \frac{F}{10} = 1.4F \text{ bits} \tag{4.2}$$

Baseline. The scheme in [77, 78] does not borrow bits from higher type subfiles during the coding process, so their scheme will end up transmitting 4 subfiles of size $F/5$ bits for user 5's request, 10 coded subfiles of size $F/10$ bits for user $2, 3$ and 4's requests and 4 more subfiles of size $F/10$ for user 1's request. This sums up to a total of $2.2F$ bits, which is considerably more than the length of our scheme.

4.5 Evaluation of the Coding Scheme

We present an evaluation of the Heterogeneous Coded Delivery (HCD) scheme using simulations for both the centralized and decentralized approaches. Our focus is mainly on the evaluation of the delivery scheme, *not* of the caching policy; the latter is fixed for each evaluation. The load $R_{(d_1,d_2,\ldots,d_k)}$ of the coded message depends on the demand vector $(d_1, d_2, \ldots, d_K)$,. Therefore, to evaluate the overall performance, we will use the expected load metric $\bar{R}(\mathbf{p}) = \sum_{(d_1,\ldots,d_K)} R_{(d_1,\ldots,d_K)} p_{d_1} p_{d_2} \cdots p_{d_K}$ throughout this section. We will be mainly comparing the expected load of our scheme to the state-of-the-art delivery scheme provided in [77, 78].

A note on the decentralized scenario. We would like to highlight some points before diving into the evaluation results. In the decentralized approach, the users randomly cache $q_n MF$ bits of file W_n in their respective caches. Because of the randomness here, the bits of a file are distributed across subfiles of various types, unlike the centralized approach where they are contained in subfiles of a single type. It is easy to see that a bit in file W_n has a probability $q_n M$ of being cached at any given user. Note $q_n M \leq 1$, as it does not make sense to allot more that the $1F$ bits for caching the file W_n. The probability that a bit gets cached at t users can be easily deduced as:

$$Pr(\text{a bit is cached at } t \text{ users}) = \binom{K}{t} (q_n M)^t (1 - q_n M)^{K-t} \tag{4.3}$$

Thus the number of bits in type t subfiles can be modeled as a binomial distribution $B(K, q_n M)$ and the bits within a type t are uniformly distributed across all the subfiles of type t. We will use this modeling to simulate the decentralized caching.

4.5.1 Uniform Caching

Under a uniform caching policy, all N files will be allocated an equal amount of cache space at each user, *i.e.,* $q_n = 1/N \ \forall \ n$. If such a caching policy is used, irrespective of the file popularity distribution, both our scheme and the scheme in [77] would yield similar load values. Especially, a centralized placement scheme following a uniform caching policy will have a strong symmetry in the cache contents, which would result in our scheme essentially reducing to the scheme in [77]. In the centralized approach, the bits of a file are split within subfiles of the same type, and due to the uniform nature of the caching policy, a similar phenomenon can be observed across all the files. Here the step creating new subfiles that borrow bits from subfiles of higher type becomes unnecessary.

In the decentralized approach, the subfile types are no longer limited to a single value. The bits of a file get distributed across several subfiles types, which can be modeled as a binomial distribution. This distribution maintains its parameters across all files because of the uniformity of q_n. All the subfiles which would be coded together in the delivery phase will have almost the same number of bits, thereby minimizing the need to borrow bits from higher type subfiles in our algorithm. Thus the improvement we get from our algorithm, compared to the state of the art, would be negligible. This was indeed observed in our simulations.

4.5.2 Non-Uniform Caching

We define non-uniform caching as a caching policy that does not allocate equal cache space for all files irrespective of their popularities. This definition includes the multilevel grouping in [78] and the two level policy used in [79].

The caching policy considered in [79] divides the files into two groups, one group with files of low popularity which will not be cached at any user and the remaining files of higher popularity which will divide the cache space equally among themselves. The bits of these files cannot be coded together with another bit, as the users do not have the side information necessary to decode. The bits of the files in the later groups are uniformly divided in subfiles of single type in a centralized approach. Therefore, the HCD scheme will not have performance gain compared to the state-of-the-art. The same arguments from decentralized uniform caching can be used to explain the negligible difference in load observed in the decentralized approach for this caching policy.

If the considered caching policy has multiple levels like the one in [78] or example 4.1, then we can see a significant difference in the expected load. Example 4.1 uses a caching policy where $\mathbf{q} = \mathbf{p}$, we were able to calculate the expected load for HCD scheme as 1.156 and the expected load of state of the art scheme as 1.696. This brings an improvement of nearly 30% for this example. We also evaluated our scheme for a system with $N = 10$ files, $K = 5$ users and $M = 4$ sized cache, where the user demands are Zipf distributed. The results of the evaluation for a centralized caching approach is presented in fig. 4.9. The figure shows a plot of the expected load vs Zipf parameter for four different caching policies. The first three are arbitrary policies with multiple level groups, we are able to see a significant improvement for these; the fourth one is similar to the policy in [79] and our scheme does not provide an improvement.

The performance evaluation of HCD for a decentralized caching approach is shown in fig. 4.10 again for a system with $N = 10$, $K = 5$ and $M = 4$, but in here we consider a caching policy that is also Zipf based, namely Zipf(0.5,N). In this case as well, there is an apparent performance benefit from HCD compared to the current state-of-the-art.

4.5.3 Comparison to State-of-the-Art.

Performance. HDC performs always at least as well as the baseline scheme, by design: HDC considers all the coding opportunities that the baseline does, and then some more. In the uniform case, however, there is no substantial benefit: due to the symmetry of the problem, our scheme essentially degenerates to the baseline. The significant benefits come in the non-uniform case, for which our scheme was specifically designed to extend beyond the baseline approach.

Complexity. HDC has a polynomial complexity in the number of subfiles $V_{k,\mathcal{S}}$ (nodes of the graph G_d), the same as the coded caching schemes in [77] and [78]. The main difference

