Title
On the Effects of Network Structure on the Achievable Rates for Multiple Unicasts

Author
Meng, Chun

Publication Date
2014

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,
IRVINE

On the Effects of Network Structure on the Achievable Rates for Multiple Unicasts

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

by

Chun Meng

Dissertation Committee:
Associate Professor Athina Markopoulou, Chair
Associate Professor Syed Ali Jafar
Professor Ender Ayanoglu

2014

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Prof. Athina Markopoulou. It will be impossible for me to complete this thesis without her support. Her insights and guidance have been very valuable for my Ph.D. studies. Her strong support during my graduate study at UC Irvine benefits me a lot for my academic career. I owe my greatest gratitude to her.

I would like to thank Prof. Syed Jafar and Prof. Ender Ayanoglu for their advice as members of my thesis committee. I am very grateful to Prof. Syed Jafar for his insightful advice and help during my work on our joint project on network alignment. I am also very thankful for the support of Prof. Ender Ayanoglu during my first year at UC Irvine.

I would like to thank Raymond Yeung for hosting my visit during my study at the Institute of Network Coding at the Chinese Univ. of Hong Kong. I am especially grateful to Minghua Chen for his insightful help and guidance during my work at INC, which inspired the fourth chapter of this thesis.

I would like to thank my collaborators on all the research projects I have been involved during my studies: Abinesh Ramakrishnan, Hulya Seferoglu, Abhik Das, Kenneth W. Shum, and Chung Chan. They have provided great help for my research.

This work would not have been possible without the support of my family and friends. I would like to thank my wife for her support during my study. Many thanks to my group mates, Abinesh, Anh, Minas, Hulya, Pegah, and Blerim, for their support.

Finally, I would like to thank the Department of Electrical Engineering and Computer Science for assisting me with fellowships during my studies. I would also like to acknowledge the support by NSF (awards 0747110 and 1028394) and AFOSR (FA9550-09-0643)

CURRICULUM VITAE

Chun Meng

EDUCATION

Doctor of Philosophy in Computer Engineering **2014**
University of California, Irvine *Irvine, California, U.S.A.*

Master of Science in Computer Engineering **2009**
Beijing University of Posts and Telecommunications *Beijing, China*

Bachelor of Science in Microelectronics **1998**
Peking University *Beijing, China*

ABSTRACT OF THE DISSERTATION

On the Effects of Network Structure on the Achievable Rates for Multiple Unicasts

By

Chun Meng

Doctor of Philosophy in Computer Engineering

University of California, Irvine, 2014

Associate Professor Athina Markopoulou, Chair

In this dissertation, we consider the problem of multiple unicast sessions over directed acyclic graphs. Although characterizing the rate region of inter-session network coding is a well-known open problem, we consider three particular network models. We design schemes, characterize the rates they achieve, and highlight the relation between achievable rates and network structure.

First, we consider networks, where the core is simple and all intelligence lies at the edge. Each intermediate node can only perform random linear network coding. We apply a precoding-based inference alignment technique at the edge, and refer to it as *precoding-based network alignment* (or PBNA). This approach combines the simplicity of RLNC in the core of the network with the guarantees of alignment (rate 1/2 per session). We observe that network structure may introduce dependencies, which we refer to as coupling relations, between elements of the transfer matrix, which might affect the achievable rate of PBNA. We identify the minimal set of such coupling relations, and we interpret them in terms of network structure properties. We also present polynomial-time algorithms to check the existence of these coupling relations on a given directed acyclic graph.

Second, we consider networks, where each node can perform linear network coding (not necessarily random). We propose a constructive method: (i) the unicast sessions are first

partitioned into multiple disjoint subsets of unicast sessions; (ii) each subset of unicast sessions is then mapped to a multicast session, for which a linear network coding scheme is constructed. Together, these serve as a linear network coding scheme for the original multiple unicast sessions, which we refer to as the *multicast-packing coding scheme (MPC)*. We show that the rate region of MPC is characterized by a set of linear constraints. We also propose a practical simulated annealing algorithm for approximating the optimal performance of MPC. Using simulations, we demonstrate the benefits of MPC as well as the efficiency of the simulated annealing algorithm.

Third, we consider networks, where each node can perform network coding (linear or non-linear) or simple routing. There exist networks, for which network coding doesn't provide any benefit over routing, which we refer to as *routing-optimal networks*. We identify a class of routing-optimal networks, which we refer to as *information-distributive networks*, defined by three structural features. We show that information-distributive networks don't subsume all routing-optimal networks. We present examples of information-distributive networks, including some examples from index coding and a single unicast session with hard deadline constraints.