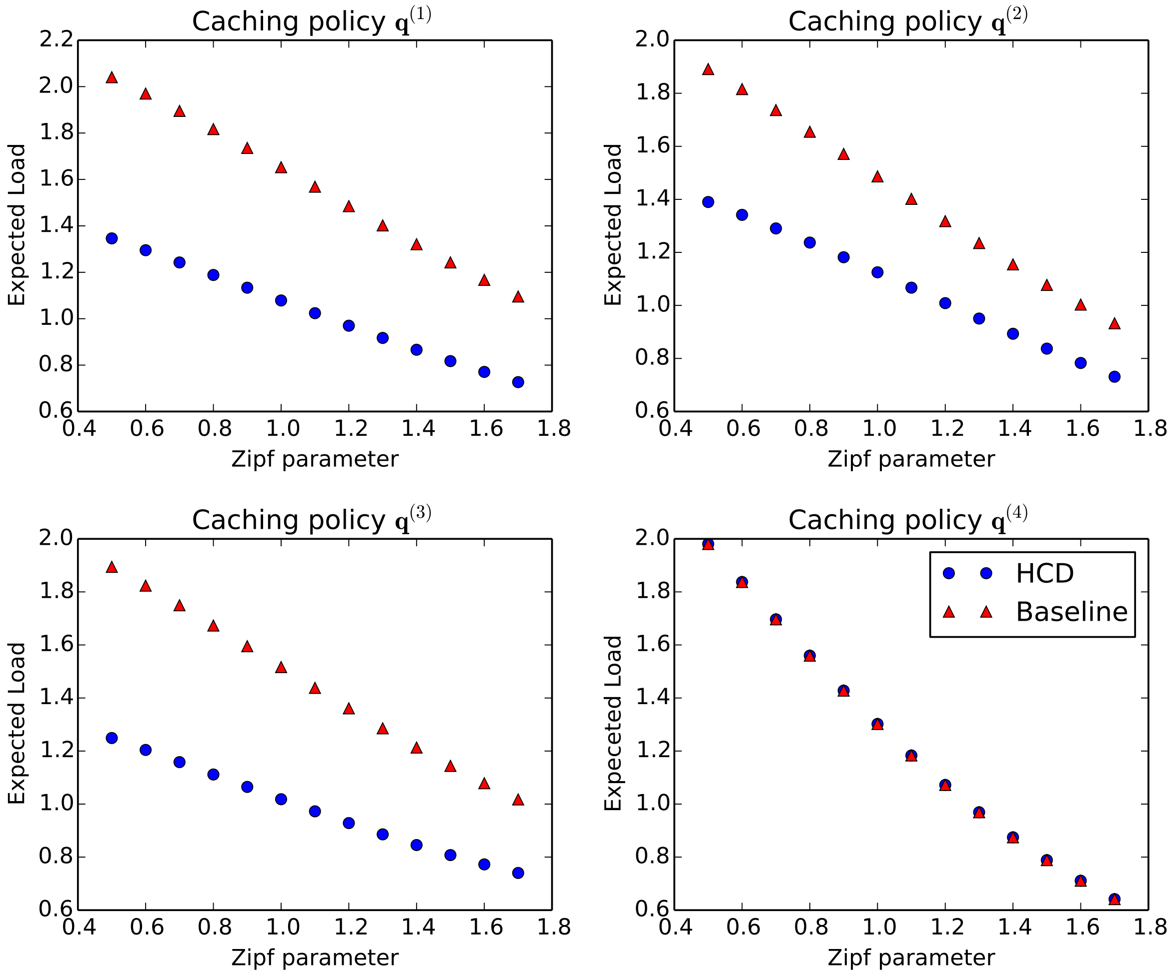


Figure 4.9: Plot of expected load for $N = 10, K = 4$ and $M = 5$ and centralized caching. $\mathbf{q}^{(1)}, \mathbf{q}^{(2)}, \mathbf{q}^{(3)}$: arbitrary multilevel caching polices - around 30% improvement can be noted. $\mathbf{q}^{(4)}$: two level caching scheme with levels 0 and 0.25 - no improvement.

between HCD and the one in [78] is the added step of borrowing bits from higher type subfiles to fill up the lower type ones. Consider the worst case scenario, where a subfile of very low type needs to borrow some bits, the algorithm will first access the next immediate higher type and will keep moving up to higher type until it is full. The algorithm will move to higher types only after exhausting all bits from the immediate higher type. So even if it ends up borrowing bits from the highest type subfiles, in the process it has covered all the bits for that user's demand.

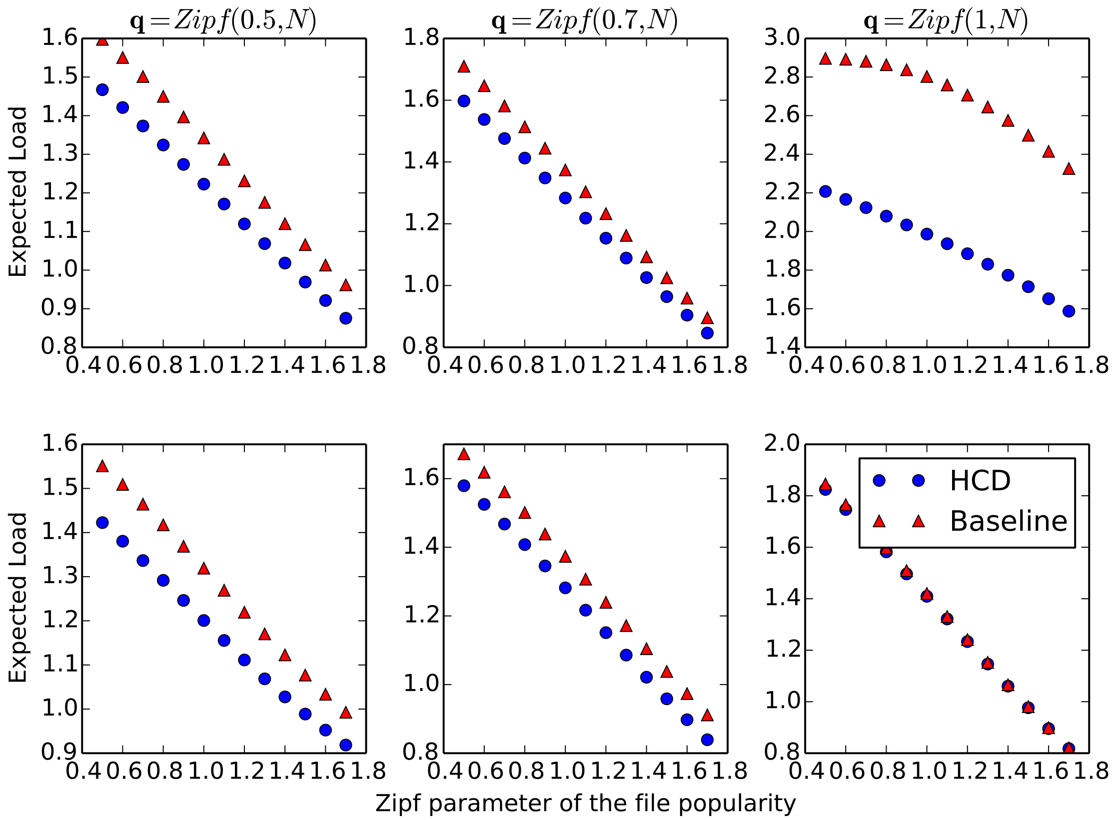


Figure 4.10: Plot of expected load for $N = 10, K = 4$ and $M = 5$ and decentralized caching. Top row: Zipf based caching policy; Bottom row: Zipf based, factor 2 grouping applied. Big gap in performance in the top right plot : due to $q_i M > 1$. Improvements of around 10% are seen in almost all cases.

4.6 Caching Policy Evaluation

In most real world scenarios the file popularities are not uniform; rather they vary over several orders of magnitude. In the uniform case each file had an equal popularity of $1/N$, thus the decision to allot $1/N$ fraction of the cache memory of every user to each file made perfect sense. The most logical way to extend this caching policy to the non-uniform case would be to allot different amount of cache memory to each file depending on the popularity of that file, but this breaks the symmetry of content placement and requires a new coding scheme in the delivery phase. In [76], the authors propose a caching policy and a coding scheme based on a grouping approach. They partition the files into groups with approximately uniform popularities and each file within the same group is allocated the same amount of cache memory. The symmetry within each group is preserved here and thus the demands within each group can be delivered using the same concept as in the uniform case. We think that this method is inefficient and can be improved upon significantly.

In the placement phase we propose to cache $p_i M$ fraction of file i at the every user. By relabeling the files, we can assume without loss of generality that $p_1 \geq p_2 \geq \cdots \geq p_N$. Note that files with higher popularity will have part of it stored in the cache of every user. The problem with this type of caching, as observed in [76], becomes apparent in the delivery phase. Due to varying size of the bins across different files, it would no longer be easy to code using the cliques technique explained in section 4.3.1 and we need the Heterogenous coded caching scheme.

4.6.1 Caching Tweaks

In section 4.3.2 we proposed to cache $p_i MF$ bits of file i at every user in the placement phase, in general file popularities are such that the resulting bin type ($t_i = P_i MK$) for caching these $p_i M$ fractions of the files won't actually be an integer. In order to overcome this inconvenience we will describe a small tweak in this section that would help us to cache the fraction we want while keeping the bin type close to the desired one too.

First lets look at what would happen if we just truncate the decimal part of the original t_i to get a integer value, *i.e.* $t'_i = \lfloor p_i MK \rfloor$. To get this value for t_i we will have to modify the amount of the file cached at each user to $p'_i M$, where $p'_i = \frac{t'_i}{MK}$. It is easy to see that the modified caching scheme does not fill up the cache since the fractions don't add up to the size of the cache, *i.e.* $\sum_i p'_i M \neq M$. In order to counter this truncation error and use the cache efficiently we propose the following scheme. Each file is split into two orthogonal parts, the first one will be of size αF bits and the second one will be the remaining $(1-\alpha)F$

bits. The intuition behind doing this is that we can apply the policy of caching $p_i'MF$ bits to one part and we can use a different caching policy to the other part, now by manipulating value of alpha and the caching policy used for the second part we could fill up the entire cache and be close to our desired bin type.

The part of the file with αF bits will follow a caching policy where some $p_i'M$ fraction of this part will be cached at each user, for the second part with $(1-\alpha)F$ bits we will cache βM fraction of it at each user. In order to stay close to the original bin type we can choose the new parameter β such that the bin type $t_i^{''}$ for the second part is equal to $t_i^{'}+1$. Now we have the following relations,

$$
\begin{aligned}
p_i'M\alpha F + \beta M(1-\alpha)F &= p_iMF && (4.4)\\
\beta(1-\alpha) &= p_i - \alpha p_i' && (4.5)\\
(1-\alpha)(\beta - p_i') &= (p_i - p_i') && (4.6)
\end{aligned}
$$

(4.6) represents a rectangular hyperbola with asymptodes parallel to co-ordinate axis, namely $\alpha = 1$ and $\beta = p_i'$. There are two choices of β (and corresponding α) that are of interest to us. We can either choose β such that βMK is close to $t_i^{'}$, *i.e.* $t''_i = \beta MK = t_i^{'}+1$, or we can choose β such that the part of the file that follows the original caching policy is kept as large as possible. For the latter, the value of β should be as big as possible, which is $1/M$ since $\beta MK \leq K$. Intuitively, we are splitting each file into two parts (α and $1-\alpha$) and cache them such that the first part could be split into bins of type $t_i^{'}$ and the second part into bins of type $t_i^{''} = \beta MK$.

4.6.2 Simulation and Inference

On running the simulation for zipf distributed file popularities, we found that the allotting cache memory based on file popularity might not be the optimal way to go. From the simulation results we can see that the new caching scheme is not necessarily better than uniform caching or greedy caching, the performance of the new caching policy along with the new coding scheme seems to always lie between the those of greedy and uniform schemes.

In evidence of these observations through simulation it becomes necessary to understand why the caching policy doesn't perform better and to do this lets start by looking at the single user case ($K = 1$). The expected rate of transmission for the single user case is as simple as calculating the weighted average for the code length required to transmit each file, the weight here being the popularity of that file. It is easy to see that caching the most popular files (greedy caching) is the optimal for the single user setting. The intuition here is that even if a fraction of the cache space allotted to a more popular file is used for caching a file of lesser popularity, it will increase the transmission length of the more popular file and decrease the transmission length of the less popular file, but the more popular file has higher weight thus this reallotment would only negatively affect the expected rate (as it increases the average transmission length). The optimality of greedy caching is not so apparent for the multiple user case. Let us look at a few multi user examples to get some intuition.

Example 4.2. *Consider a database with $N = 2$ files, each of size F bits, $K = 2$ users and a cache size of $M = 2$ files. Let p denote the popularity of file A and $(1-p)$ the popularity of file B. Now consider a caching scheme where the user caches a fraction of the file proportional the popularity of the corresponding file.*

For this example we try to assume that the two users will try to cache different parts of the files as possible, in other words the bits of the files cached by the two users will have as little overlap as possible. Lets call the part of file A that is cached by user 1 as A_1 and the part cached by user 2 as A_2, similarly B_1 the part of file B that is cached by user 1 and B_2 as the part cached by user 2. For notational convenience lets call $A_1 \cap A_2$ as A' and $B - (B_1 \cup B_2)$ as B'. The average length of code can be calculated as follows:

- When both users demand file A, they both already have the part A' in their cache, so the server needs to send $A_2 - A'$ and $A_1 - A'$ which can be coded together and effectively transmitted as $(1-p)F$ bits.
- When one of the users demand file A and the other demands file B, the server needs to send $A_2 - A'$ $(A_1 - A')$, B_1 $(B2)$ and B'. Among these $A_2 - A'$ and B_1 $(A_1 - A'$ and $B_2)$ can be coded together into $(1-p)F$ bits . Thus the server needs to send $(1-p)F + (2p-1)F$ bits.
- When both users demand file B, the server can code together B_1 and B_2 into $(1-p)F$ bits and B' will be sent separately. Thus the server needs to send $(1-p)F+(2p-1)F$ bits

$$\begin{aligned} avg &= p^2 * (1-p)F + 2p(1-p) * [(1-p)F + (2p-1)F] + (1-p)^2 * [(1-p)F + (2p-1)F] \quad (4.7) \\ &= [(1-p) + (1-p^2)(2p-1)]F\ bits \quad (4.8) \end{aligned}$$

Figure 4.11 shows the this avg rate plotted for various values of p along with the plot for the corresponding avg rate for greedy and uniform caching scheme.

Example 4.3. *For this example let's consider a database with 4 files , A, B, C and D. We will consider two sets of file popularities (as shown in table 4.5) in this example to see when the proposed scheme could do better.*

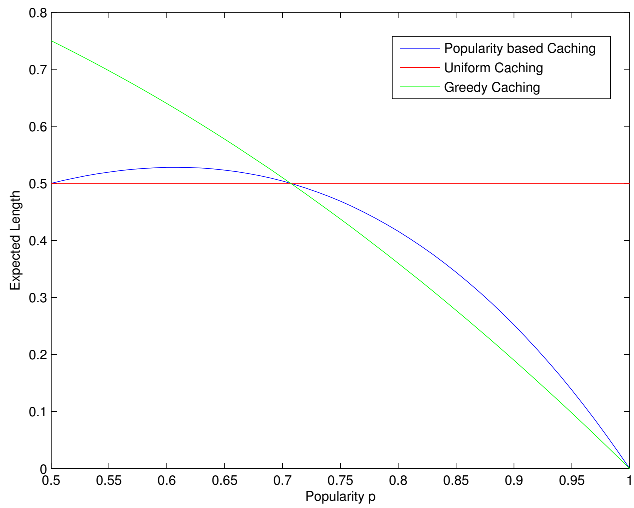


Figure 4.11: Plot showing the average code length for various values of p

Table 4.5

Files	Set 1	Set 2
A	0.3	0.32
B	0.3	0.32
C	0.3	0.32
D	0.1	0.04

The rates corresponding to the both the sets of popularity for various schemes are shown in table 4.6. The table also depicts the rate for a caching scheme that employs a type of uniform caching among the three equiprobable file A, B and C.

Table 4.6

	Greedy	Uniform	3 files Uniform	Proposed scheme ($t'_i = t_i + 1$)
Set 1	0.46	0.5	0.4119	0.4315
Set 2	0.3856	0.5	0.52	0.548

Let us explore an alternate scheme of caching uniformly over n files out of the total N files and try to get an intuition why such schemes don't perform better. For the following example we will consider a zipf factor of 0.7 for which we already know uniform does better, so we will compare this partially uniform scheme with the actually uniform caching scheme.

Example 4.4. *In this example we have $N = 20$, $M = 10$, $K = 3$ and additionally $n = 19$. The value of n represents the files over which the uniform scheme will be applied, in this case we will uniformly cache the first 19 most popular files. No part of the least popular file will be be cached. Lets analyze this further to understand why such partially uniform scheme might not help in improving the average length.*

Let p denote the popularity of the least popular file. The table 4.7 shows the rate for various types of combination of demanded files along with the probability of such a demand. The first row represents the case where all there users demand from the least popular file, we need to send the whole F bits of the file. The second row represents the case where two users demand the 20th file and the 3rd user demands one of the other files, here the server sends F bits of the 20th file and it only needs to send $0.473F$ not-cached bits of the other file. Similarly, the third row represents the case where only one user demands the least popular file and the other 2 users demand one of the other files, here again the server sends F bits of the least popular file in addition to $0.614F$ bits required for transmitting the coded bins. Finally, the last column represent the case where all 3 users demand one of the first 19 files , the server needs to send $0.614F$ bits of coded bins.

Table 4.7

Probability	Rate of Partially uniform scheme	Rate of uniform scheme
p^3	1	0.66
$3p^2(1-p)$	1.473	0.66
$3p(1-p)^2$	1.614	0.66
$(1-p)^3$	0.614	0.66
Expected length	0.678	0.66

From table 4.7, it is clear that gain for the partially uniform approach occurs only for the last case and this is why the final average rate is not better than uniform.

Another scheme to explorer is a hybrid, it tries to combine both the greedy and uniform strategies. We will check out two variations of the hybrid scheme here.