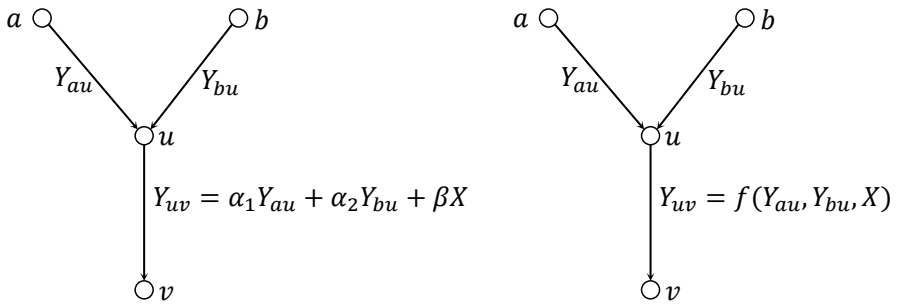
Chapter 1

Introduction

1.1 Overview of Network Coding

Network coding was first proposed as an alternative transmission method to routing scheme [1]. In a network coding scheme, in addition to simply replicating and forwarding received data, each node in the network can perform coding operations on the data it receives. Compared with routing, network coding provides more flexibility to utilize network resources, and thus more opportunities to enhance the performance of the network. It was shown that for a single multicast, network coding scheme can achieve better transmission rate than routing scheme [1]. It was also shown that network coding scheme usually incurs lower cost than routing scheme, when both transmission schemes achieve the same rate [2]. Some researchers showed that network coding can increase the power-efficiency of wireless networks [3, 4].

According to the types of coding operations performed by the nodes in a network, network coding schemes can be classified into two classes, *i.e., linear* network coding schemes and *non-linear* network coding schemes. In a linear network coding scheme [5, 6], the symbols transmitted along each edge are treated as elements from a finite field; at each node, the



(a) Linear network coding scheme

(b) Nonlinear network coding scheme

Figure 1.1: Illustration of linear and non-linear network coding schemes. In the above figures, X is a symbol generated by u, and Y_{au}, Y_{bu}, Y_{uv} denote the symbols transmitted along the edges $(a,u),(b,u),(u,v)$ respectively. In a linear network coding scheme, as shown in (a), $X, Y_{au}, Y_{bu}, Y_{uv}$ belong to a finite field $\mathbb{F}_q$, and Y_{uv} is a linear combination of Y_{au}, Y_{bu} and X, $Y_{uv} = \alpha_1 Y_{au} + \alpha_2 Y_{bu} + \beta X$, where $\alpha_1, \alpha_2, \beta \in \mathbb{F}_q$. In a nonlinear network coding scheme, as shown in (b), $X, Y_{au}, Y_{bu}, Y_{uv}$ belong to a finite alphabet Σ (not necessarily a finite field), and Y_{uv} is an arbitrary function (not necessarily linear combination) of Y_{au}, Y_{bu} and X, $Y_{uv} = f(Y_{au}, Y_{bu}, X)$.

symbols transmitted along an outgoing edge are linear combinations of the symbols received and generated (in case the node is a sender) by the node. In contrast, in a non-linear network coding scheme [1], the symbols transmitted along each edge are treated as elements from a finite alphabet (not necessarily a finite field), and at each node, the symbols transmitted along an outgoing edge can be any arbitrary functions (not necessarily linear combinations) of the symbols received and generated by the node. An illustration of these two network coding schemes is presented in Fig. 1.1.

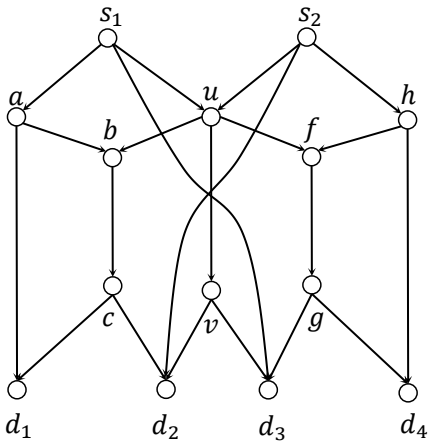
The fundamental network coding problem is to characterize the rate region achieved by network coding schemes (linear or non-linear) for various transmission scenarios, e.g., single multicast and multiple unicasts, and design network coding schemes that achieve the rate region. The research work on this problem can be roughly grouped into two categories, *i.e., intra-session* network coding schemes and *inter-session* network coding schemes. In the followings, we will introduce these two categories of network coding schemes, and briefly review related work.

1.1.1 Intra-Session Network Coding

In an intra-session network coding scheme, only symbols from the same sender can be coded together. We will present an example of intra-session network coding scheme.