Hybrid-1: Take a major portion(α) of each file and employ uniform caching for this portion. Lets say αF bits of each file is considered and M/N of this portion will be cached at each user following the uniform caching policy. Now $(1-\alpha)MF$ bits of the cache is still empty and $(1-\alpha)F$ bits of each file hasn't been considered for any caching scheme. We can make use of this $(1-\alpha)MF$ bits of cache space such that the whole of the first few popular files gets cached at each user.

Example 4.5. *Consider $N = 20$, $M = 10$ and $K = 2$. For this example we will assume a value of 0.8 to α. As per the scheme some $0.5 * 0.8F = 0.4F$ bits of each file will be cached at each user. About $2F$ bits of the cache at each user is still unused, we can use that to cache the remaining $0.6F$ bits of the first 3 most popular files. Now let's look at the rate for various types of demands. Let p be the probability of the first 3 most popular files. This*

Table 4.8

Probability	Rate - Hybrid-1	Rate - uniform
p^2	0	0.5
$3p^2(1-p)$	0.6	0.5
$3p(1-p)^2$	0.6	0.5
Expected length (zipf=0.7)	0.5134	0.5
(zipf=1)	0.444	0.5

method seems have potential as we get an expected code length lower than uniform. Let's look at the gain this method provides compared to uniform approach:

$$Gain = p^2 * (-0.5) + 2p(1-p) * 0.1 + (1-p)^2 * 0.1 \tag{4.9}$$

$$= 0.1 - 0.6p^2 \tag{4.10}$$

A negative value for the gain implies that the code length of the hybrid scheme is lesser than uniform and in order to keep the gain negative the value of p has to be greater than 0.4802 for this example.

Hybrid-2: In this scheme we will first choose the number of files that we want to be cached fully at each user, let's call this n. Now $M' = M - n$ cache only remains at each user and $N' = N - n$ files still left. We will employ a uniform caching scheme for these modified parameters N', M' and K.

Example 4.6. *Consider* $N = 20$, $M = 10$ *and* $K = 2$. *Let's do greedy caching on the first 2 popular files,* i.e. $n = 2$. *Now,* $N' = N - n = 18$ *and* $M' = M - n = 8$. *These remaining 18 files are cached by following the uniform caching policy. Let* p *denote the combined popularity of the first* n *most popular files. Then the code length for various demand types can be calculated as shown in the table below.*

Table 4.9

Probability	Rate - Hybrid-2	Rate - uniform
p^2	0	0.5
$3p^2(1-p)$	0.5882	0.5
$3p(1-p)^2$	0.5882	0.5
Expected length (zipf=0.7)	0.5032	0.5
(zipf=1)	0.4354	0.5

The gain for the scheme can be derived similar to last example.

For this scheme we can also derive a closed form expression of the expected length as follows:

$$\textit{Expected length} = (1-p^2)\frac{K - \frac{KM'}{N'}}{1 + \frac{KM'}{N'}} = (1-p^2)\frac{K(N-M)}{N + KM - n(K+1)}$$

Would we be able to lower the code length using these schemes for more number of users?, this still remains to be seen.

4.7 Summary

The coded caching problem consists of two phases: one involves optimization of the caching policy and the other is the design of the delivery scheme that minimizes the load on the shared link for any given demand. The caching policy optimization is a very interesting problem, with recent works from [78, 79] showing some order optimality results, but *not* the focus of this chapter. Given the cache content and the demands of the users, the delivery phase optimization is basically an instance of the index coding problem, which is known to be NP-hard. We provide a practical algorithm, called the *Heterogenous Coded Delivery (HCD)* scheme, that performs significantly better than the current state-of-the- art scheme in coded caching for all multilevel (more than two) caching policies. An open question for future work is whether multiple levels are necessary for optimal caching policies, or two levels are sufficient.

Chapter 5

Conclusion

In this thesis, we tackle a very important open problem of multiple unicasts. Due to the obvious complexity of the problem, and the fact that it still remains an open problem, we focus primarily on precoding based techniques. We started by considering three unicast sessions in the SISO scenario (i.e., with min-cut one per session) and following a precoding-based approach (all precoding is performed at the end nodes, while intermediate nodes perform random network coding). This is the simplest, yet highly non-trivial instance of the general problem of network coding across multiple unicasts. Apart from being of interest on its own right, we hope that it can be used as a building block and provide insight into the general problem. We applied a alignment based scheme called the precoding-based network alignment (PBNA) and discussed the feasibility conditions for achievability of the desired rates under this scheme. More importantly we gave a graph theoretic characterization of these feasibility conditions which in turn gave us the necessary insight and helped us to rigorously prove them in [47, 48].

We also compared alignment to alternative approaches that can achieve half the rate depending on the network topology. For three unicast sessions with min-cut one, we show a negative result: whenever alignment is possible, alternative approaches can also achieve half the min-cut. However, for more than three sessions and/or for min-cut per session greater than one, we show examples where alignment is necessary. We also classification of the network based in the network structure and compare the rates achievable by PBNA to routing.

There are still many problems that remain to be solved regarding applying interference alignment techniques to this network setting. For example, one important problem is the complexity of PBNA, which arises in two aspects, i.e., precoding matrix and field size, and is inherent in the framework of PBNA. One direction for future work is to apply other alignment techniques (with lower complexity) to the network setting, like alignment by network code design in the middle of the network.

The extensions to other network scenarios beyond SISO or more than three unicast sessions are highly non-trivial. Applying alignment to more that three sessions gets very complicated, so we in order to simplify things we focused on the interference channel topology where the network transfer matrix is rank deficient. The rank deficiency in the transfer matrix causes algebraic dependencies between the channel coefficients and this provides an interesting problem to solve. We studied the problem of characterizing the DoF for the K-user interference channel with rank deficient transfer matrix. We presented the optimal DoF for the basic cases of $K = 4$ and $K = 5$ and illustrated the difficulties involved in finding the same for $K \geq 6$. In conclusion, finding the optimal DoF of the general K user interference channels with rank deficient transfer matrix still remains open and presents a considerable challenge.

We also looked into the problem of caching in content distribution networks with caches at the end users. Another multiple unicast problem that is of great interest in the information theory community is the index coding problem. The side information provided by the content of the cache induces index coding instances. The coded caching problem consists of two phases: one involves optimization of the caching policy and the other is the design of the delivery scheme that minimizes the load on the shared link for any given demand. The caching policy optimization is a very interesting problem, with recent works from [78, 79] showing some order optimality results, but *not* the focus of this thesis. Given the cache content and the demands of the users, the delivery phase optimization is basically an instance of the index coding problem, which is known to be NP-hard. We provide a practical algorithm, called the *Heterogenous Coded Delivery (HCD)* scheme, that performs significantly better than the current state-of-the- art scheme in coded caching for all multilevel (more than two) caching policies. An open question for future work is whether multiple levels are necessary for optimal caching policies, or two levels are sufficient.

Bibliography

[1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *Information Theory, IEEE Transactions on*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[2] R. Koetter and M. Medard, "An algebraic approach to network coding," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[3] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proc. of IEEE INFOCOM*, Mar. 2005.

[4] M. Wang and B. Li, "r^2: Random push with random network coding in live peer-to-peer streaming," *in the IEEE Journal on Selected Areas in Communications, Special Issue on Advances in Peer-to-Peer Streaming Systems*, vol. 25, pp. 1655–1666, Dec. 2007.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, pp. 243–254, 2006.

[6] N. Harvey, R. Kleinberg, and A. Lehman, "Comparing network coding with multicommodity flow for the k-pairs communication problem," *Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, http://theory. csail. mit. edu/nickh/Publications/Gaps/TR964. pdf, Tech. Rep*, 2004.

[7] ——, "On the capacity of information networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2345–2364, 2006.

[8] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Medard, "Network coding for multiple unicasts: An approach based on linear optimization," July 2006, pp. 1758–1762.

[9] C. Fragouli and E. Soljanin, "Monograph on Network Coding: Fundamentals and Applications," *Foundations and Trends in Networking*, vol. 2, no. 1.

[10] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," in *Allerton Conference on Communications*. Citeseer, 2004.

[11] V. Cadambe and S. Jafar, "Interference alignment and the degrees of freedom of the k user interference channel," *IEEE Trans. on Information Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[12] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *Information Theory, IEEE Transactions on*, vol. 49, no. 2, pp. 371–381, Feb 2003.

[13] T. Ho and D. Lun, *Network Coding, An Introduction.* Cambridge University Press, 2008.

[14] C. Fragouli and E. Soljanin, "Monograph on Network Coding: Fundamentals and Applications," in *Foundations and Trends in Networking.* NOW Publishers, 2007, vol. 2.

[15] A. Rasala-Lehman and E. Lehman, "Complexity classification of network information flow problems," in *15th Annual ACM-SIAM SODA*.

[16] S. Riis, "Linear versus non-linear boolean functions in network flow," in *Proc. of CISS.*

[17] M. Médard, M. Effros, T. Ho, and D. Karger, "On coding for nonmulticast networks," in *Proc. of 41st Allerton Conference*, Oct 2003.

[18] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2745 – 2759, Aug. 2005.

[19] X. Yan, J. Yang, and Z. Zhang, "An outer bound for multisource multisink network coding with minimum cost consideration," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2373–2385, 2006.

[20] X. Yan, R. W. Yeung, and Z. Zhang, "The capacity region for multi-source multi-sink network coding," *Information Theory, IEEE International Symposium on*, pp. 116 –120, June 2007.

[21] R. Yeung, S. Li, N. Cai, and Z. Zhang, "Network coding theory part ii: multiple source," *Foundations and Trends in Communications and Information Theory*, vol. 2, no. 5, pp. 330–381, 2005.

[22] A. Lehman, "Network Coding," Ph.D. dissertation, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, 2005.

[23] R. Dougherty, C. Freiling, and K. Zeger, "Networks, matroids, and non-Shannon information inequalities," *Information Theory, IEEE Transactions on*, vol. 53, no. 6, pp. 1949–1969, 2007.

[24] ——, "Linear network codes and systems of polynomial equations," *Information Theory, IEEE Transactions on*, vol. 54, no. 5, pp. 2303– 2316, May 2008.

[25] A. Subramanian and A. Thangaraj, "A simple algebraic formulation for the scalar linear network coding problem," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on.* IEEE, 2009, pp. 177–184.

[26] C. Wang and N. Shroff, "Random linear intersession network coding with selective cancelling," in *Information Theory Workshop, 2009. ITW 2009. IEEE.* IEEE, 2009, pp. 559–563.

[27] M. Kim, M. Medard, U.-M. O'Reilly, and D. Traskov, "An evolutionary approach to inter-session network coding," in *Proc. of IEEE INFOCOM*, Apr. 2009, pp. 450 –458.

[28] C. Wang and N. Shroff, "Beyond the butterfly-a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions," in *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*. IEEE, 2008, pp. 121–125.

[29] K. Jain, V. Vazirani, and G. Yuval, "On the capacity of multiple unicast sessions in undirected graphs," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2805–2809, 2006.

[30] G. Kramer and S. Savari, "Edge-cut bounds on network coding rates," *Journal of Network and Systems Management*, vol. 14, no. 1, pp. 49–67, 2006.

[31] A. Eryilmaz and D. Lun, "Control for inter-session network coding," in *Proc. Workshop on Network Coding, Theory & Applications*. Citeseer, 2007.

[32] Y. Wu, "On constructive multi-source network coding."

[33] N. Harvey, K. Jain, L. Lau, C. Nair, and Y. Wu, "Conservative network coding," in *Proc. 44th Annual Allerton Conf. on Comm., Contr., and Computing*. Citeseer.

[34] T. Cui, L. Chen, and T. Ho, "Energy efficient opportunistic network coding for wireless networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008, pp. 361–365.

[35] A. Khreishah, C. Wang, and N. Shroff, "Optimization based rate control for communication networks with inter-session network coding," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008, pp. 81–85.

[36] S. Omiwade, R. Zheng, and C. Hua, "Butterflies in the mesh: lightweight localized wireless network coding," in *Proc. of NetCod*, Jan. 2008.

[37] M. Effros, T. Ho, and K. S., "A tiling approach to netwrk code design for wireless networks," in *Proc. of IEEE Information Theory Workshop (ITW)*, Mar. 2006.

[38] S. Jafar and S. Shamai, "Degrees of freedom region for the MIMO X channel," *IEEE Trans. on Information Theory*, vol. 54, no. 1, pp. 151–170, Jan. 2008.

[39] V. R. Cadambe, S. A. Jafar, and C. Wang, "Interference alignment with asymmetric complex signaling - settling the host-madsen-nosratinia conjecture," in *arXiv:0904.0274*, 2009.

[40] S. Gollakotta, S. Perli, and D. Katabi, "Interference alignment and cancellation," *Proceedings of ACM SIGCOMM*, 2009.

[41] B. Nazer, M. Gastpar, S. A. Jafar, and S. Vishwanath, "Ergodic interference alignment," June 2009.

[42] V. Cadambe and S. Jafar, "Reflections of interference alignment and the degrees of freedom of the K user interference channel," *IEEE Theory Society Newsletter*, vol. 59, no. 4, pp. 5–9, 2009.

[43] S. A. Jafar, "The ergodic capacity of interference networks," e-print,arxiv:0902.0838.

[44] A. Das, S. Vishwanath, S. Jafar, and A. Markopoulou, "Network coding for multiple unicasts: An interference alignment approach," *Proc. IEEE International Symposium on Information Theory. arXiv:1008.0235*, 2010.

[45] V. Cadambe and S. Jafar, "Parallel gaussian interference channels are not always separable," *IEEE Trans. on Info. Theory*, vol. 55, pp. 3983–3990, Sep. 2009.

[46] A. Ramakrishnan, A. Das, A. Markopoulou, S. Jafar, and S. Vishwanath, "Network Coding for Three Unicast Sessions: Interference Alignment Approaches," *Proc. Allerton Conference on Commun. Control and Computing*, 2010.

[47] C. Meng, A. Ramakrishnan, A. Markopoulou, and S. A. Jafar, "On the feasibility of precoding-based network alignment for three unicast sessions," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on.* IEEE, 2012, pp. 1907–1911.

[48] C. Meng, A. Das, A. Ramakrishnan, S. Jafar, A. Markopoulou, and S. Vishwanath, "Precoding-based network alignment for three unicast sessions," *Information Theory, IEEE Transactions on*, vol. 61, no. 1, pp. 426–451, Jan 2015.

[49] J. Han, C. Wang, and N. Shroff, "Analysis of precoding-based intersession network coding and the corresponding 3-unicast interference alignment scheme," Technical Report, Purdue University, http://web. ics. purdue. edu/ han83, Tech. Rep., 2011.

[50] A. Ramakrishnan, A. Das, H. Maleki, A. Markopoulou, S. Jafar, and S. Vishwanath, "Network Coding for Three Unicast Sessions: Interference Alignment Approaches."

[51] C. Wang, T. Gou, and S. Jafar, "Subspace alignment chains and the degrees of freedom of the three user MIMO interference channel," *Arxiv preprint ArXiv:1109.4350*, September 2011. [Online]. Available: http://arxiv.org/pdf/1109.4350v1

[52] S. Jafar and M. Fakhereddin, "Degrees of freedom for the MIMO interference channel," *IEEE Transactions on Information Theory*, vol. 53, no. 7, pp. 2637–2642, July 2007.

[53] T. Gou and S. Jafar, "Degrees of freedom of the K user M $\times$ N MIMO interference channel," *IEEE Trans. on Information Theory*, vol. 56, no. 12, pp. 6040–6057, December 2010.

[54] S. H. Chae and S.-Y. Chung, "On the degrees of freedom of rank deficient interference channels," in *2011 IEEE International Symposium on Information Theory Proceedings (ISIT)*, 2011, pp. 1367–1371.

[55] S. R. Krishnamurthy and S. A. Jafar, "Degrees of freedom of 2-user and 3-user rank-deficient MIMO interference channels," in *Global Communications Conference (GLOBECOM), 2012 IEEE*. IEEE, 2012, pp. 2462–2467.

[56] S. H. Chae, C. Suh, and S.-Y. Chung, "Degrees of freedom of the rank-deficient interference channel with feedback," *CoRR*, vol. abs/1307.1461, 2013.