Example 1.1.1. Consider the network as shown in Fig. 1.2, where each edge has capacity of one, and represents a delay-free and error-free channel. There exist two multicast sessions: $s_1 \rightarrow \{d_1, d_2\}$ and $s_2 \rightarrow \{d_3, d_4\}$, *i.e.,* d_1 and d_2 both require the symbols sent by s_1, and d_3 and d_4 both require the symbols sent by s_2. Assume s_i $(i = 1, 2)$ transmits three symbols $X_i^{(1)}, X_i^{(2)}, X_i^{(3)}$, all of which belong to the binary field $\mathbb{F}_2$. The transmission process continues for two time slots. For an edge (p, q), let $Y_{pq}^{(t)}$ denote the symbol transmitted along (p, q) during time slot t. The network coding scheme for the first time slot is: $Y_{s_1a}^{(1)} = Y_{ab}^{(1)} = Y_{ad_1}^{(1)} = X_1^{(1)}$, $Y_{s_1u}^{(1)} = Y_{ub}^{(1)} = Y_{uv}^{(1)} = Y_{vd_2}^{(1)} = X_1^{(2)}$, $Y_{bc}^{(1)} = Y_{cd_1}^{(1)} = Y_{cd_2}^{(1)} = X_1^{(1)} + X_1^{(2)}$, $Y_{s_2h}^{(1)} = Y_{hf}^{(1)} = Y_{fg}^{(1)} = Y_{hd_4}^{(1)} = Y_{gd_3}^{(1)} = Y_{gd_4}^{(1)} = X_2^{(1)}$, and the other edges transmit zeros. Apparently, d_3, d_4 both receive $X_2^{(1)}$. Since d_1 receives $X_1^{(1)}$ and $X_1^{(1)} + X_1^{(2)}$ along (a, d_1) and (c, d_1) respectively, it can decode both $X_1^{(1)}, X_1^{(2)}$. Similarly, d_2 can also decode $X_1^{(1)}, X_1^{(2)}$. The network coding scheme for the second time slot is: $Y_{s_1a}^{(2)} = Y_{ab}^{(2)} = Y_{ad_1}^{(2)} = Y_{bc}^{(2)} = Y_{cd_1}^{(2)} = Y_{cd_2}^{(2)} = X_1^{(3)}$, $Y_{s_2u}^{(2)} = Y_{uf}^{(2)} = Y_{uv}^{(2)} = Y_{vd_3}^{(2)} = X_2^{(2)}$, $Y_{s_2h}^{(2)} = Y_{hf}^{(2)} = Y_{hd_4}^{(2)} = X_2^{(3)}$, $Y_{fg}^{(2)} = Y_{gd_3}^{(2)} = Y_{gd_4}^{(2)} = X_2^{(2)} + X_2^{(3)}$, and the other edges transmit zeros. Thus, d_1, d_2 both receive $X_1^{(3)}$, and d_3, d_4 both decode $X_2^{(2)}, X_3^{(3)}$. Clearly, the above network coding scheme achieves a rate of $\frac{3}{2}$ for each multicast session. ■

Remark. In the above example, since the symbols from different senders cannot be coded together, (u, v) can only transmit the symbols sent by either s_1 or s_2, but not linear combinations of the symbols sent by them. Also note that both (s_1, d_3) and (s_2, d_2) are not used in this network coding scheme. This is because the receivers only receive linear combinations of the symbols sent by the same sender, and thus, the symbols transmitted along these two edges cannot help the receivers decode their required symbols.



(a)

Figure 1.2: Intra-session network coding vs. inter-session network coding. In the figure, each edge has capacity of one, and represents a delay-free and error-free channel. There are two multicast sessions: $s_1 \rightarrow \{d_1, d_2\}$ and $s_2 \rightarrow \{d_3, d_4\}$. The maximal symmetrical rate achieved by an intra-session network coding scheme is $\frac{3}{2}$, whereas an inter-session network coding scheme achieves a symmetrical rate of two, which is 30% better than the optimal intra-session network coding scheme.

The intra-session network coding problem is greatly simplified by the constraint that only symbols from the same sender can be coded together. As shown in [1, 6], the rate region of intra-session network coding scheme (linear or nonlinear) can be characterized by the minimum cut bounds. Using the duality between minimum cut and maximum flow, the rate region can be easily calculated by simple linear programming technique [7, 8]. Moreover, it is known that linear network coding scheme is sufficient to achieve the whole rate region of any (linear or nonlinear) intra-session network coding schemes [5]. Either a centralized polynomial-time algorithm [9] or distributed approaches [10, 11] can be used to find such a network coding scheme in an efficient manner.

1.1.2 Inter-Session Network Coding

In contrast to intra-session network coding schemes, in an inter-session network coding scheme, both symbols from the same sender and those from different senders can be coded

together. As we will show below, for some networks, intra-session network coding schemes are unable to unleash the full potential of the network, and inter-session network coding schemes can achieve better rates than intra-session network coding schemes.