[57] Y. Zeng, X. Xu, Y. L. Guan, and E. Gunawan, "On the achievable degrees of freedom for the 3-user rank-deficient MIMO interference channel," *CoRR*, vol. abs/1211.4198, 2012.

[58] A. K. Das, S. Vishwanath, S. A. Jafar, and A. Markopoulou, "Network coding for multiple unicasts: An interference alignment approach," *CoRR*, vol. abs/1008.0235, 2010. [Online]. Available: http://arxiv.org/abs/1008.0235

[59] A. Ramakrishnan, A. Das, H. Maleki, A. Markopoulou, S. Jafar, and S. Vishwanath, "Network Coding for Three Unicast Sessions: Interference Alignment Approaches," *Allerton Conference on Communications, Control and Computing*, October 2010.

[60] A. Agustin and J. Vidal, "Degrees of freedom region of the MIMO X channel with an arbitrary number of antennas," *Arxiv preprint ArXiv:1210.2582*, October 2012. [Online]. Available: http://arxiv.org/pdf/1210.2582v1

[61] V. Annapureddy and A. El Gamal and V. Veeravalli, "Degrees of Freedom of Interference Channels With CoMP Transmission and Reception," *IEEE Trans. on Information Theory*, vol. 58, no. 9, pp. 5740–5760, Sep. 2012.

[62] S. R. Krishnamurthy, A. Ramakrishnan, and S. A. Jafar, "Degrees of freedom of rank-deficient MIMO interference networks," *Submitted to IEEE Trans. on Information Theory*. [Online]. Available: http://escholarship.org/uc/item/1c17p6zq

[63] D. Cox, J. Little, and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, ser. Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra. Springer, 2007, no. v. 10. [Online]. Available: http://books.google.com/books?id=yCsDO425PC0C

[64] W. Hodge, W. Hodge, and D. Pedoe, *Methods of Algebraic Geometry:*, ser. Cambridge Mathematical Library. Cambridge University Press, 1994. [Online]. Available: http://books.google.com/books?id=bJwbn3RSWhwC

[65] S. A. Jafar, "Interference alignment : A new look at signal dimensions in a communication network," *Foundations and Trends in Communications and Information Theory*, vol. 7, no. 1, pp. 1–134, 2011. [Online]. Available: http://dx.doi.org/10.1561/0100000047

[66] A. Ramakrishnan, S. R. Krishnamurthy, S. A. Jafar, and Y. Yu, "Degrees of freedom of interference channel with rank-deficient transfer matrix," in *Information Theory (ISIT), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 1221–1225.

[67] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 1. IEEE, 1999, pp. 126–134.

[68] A. Leff, J. L. Wolf, and P. S. Yu, "Replication algorithms in a remote caching architecture," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 4, no. 11, pp. 1185–1204, 1993.

[69] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman, "Placement algorithms for hierarchical cooperative caching," *Journal of Algorithms*, vol. 38, no. 1, pp. 260–302, 2001.

[70] I. Baev, R. Rajaraman, and C. Swamy, "Approximation algorithms for data placement problems," *SIAM Journal on Computing*, vol. 38, no. 4, pp. 1411–1429, 2008.

[71] A. Meyerson, K. Munagala, and S. Plotkin, "Web caching using access statistics," in *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2001, pp. 354–363.

[72] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[73] A. Wolman, M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy, "On the scale and performance of cooperative web proxy caching," in *ACM SIGOPS Operating Systems Review*, vol. 33, no. 5. ACM, 1999, pp. 16–31.

[74] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *Networking, IEEE/ACM Transactions on*, vol. 16, no. 6, pp. 1447–1460, 2008.

[75] D. Lee, J. Choi, J.-H. Kim, S. H. Noh, S. L. Min, Y. Cho, and C. S. Kim, "LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies," *IEEE transactions on Computers*, vol. 50, no. 12, pp. 1352–1361, 2001.

[76] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *arXiv preprint arXiv:1209.5807*, Sept. 2012.

[77] ——, "Decentralized caching attains order-optimal memory-rate tradeoff," *arXiv preprint arXiv:1301.5848*, Jan. 2013.

[78] U. Niesen and M. A. Maddah-Ali, "Coded caching with nonuniform demands," *arXiv preprint arXiv:1308.0178*, Aug. 2013.

[79] M. Ji, A. M. Tulino, J. Llorca, and G. Caire, "Order optimal coded caching-aided multicast under zipf demand distributions," *arXiv preprint arXiv:1402.4576*, 2014.

[80] Z. Bar-Yossef, Y. Birk, T. Jayram, and T. Kol, "Index coding with side information," *Information Theory, IEEE Transactions on*, vol. 57, no. 3, pp. 1479–1494, 2011.

[81] Y. Birk and T. Kol, "Coding on demand by an informed source (ISCOD) for efficient broadcast of different supplemental data to caching clients," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2825–2830, 2006.

[82] M. Langberg and A. Sprintson, "On the hardness of approximating the network coding capacity," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on.* IEEE, 2008, pp. 315–319.

Appendices

A Network Alignment

A.1 Schwartz-Zippel Lemma

Schwartz-Zippel lemma is a tool commonly used in probabilistic polynomial identity testing, *i.e.,* in the problem of determining whether a given multivariate polynomial is a zero-polynomial or identically equal to 0. It bounds the probability that a non-zero polynomial will have roots at randomly selected test points. The formal statement is as follows:

Lemma A.1. *Lemma Let $P(x_1, \ldots, x_n)$ be an n-variate polynomial of degree exactly d over a field $\mathbb{F}$, $P(x_1, \ldots, x_n)$ is not identically zero. Let $\mathbb{S}$ be a non-empty finite subset of the field $\mathbb{F}$ and let $r_1, r_2, \ldots, r_n$ be selected randomly from $\mathbb{S}$. then*

$$\Pr[P(r_1, r_2, \ldots, r_n) = 0] \leqslant \frac{d}{|\mathbb{S}|}$$

The lemma has an inductive proof. For univariate polynomials, *i.e.,* $m = 1$, the lemma follows directly from the fact that a polynomial of degree d can have no more than d roots. The lemma works for any number of variables $m \geq 1$.

A.2 Proof for Graph Theoretic Representation

For the sake of notational convenience, we will omit the argument term, $(\underline{\xi})$, in the rest of the proof. Also, let us use m'_{ii} and m''_{ii} to represent the terms $\frac{m_{ij}m_{ki}}{m_{kj}}$ and $\frac{m_{ji}m_{ik}}{m_{jk}}$ respectively. We also use m_{e_1,e_2} to represent the polynomial coefficient from node/edge e_1 to node/edge e_2.

Proof of Theorem 2.4. First, we prove the "if" part of the theorem. It is given that α_{kij} is a bottleneck between S_i and D_j and α_{jik} is a bottleneck between S_i and D_k. With this, it is easy to see that m'_{ii} reduces to $m_{S_1,\alpha_{jik}}m_{\alpha_{jik},D_1}$ and m''_{ii} reduces to $m_{S_1,\alpha_{kij}}m_{\alpha_{kij},D_1}$. Hence, $m'_{ii} + m''_{ii} = m_{S_1,\alpha_{jik}}m_{\alpha_{jik},D_1}$ and $m''_{ii} + m_{S_1,\alpha_{kij}}m_{\alpha_{kij},D_1}$. It is also given that α_{kij} is parallel to α_{jik} and$\{\alpha_{jik}, \alpha_{kij}\}$ forms a cut-set between S_i and D_i, this implies m'_{ii} and m''_{ii} represent two set of disjoint paths connecting S_i to D_i, thus proving $m_{ii} = m'_{ii} + m''_{ii}$.

Next, we prove the converse part of the theorem. Now, it is given that $m_{ii} = m'_{ii} + m''_{ii}$. If α_{kij} is a bottleneck between S_i and D_j and α_{jik} is not a bottleneck between S_i and D_k, then m'_{ii} is a rational function whose denominator is a non-constant polynomial, while m''_{ii} is a polynomial. This leads to $m'_{ii} + m''_{ii}$ being a rational function with a non-constant denominator polynomial, and since m_{ii} is a polynomial we can clearly see that $m_{ii} \neq m'_{ii}+m''_{ii}$, which is a contradiction. The same argument can be made for when α_{kij} is not a bottleneck between S_i and D_j and α_{jik} is a bottleneck between S_i and D_k.

Now let us assume α_{kij} is not a bottleneck between S_i and D_j and α_{jik} is also not a bottleneck between S_i and D_k. We have $m'_{ii} = \frac{m_{S_i,\beta_{jik}}m_{\alpha_{jik},D_1}}{m_{\alpha_{jik},\beta_{jik}}}$ and $m''_{ii} = \frac{m_{S_1,\beta_{kij}}m_{\alpha_{kin},D_i}}{m_{\alpha_{kij},\beta_{kij}}}$. From Theorem 2.3, we also know that $\alpha_{jik} \neq \alpha_{kij}$ or $\beta_{jik} \neq \beta_{kij}$. This implies that one of the following must hold:

1. There exists an irreducible polynomial $m_{ee'}$, such that $m_{ee'}$ is divisible by $m_{\alpha_{jik},\beta_{jik}}$ but not divisible by $m_{\alpha_{kij},\beta_{kij}}$.

2. There exists an irreducible polynomial $m_{ee'}$, such that $m_{ee'}$ is divisible by $m_{\alpha_{kij},\beta_{kij}}$ but not divisible by $m_{\alpha_{jik},\beta_{jik}}$.

We will use $\mathrm{lcm}(p, q)$ to denote the least common multiple of the two polynomials p and q. We also define the following polynomials :

$$
\begin{aligned}
f &= \mathrm{lcm}(m_{\alpha_{jik},\beta_{jik}}, m_{\alpha_{kij},\beta_{kij}}) \\
f_1 &= f/m_{\alpha_{jik},\beta_{jik}} \\
f_2 &= f/m_{\alpha_{kij},\beta_{kij}} \\
m'_{ii} + m''_{ii} &= \frac{m_{S_1,\beta_{jik}} m_{\alpha_{jik},D_1} f_1 + m_{S_1,\beta_{kij}} m_{\alpha_{kij},D_1} f_2}{f}
\end{aligned}
$$

Consider the first case. We have $m_{ee'}$ divisible by $m_{\alpha_{jik},\beta_{jik}}$ but not divisible by $m_{\alpha_{kij},\beta_{kij}}$. Moreover, we also know $\gcd(m_{\alpha_{jik},\beta_{jik}}, m_{S_1,\beta_{kij}} m_{\alpha_{kij},D_1}) = 1$. This implies $m_{ee'}$ is not divisible by $m_{S_1,\beta_{kij}} m_{\alpha_{kij},D_1}$, which intern implies $m_{ee'}$ is not divisible by $m_{S_1,\beta_{jik}} m_{\alpha_{jik},D_1} f_1 + m_{S_1,\beta_{kij}} m_{\alpha_{kij},D_1} f_2$. However, $m_{ee'}$ is divisible by f. This shows that $m'_{ii} + m''_{ii}$ is a rational function with non-constant denominator. Thus showing $m_{ii} \neq m'_{ii} + m''_{ii}$, which is a contradiction to our initial assumption. We can use similar arguments for the second case. Thus proving α_{kij} is a bottleneck between S_i and D_j and α_{jik} is a bottleneck between S_i and D_k. This shows that $m_{ii} = m_{S_i,\beta_{jik}} m_{\alpha_{jik},D_i} + m_{S_i,\beta_{kij}} m_{\alpha_{kij},D_i}$. Hence any path connecting source S_i to sink D_i should pass through either α_{jik} or α_{kij}, implying that $\{\alpha_{jik}, \alpha_{kij}\}$ forms a cut set separating S_i from D_i. Also, it is obvious that α_{kij} is parallel to α_{jik}, as there won't be two disjoint paths connecting S_i to D_i if they are not parallel. $\square$

B Rank Deficient Transfer Matrix

B.1 Achievability for Other Points of the Outer Bound

In this section we will show how to prove the *DoF* achievability when the rank of the transfer matrix is either less than $\lceil \frac{K}{2} \rceil$ or greater than $\lceil \frac{K}{2} \rceil$, *i.e.* $D < \lceil \frac{K}{2} \rceil$ or $D > \lceil \frac{K}{2} \rceil$, provided we could show the achievability for $D = \lceil \frac{K}{2} \rceil$. We will assume that DoF of 1/2 per user is achievable when $D = \lceil \frac{K}{2} \rceil$ throughout this section.

First let us consider $D < \lceil \frac{K}{2} \rceil$ case, the DoF outer bound here is simply D. The achievability for this case is based on the achievable scheme for a $2D$ user interference channel with rank D transfer matrix. Consider the K user channel, if we choose only $2D$ users among the K user to be active at any given time, it reduces to the case of $D =$ Number of users/2 thus making it possible to achieve the sum *DoF* of D. Symmetric DoF can be achieved by cycling through the $\binom{K}{2D}$ combinations of active users. Here each user is active in $\binom{K-1}{2D-1}$ combinations and a *DoF* of half per user is assumed to be achievable in combination, thus each user will get total *DoF* of $\frac{1}{2} \times \frac{\binom{K-1}{2D-1}}{\binom{K}{2D}} = \frac{D}{K}$.

Now consider the $D > \lceil \frac{K}{2} \rceil$ case, we will show that a sum *DoF* of $K/2$ is achievable using the scheme from section 3.5.1 provided it works for $D = \lceil \frac{K}{2} \rceil$. More specifically, we will assume that the direct channels cannot be represented as a rational function of the cross channels. We will make use of concepts of Varieties and Ideals from algebraic geometry to prove this case. We know that the determinant of any $D + 1 \times D + 1$ sub matrix is going to be zero. Let V_D denote the *affine variety* generated by these determinant polynomials. Without loss of generality consider the case of $D = \lceil \frac{K}{2} \rceil + 1$, $V_{\lceil \frac{K}{2} \rceil + 1}$ denotes the *affine variety* and $I_{\lceil \frac{K}{2} \rceil + 1}$ can be used to represent the ideal generated by this variety, *i.e.* $I_{\lceil \frac{K}{2} \rceil + 1} = I(V_{\lceil \frac{K}{2} \rceil + 1})$. The following are true,

$$V_{\lceil \frac{K}{2} \rceil} \subseteq V_{\lceil \frac{K}{2} \rceil + 1} \tag{B.1}$$

The above equation implies that,

$$I_{\lceil \frac{K}{2} \rceil} \supseteq I_{\lceil \frac{K}{2} \rceil + 1} \tag{B.2}$$

Now assume, under $D = \lceil \frac{K}{2} \rceil + 1$, the direct channels can be expressed as a rational function of the cross channels. This tells us that there exists a polynomial, let us call it f_1, of the form $q(X)H_{kk} - p(X)$ that evaluates to 0, *i.e.* $f_1 = 0$, for any realization in the variety, where X is the vector of cross channels and $p(X)$ and $q(X)$ are multivariate polynomials in the cross channels. Since $f_1 = 0$ under the affine variety, we get $f_1 \in I_{\lceil \frac{K}{2} \rceil + 1}$, which implies $f_1 \in I_{\lceil \frac{K}{2} \rceil}$. This contradict our primary assumption that direct channels cannot be expressed as a rational function of the cross channels under $D = \lceil \frac{K}{2} \rceil$, thus we can conclude that direct channels cannot be expressed as a rational function of the cross channels even under $D = \lceil \frac{K}{2} \rceil + 1$. This argument can be extended to all cases of $D > \lceil \frac{K}{2} \rceil$.

B.2 Schwartz-Zippel Lemma for the Variety

We have a transfer matrix $\mathbf{H} = [H_{ij}]$ of size $K \times K$ and rank D. The sample space, $\mathbb{S}$ here is the set of all channel realizations for which the rank of the transfer matrix does not exceed D, and as seen in Appendix B.1 this is same as $\mathbb{V}_D$. Since the sample space $\mathbb{S}$ is an affine variety and not a field, it is not clear how Schwartz-Zippel lemma would be applicable here. In this section we will show that the Schwartz-Zippel lemma is valid even under $\mathbb{S}$.