Example 1.1.2. We reconsider the example as shown in Fig. 1.2. Using the linear programming formulations presented in [7], it can be verified that the maximum symmetrical rate achieved by intra-session network coding schemes for the two multicast sessions is $\frac{3}{2}$, *i.e.,* using intra-session network coding schemes, each sender can send at most three symbols to their corresponding receivers in two time slots. We will show that inter-session network coding can achieve a symmetrical rate of two for the two multicast sessions, *i.e.,* each sender can send four symbols to their corresponding receivers in two time slots, which is 33% better than the optimal intra-session network coding scheme. Let $X_i^{(1)}, X_i^{(2)}$ denote the two symbols sent by s_i $(i = 1, 2)$. The network coding scheme is as follows: $Y_{s_1a} = Y_{ab} = Y_{ad_1} = X_1^{(1)}$, $Y_{s_1u} = Y_{ub} = Y_{s_1d_3} = X_1^{(2)}$, $Y_{bc} = Y_{cd_1} = Y_{cd_2} = X_1^{(1)} + X_1^{(2)}$, $Y_{s_2h} = Y_{hf} = Y_{hd_4} = X_2^{(1)}$, $Y_{s_2u} = Y_{uf} = Y_{s_2d_2} = X_2^{(2)}$, $Y_{fg} = Y_{gd_3} = Y_{gd_4} = X_1^{(2)} + X_2^{(2)}$, $Y_{uv} = Y_{vd_2} = Y_{vd_3} = X_1^{(2)} + X_2^{(2)}$. Note that in this network coding scheme, (u, v) transmits $X_1^{(2)} + X_2^{(2)}$, which is a linear combination of the symbols sent by s_1 and s_2, whereas in the intra-session network coding scheme constructed in Example 1.1.1, (u, v) transmits uncombined symbols $X_1^{(2)}$ and $X_1^{(2)}$. Similar to Example 1.1.1, d_1 can decode both $X_1^{(1)}$ and $X_1^{(2)}$. Upon reception of $X_1^{(2)} + X_2^{(2)}$ and $X_2^{(2)}$, d_2 decodes $X_1^{(2)}$. Together with $X_1^{(1)} + X_1^{(2)}$ that is received along (c, d_2), it then decodes $X_1^{(1)}$. Likewise, both d_3 and d_4 can decode $X_2^{(1)}, X_2^{(2)}$. Clearly, the above network coding scheme achieves a symmetrical rate of two. ■

Remark. In the inter-session network coding scheme constructed in the above example, both (s_1, d_3) and (s_2, d_2) are useful, because they help d_2 and d_3 decode their required symbols from the combined symbol Y_{uv}. In contrast, as shown in Example 1.1.1, these two edges are useless in any intra-session network coding schemes, because the symbols transmitted along these two edges come from different senders, and cannot help the receivers to decode their

required symbols in any intra-session network coding scheme. This suggests that inter-session network coding schemes are more efficient in utilizing network resources than intra-session network coding schemes.

However, allowing symbols from different senders to be mixed together makes the problem of characterizing the rate region of inter-session network coding schemes significantly difficult. For example, it is known that there are networks with multiple unicasts for which linear network coding scheme[1] can achieve arbitrarily better rate than routing schemes [12]. However, the rate region of linear network coding schemes for the general multi-sender and multi-receiver transmission scenarios is still unknown except for simple networks. Moreover, there exist networks for which nonlinear network coding schemes achieve better rate than linear network coding schemes [13, 14]. Thus, in general, nonlinear network coding should be considered in order to fully utilize network resources. The consideration of nonlinear network coding makes the network coding problem even more challenging, because the coding operations in a nonlinear network coding scheme are literally unlimited. Most of recent approaches to characterizing the rate region of nonlinear network coding schemes employ tools from information theory. For example, in [15], the authors proposed to use Shannon-type information inequalities to approximate the rate region of nonlinear network coding schemes. However, it was later pointed out that there are networks for which non-Shannon-type information inequalities provide tighter bounds than Shannon-type information inequalities [16]. Some researchers proposed to use entropy functions to calculate the rate region of nonlinear network coding schemes [17, 18]. Unfortunately, this approach is difficult to use because the entropy functions are vectors of an exponential number of dimensions, which explode very quickly with network size.