The transfer matrix $\mathbf{H}$ can be written as the product of a $K \times D$ matrix $\mathbf{G}$ and a $D \times K$ matrix $\mathbf{F}$.

$$
\begin{aligned}
\mathbf{H} &= \mathbf{G}_{|K \times D|} * \mathbf{F}_{|D \times K|} \\
&= \begin{bmatrix} g_{11} & \cdots & g_{1D} \\ g_{21} & \ddots & g_{2D} \\ \vdots & \ddots & \vdots \\ g_{K1} & \cdots & g_{KD} \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1K} \\ \vdots & \ddots & \ddots & \vdots \\ f_{D1} & f_{D2} & \cdots & f_{DK} \end{bmatrix}
\end{aligned}
$$

Each of the channel coefficient (H_{ij}) in $\mathbf{H}$ can be expressed as a polynomial of certain g_{ij}'s and f_{ij}'s, namely $H_{ij} = \sum_{k=1}^{D} g_{ik} f_{kj}$, this is a parametric representation of H_{ij}. This implies that all realizations of $\mathbf{H}$ are given by $\mathbf{G} * \mathbf{F}$, as g_{ij}'s and f_{ij}'s vary over $\mathbb{C}$.

Consider a non-trivial polynomial in H_{ij}'s, that evaluates to zero always. Substituting H_{ij} with its parametric form, we can see that the polynomial will trivially reduce to zero. But for a non-zero polynomial in H_{ij}'s, we can see that the parameteric representation of the polynomial will not trivially reduce to zero, i.e., we will have a non-zero polynomial in f_{ij}'s and g_{ij}'s instead. Since the f_{ij}'s and g_{ij}'s take values from the filed $\mathbb{C}$, we can now use Schwartz-Zippel lemma for any polynomials in these variables.

B.3 Algebraic Independence of Channels

For a K user interference channel with a transfer matrix of rank $D = \lceil K/2 \rceil$, the transfer metric $\mathbf{H}$ can be written as the product of a $K \times D$ matrix $\mathbf{G}$ and a $D \times K$ matrix $\mathbf{F}$.

$$
\begin{aligned}
\mathbf{H} &= \mathbf{G}_{|K\times D|} * \mathbf{F}_{|D\times K|} \\
&= \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1D} \\ \vdots & \ddots & \ddots & \vdots \\ g_{K1} & g_{K2} & \cdots & g_{KD} \end{bmatrix} \begin{bmatrix} f_{11} & \cdots & f_{1K} \\ f_{21} & \ddots & f_{2K} \\ \vdots & \ddots & \vdots \\ f_{D1} & \cdots & f_{DK} \end{bmatrix}
\end{aligned}
$$

The elements of $\mathbf{G}$ and $\mathbf{F}$ form an algebraically independent set since they are generic variables. We have $K(K-1)$ cross channel coefficients each of which can be expressed as a polynomial function of the generic variables g_ij's and f_{ij}'s. For the sake of convenience let's represent the cross channel coefficients as T_i's, we have have set of multivariate polynomials $T_1, T_2, \ldots, T_{K(K-1)} \in \mathbb{C}\left[f_{11}, f_{12}, \ldots, f_{DK}, g_{11}, g_{12}, \ldots, g_{KD}\right]$.

$K = 4$ **and** $D = 2$

Proof of lemma 3.6. The transfer matrix for the 4 user case has 12 cross channels and there are 16 generic variables as the rank $D = 2$. In order to prove the 12 cross channels are algebraically independent, we write down the 12×16 Jacobian matrix as described in lemma 3.2, since we have 12 cross channels and 16 independent variables from $\mathbf{G}$ and $\mathbf{F}$. It can be noted that in $\mathbf{H} = \mathbf{G} \times \mathbf{F}$, $\mathbf{G}$ is a matrix of size 4×2 and $\mathbf{F}$ is a matrix of size 2×4. Each of the 12 cross channels is a polynomial function of these 16 independent variables. Let us denote the 16 variables of $\mathbf{G}, \mathbf{F}$ as original variables, $\{S : s_1, \ldots, s_{16}\}$ and 12 cross channels as the derived variables, $\{T : T_1, \ldots, T_{12}\}$.

$$\text{Original variables}: \quad S = \{g_{ij}, f_{ji}\}, \quad i \in \{1,2,3,4\}, j \in \{1,2\} \tag{B.3}$$

$$\text{Derived variables}: \quad T = \{H_{ij}\}, \quad i, j \in \{1,2,3,4\}, i \neq j \tag{B.4}$$

Now we have the Jacobian matrix defined element-wise by

$$J_{ij} = \{\frac{\partial T_i}{\partial s_j}\}_{1 \le i \le 12, 1 \le j \le 16} \tag{B.5}$$

$$= \begin{pmatrix}
f_{12} & f_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{11} & 0 & 0 & 0 & g_{12} & 0 & 0 \\
f_{13} & f_{23} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{11} & 0 & 0 & 0 & g_{12} & 0 \\
f_{14} & f_{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{11} & 0 & 0 & 0 & g_{12} \\
0 & 0 & f_{11} & f_{21} & 0 & 0 & 0 & 0 & g_{21} & 0 & 0 & 0 & g_{22} & 0 & 0 & 0 \\
0 & 0 & f_{13} & f_{23} & 0 & 0 & 0 & 0 & 0 & 0 & g_{21} & 0 & 0 & 0 & g_{22} & 0 \\
0 & 0 & f_{14} & f_{24} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & g_{21} & 0 & 0 & 0 & g_{22} \\
0 & 0 & 0 & 0 & f_{11} & f_{21} & 0 & 0 & g_{31} & 0 & 0 & 0 & g_{32} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & f_{12} & f_{22} & 0 & 0 & 0 & g_{31} & 0 & 0 & 0 & g_{32} & 0 & 0 \\
0 & 0 & 0 & 0 & f_{14} & f_{24} & 0 & 0 & 0 & 0 & 0 & g_{31} & 0 & 0 & 0 & g_{32} \\
0 & 0 & 0 & 0 & 0 & 0 & f_{11} & f_{21} & g_{41} & 0 & 0 & 0 & g_{42} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & f_{12} & f_{22} & 0 & g_{41} & 0 & 0 & 0 & g_{42} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & f_{13} & f_{23} & 0 & 0 & g_{41} & 0 & 0 & 0 & g_{42} & 0
\end{pmatrix} \tag{B.6}$$

We need to prove the rank of this matrix is almost always 12 and for that all we need to show is a single realization of g_{ij}'s and f_{ij}'s that gives a non zero value for the determinant of any 12×12 sub matrix. The idea here is that the determinant of the jacobian matrix is a multivariate polynomial in g_{ij}'s and f_{ij}'s and using the Schwartz-Zippel lemma we can argue that if this polynomial is has a non-zero realization that almost surely the polynomial by itself is non-zero. Using MATLAB, we can see that the determinant polynomial of the Jacobian matrix is non-zero for a random realization of g_{ij}'s and f_{ij}'s.

This proves the algebraic independence of the 12 cross channels of the 4-user interference channel with rank, D=2. $\square$

Similar to the proof of lemma 3.6, we can also show that the channels H_{11}, H_{13}, H_{41} and H_{43} are algebraically independent. This in turn shows that the polynomial $zH_{43} - H_{41}H_{13}$ is non-zero.

$K = 5$ **and** $D = 3$

Proof of lemma 3.7. The transfer matrix for this case has 20 cross channels and the number of generic variables here is 30. Here we consider 21 channels of the transfer matrix comprising of 20 cross channels and anyone of the direct channels, say H_{55}, without loss of generality. The proof for this is similar to the case in B.3, we just consider 1 direct channel along with 20 cross channels. The matrix from lemma 3.2 is of size 21×30 for this case. Similar to the 4-user case, using MATLAB we can see that the determinant polynomial of the Jacobian matrix is non-zero for a random realization of g_{ij}'s and f_{ij}'s. $\square$

Similar to the proof of lemma 3.7, we can also show that the channels involved in the denominator polynomial of $f_2(z, X, Z_{AI})$ in equation 3.19, are algebraically independent. This in turn shows that the denominator polynomial of $f_2(z, X, Z_{AI})$ is non-zero.