Finding an inter-session network coding scheme that achieves the optimal rate is also difficult. It was shown that finding the optimal linear network coding scheme is NP-hard [19]. More-

[1]In the rest of this thesis, unless specified otherwise, all network coding schemes refer to inter-session network coding schemes.

over, it was later found that even finding a linear / nonlinear network coding scheme that achieves a rate close to the optimal rate is NP-hard [20]. Therefore, most researchers resort to heuristic or constructive approaches that achieves only sub-optimal rates. For example, in [21, 22], the authors proposed to use linear optimization methods to find linear network coding schemes over butterfly structures in the network. Based on the linear programming formulations proposed in [21], a routing-scheduling-coding strategy was proposed to jointly solve the network coding design problem and queue scheduling problem [23]. In [24], the authors presented a tiling approach to finding linear network coding schemes by applying dynamic programming technique. Some researchers applied game theory to find network coding schemes for multiple unicasts [25, 26]. An evolutionary approach was proposed to find linear network coding schemes for multiple unicasts [27]. Some researchers proposed simple XOR-based network coding schemes for wireless networks by utilizing the broadcast nature of wireless channel [28, 29]. Most of these approaches focus on finding sub-optimal linear network coding schemes, and the rate region of which is usually calculated via simulations.

1.2 Motivation

Network structure plays a central role in determining the rate region of network coding schemes, and directly affects the ways the symbols transmitted in the network are combined together. In the followings, we will use examples to illustrate how network structure affects the rate region of network coding schemes.

One important effect of network structures is that they may introduce dependency relationships among the symbols transmitted in the network, and the symbols transmitted through the network may be correlated. These correlations are important in determining the achievable rate of a random linear network coding scheme, as we will see in Chapter 2.

Example 1.2.1. Consider the network shown in Fig. 1.3a, where there are three unicast

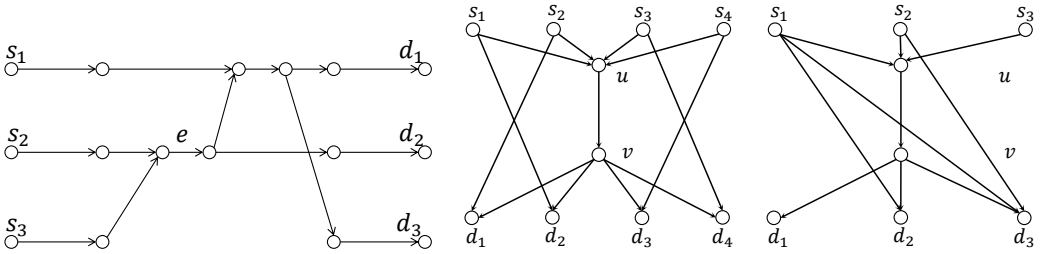


Figure 1.3: Effects of network structure on the rate region of network coding scheme. In each of the above networks, the capacities of all the edges equal one, and there are multiple unicast sessions in the network, with s_i $(1 \leq i \leq 4)$ and d_i being the sender and the receiver of the ith unicast session respectively. In (a), the network structure introduces dependency among the received symbols at d_1 and d_2, which will affect the achievable rate of a random network coding scheme, as shown in Chapter 2. In (b), due to the network structure, combing X_1 and X_3 brings no benefit since it takes two more time slots for d_1, d_3 to decode the combined symbols, but combining X_1 with X_2 or X_3 with X_4 is beneficial because the involved receivers can immediately decode their required symbols. In (c), the network structure make it unnecessary to combine symbols throughout the network, and routing can achieve the same rate vector as any network coding schemes.

sessions (s_i, d_i) $(i = 1, 2, 3)$, and each edge has capacity of one. Suppose each sender s_i transmits a symbol X_i to its corresponding receiver d_i. Under linear network coding scheme, the received symbol at d_i is a linear combination of X_1, X_2, X_3: $Z_i = m_{1i}(\mathbf{x})X_1 + m_{2i}(\mathbf{x})X_2 + m_{3i}(\mathbf{x})X_3$, where $m_{ji}(\mathbf{x})$ $(j = 1, 2, 3)$ denotes the linear coefficient of X_j. Each $m_{ji}(\mathbf{x})$ is called a *transfer function*, and is a polynomial in terms of the coding coefficients in the network[2]. Note that if we remove edge e from the network, d_1 and d_2 will be both disconnected from s_2 and s_3. As we will show in Section 2.5, this implies that $m_{22}(\mathbf{x})m_{31}(\mathbf{x}) = m_{32}(\mathbf{x})m_{21}(\mathbf{x})$, *i.e.,* the symbols received at d_1 and d_2 are correlated. ■

At a high level, there are two contradicting trends in an inter-session network coding scheme. One trend is to mix symbols from different senders such that useful information can be transmitted through bottlenecks in the network. For instance, in the network coding scheme in Example 1.1.2, (u, v) is a bottleneck between the two multicast sessions, and it transmits a linear combination $X_1^{(2)} + X_2^{(2)}$, which carries useful information for both multicast sessions.

[2] $\mathbf{x}$ denotes the vector of all the coding coefficients in the network.

The other trend is to decode the mixed symbols by using network capacities other than those occupied by the mixed symbols. For example, in Example 1.1.2, (s_1, d_3) and (s_2, d_2) transmit symbols that are used to decode the mixed symbol along (u, v). While the first trend tends to reduce consumption of network resources, the second trend usually increases consumption of network resources. As we will illustrate below, network structure is a determining factor on how to make the trade-off between these two trends. Specifically, it determines whether the symbols from one sender can or cannot be combined with the symbols from another sender such that the cost incurred by decoding symbols is paid off by the benefits gained from combining symbols.

Example 1.2.2. Consider the network shown in Fig. 1.3b, in which each edge has capacity of one. There are four unicast sessions in the network, with the ith ($1 \leq i \leq 4$) unicast session being $\omega_i = (s_i, d_i)$, where s_i and d_i are the sender and the receiver of ω_i respectively. Suppose each sender s_i transmits a symbol X_i to its corresponding receiver d_i. It can be easily seen that both X_1 and X_2 can be combined together along (u, v), because both d_1 and d_2 can decode their required symbol from this combined symbol by using the symbol transmitted along (s_2, d_1) and (s_1, d_2) respectively. For instance, let $Y_{uv} = X_1 + X_2$, $Y_{s_2d_1} = X_2$, and $Y_{s_1d_2} = X_1$. Clearly, d_1 and d_2 can decode X_1 and X_2 respectively from their received symbols. Similarly, X_3 can be combined with X_4 along (u, v). For either case, the edge capacities used for decoding purpose are disjoint with the edge capacities used for transmitting combined symbols, and hence the combined symbol and the symbol used to decode the combined symbol can be transmitted simultaneously. However, it does no good to combine X_1 with X_3 along (u, v). For example, suppose (u, v) transmits a linear combination $Y_{uv} = X_1 + X_3$. In order to decode X_1, d_1 must receive X_3. Since (u, v) is the only edge that connects s_3 to d_1, the only way for d_1 to receive X_3 is to let s_3 transmit X_3 along (u, v). Similarly, in order to decode X_3, s_1 must send X_1 to d_3 along (u, v). Since all the three transmissions need to occupy the edge capacity of (u, v), it takes three time slots for the two receivers d_1, d_3 to decode their required symbols. This is even worse than purely

routing the two symbols X_1, X_3 along (u, v), which takes only two time slots. Hence, the benefit of combining X_1 and X_3 is surpassed by the cost incurred by decoding the combined symbol. ■

Furthermore, network structure determines whether it is necessary to mix symbols throughout the network. As we've seen in the above example, combining symbols is usually accompanied with sending symbols for the purpose of decoding the combined symbols. In some networks, the network capacities consumed by these two types of symbols can be used to simply routing uncoded symbols such that the two corresponding transmission schemes achieve the same transmission rate, *i.e.,* mixing symbols is not necessary for such networks. In the followings, we present such an example.

Example 1.2.3. Consider the network shown in Fig. 1.3c, in which each edge has capacity of one, and there are three unicast sessions (s_i, d_i) $(i = 1, 2, 3)$. We'll show that mixing symbols is not necessary for this network. Assume each sender s_i sends a symbol X_i to d_i. We consider two network coding schemes. In the first network coding scheme, (u, v) transmits a linear combination $X_1 + X_2$. It can be easily seen that d_2 can immediately decode X_2 from the combined symbol if (s_1, d_2) transmits X_1. However, in order for d_1 to decode the combined symbol, the only choice is to let s_2 transmit X_2 along (u, v). Since both $X_1 + X_2$ and X_2 need to occupy the edge capacity of (u, v), it takes two time slot for d_1, d_2 to decode their required symbols. Apparently, a more straightforward way is to route X_1 and X_2 along the paths $P_1 = \{(s_1, u), (u, v), (v, d_1)\}$ and $P_2 = \{(s_2, u), (u, v), (v, d_2)\}$ respectively. Hence, it is not necessary to mix X_1 and X_2. In the second network coding scheme, (u, v) transmits a linear combination $X_1 + X_2 + X_3$. Similar to above, d_3 can immediately decode X_3 from the combined symbol, if (s_1, d_3), and (s_2, d_3) transmit X_1 and X_2 respectively. However, it takes at least two additional time slots for d_2 and d_1 to decode their required symbols from the combined the symbol. For instance, in one time slot, (u, v) transmits $X_1 + X_3$, and thus d_2 can decode X_2 by adding $X_1 + X_2 + X_3$ and $X_1 + X_3$ together; in the next time slot, (u, v)

transmits X_2+X_3, and thus d_1 can decode X_1 by adding $X_1+X_2+X_3$ and X_2+X_3 together. However, a straightforward way is to simply route X_1, X_2, X_3 along the paths P_1, P_2, and $P_3 = \{(s_3, u), (u, v), (v, d_3)\}$ respectively. Therefore, it is not necessary to mix X_1, X_2 and X_3. In fact, as we will show in Chapter 4, for this network, given a network coding scheme (linear or nonlinear), there is always a routing scheme that achieves the same rate vector as the network coding scheme, implying that it is not necessary to combine symbols in the network. ■

1.3 Contributions

In this thesis, we consider inter-session network coding for multiple unicasts on directed acyclic graphs. In particular, we attempt to understand the effects of network structure on the region of network coding schemes. We consider three network models: (i) dummy networks in which except for the senders and the receivers, all the intermediate nodes can only perform random linear network coding, *i.e.,* there is no intelligence in the middle of the network; (ii) linear networks where each node can perform linear network coding, *i.e.,* coding at each node are limited to linear operations; (iii) nonlinear networks in which each node can perform nonlinear network coding, *i.e.,* the coding operations at each node are unlimited. Clearly, the three network models describe three different levels of complexity in terms of coding operations that each node in the network is able to perform. We will study how network structure affects the rate region of network coding for the three network models. We make the following contributions.

1.3.1 Precoding-Based Network Alignment for Three Unicast Sessions

We consider the network coding problem across three unicast sessions under the dummy network model, where except for the senders and the receivers, all the intermediate nodes in the network can only perform random linear network coding operations. We assume the sender and the receiver of each unicast session are both connected to the network via a single edge of unit capacity. We refer to this communication scenario as a *Single-Input Singe-Output scenario* or SISO scenario for short. In particular, we consider precoding-based linear schemes, in which each sender uses a precoding-matrix to encode symbols, and all the intermediate nodes perform random linear network coding.

Our basic idea is that under the dummy network model, the network behaves like a wireless interference channel, and hence we can borrow some of the techniques, such as precoding-based interference alignment [30], which are originally developed for wireless interference channel, to the network setting. We refer to this approach as precoding-based network alignment approach, or PBNA for short. One advantage of PBNA is that it significantly simplifies network code design since the nodes in the middle of the network perform random network coding. Another advantage is that PBNA can achieve the optimal symmetrical rate achieved by any precoding-based linear schemes.

However, unlike wireless interference channel, network structure may introduce dependency between transfer functions, which we refer to coupling relations (see Example 1.2.1). These coupling relations may affect the rate achieved by PBNA. Therefore, traditional interference alignment techniques, which are developed for the wireless interference channel, cannot be directly applied to DAGs with network coding but (i) they need to be properly adapted in the new setting and (ii) their achievability conditions need to be characterized in terms of the network topology.

In this part of the thesis, we make the following contributions:

- *Graph-related properties*: We find that due to network topology, the transfer functions usually possess special properties, called graph-related properties, which are absent in general polynomials. These graph-related properties play important roles in identifying all possible coupling relations that may affect the achievable rate of PBNA, and characterizing the topological features of these coupling relations.

- *Achievability Conditions*: Using two graph-related properties, *i.e., Linearization Property* and *Square-Term Property*, we identify the minimal set of coupling relations between network transfer functions, the presence of which will potentially affect the achievable rate of PBNA.

- *Interpretation of Coupling Relations*: We present interpretations of these coupling relations in terms of network topology. Based on these interpretations, we present polynomial-time algorithm to check the existence of these coupling relations.

1.3.2 Multicast-Packing Coding Scheme for Multiple Unicast Sessions

We consider the network coding problem for multiple unicast sessions under the linear network model, where all the nodes in the network can perform any linear network coding operations, including random linear network coding operations. As it is NP-hard to find the optimal linear network coding scheme for the network setting, we consider a constructive approach to constructing liner network coding schemes. The ideas of this approach consists of the following points: (i) we can partition the multiple unicast sessions into several disjoint groups; (ii) we map each group of unicast sessions into multiple multicast sessions, such that each receiver in the group can decode all the symbols sent by the senders in the group; (iii) we

construct linear network coding schemes for these multicast sessions by using a deterministic approach [9] or a random approach [10]. These linear network coding schemes collectively serve as a linear network coding scheme for the original multiple unicast sessions. We refer to such a linear network coding scheme as a multicast-packing coding scheme or MPC for short.

Our proposed scheme, MPC, has the following strengths:

- The MPC approach, *i.e.,* partitioning the unicast sessions to subsets of unicast sessions and mapping each subset to a multicast network coding problem, is general enough to be applied to any directed acyclic graph.
- Given a partition of the set of the unicast flows, the rate region of MPC can be characterized by using the minimum cut bounds. Using the duality between minimum cut and max flow, the rate region can be quickly calculated by using linear programming technique. In contrast, previous constructive approaches are difficult to analyze due to the lack of succinct mathematical formulations.
- In order to find the best MPC, we only need to search the space of all partitions of the set of unicast flows, *independently of the network size.* This is clearly more efficient and scalable than other constructive approaches, whose combinatorial optimization involved the network graph in addition to the set of flows.

Although the search space is independent of the network size, its size is still exponential in the number of unicast sessions. Thus, we utilize a suboptimal, yet efficient, simulated annealing technique to find good partitions of the unicast flows.

We use simulations over appropriately chosen scenarios to evaluate the performance of MPC. Simulation results show that MPC achieves up to 30% performance gain over routing, and

the convergence speed of the simulated annealing algorithm is 5 times faster than the evolutionary approach [27], which is, to our best knowledge, the fastest algorithm in the literature.

1.3.3 Routing-Optimal Networks for Multiple Unicast Sessions

We consider nonlinear networks, where each node in the network can perform nonlinear coding operations, *i.e.,* the coding operations allowed at each node are unlimited. In general, for nonlinear networks, linear network coding is insufficient to utilize network capacities to its full potential, and nonlinear network coding schemes can achieve better rates than linear network coding schemes. Yet, there exist networks for which simple routing schemes can achieve the whole rate region achieved by *any* linear and nonlinear network coding schemes, *i.e.,* it is not necessary to combine symbols in these networks. We refer to these networks as *routing-optimal* networks. We attempt to answer the following questions: (i) what are the distinct topological features of routing-optimal networks? (ii) why do these features make the network routing-optimal? The answers to these problems will not only explain which kind of networks can or cannot benefit from network coding, but will also better our understanding on how network topologies affect the rate region of network coding.

In this part of the thesis, we make the following contributions:

- We identify a class of networks, called *information-distributive* networks, which are defined by three topological features. The first two features capture how the edges in the cut-sets are connected to the sources and the sinks, and the third feature captures how the paths in the path-sets overlap with each other. Due to these features, given a network code, there is always a routing scheme such that it achieves the same rate vector as the network code, and the traffic transmitted through the network is exactly the source information distributed over the cut-sets between the sources and the sinks.

- We prove that if a network is information-distributive, it is routing-optimal. We also show that the converse is not true. This indicates that the three features might be too restrictive in describing routing-optimal networks.

- We present examples of information-distributive networks taken from the index coding problem [31] and single unicast with hard deadline constraint.

1.4 Thesis Outline

The structure of the rest of the thesis is as follows. In Chapter 2, we present a detailed discussion about precoding-based network alignment scheme for three unicast sessions. In Chapter 3, we present the multicast-packing coding scheme (MPC). In Chapter 4, we discuss about routing-optimal networks, in particular information-distributive networks. Finally, in Chapter 5, we conclude and summarize.

Chapter 2

Precoding-Based Network Alignment for Three Unicast Sessions

2.1 Introduction

In this chapter, we consider the problem of linear network coding across *three* unicast sessions over a network represented by a directed acyclic graph (DAG), where the sender and the receiver of each unicast session are both connected to the network via a single edge of unit capacity. We refer to this communication scenario as a *Single-Input Single-Output* scenario or *SISO* scenario for short. This is the smallest, yet highly non-trivial, instance of the problem. Furthermore, 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[1]. Apart from being of interest on its own right, we hope that this can be used as a building block and for better understanding of the general network coding problem across multiple unicasts.

[1]The precise definition of precoding-based linear scheme is presented in Section 2.3.

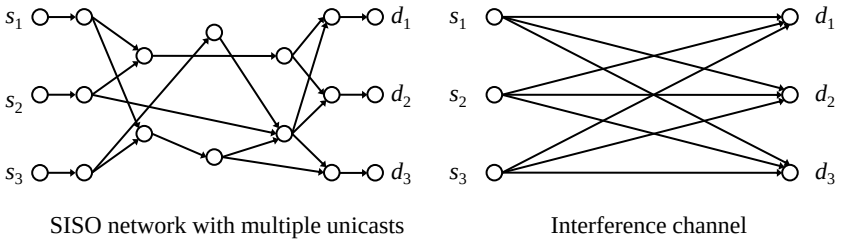


Figure 2.1: Analogy between a SISO scenario employing linear network coding and a wireless interference channel, each with three unicast sessions (s_i, d_i), $i = 1, 2, 3$. Both these systems can be treated as linear transform systems and are amenable to interference alignment techniques.

We consider the following approach to finding a network coding scheme, originally proposed by Das et al. [32]. The idea is that under the linear network coding framework, a SISO scenario behaves roughly like a wireless interference channel. As shown in Fig. 2.1, the entire network can be viewed as a channel with a linear transfer function, albeit this function is no longer given by nature, as it is the case in wireless, but is determined by the network topology, routing and coding coefficients. This analogy enables us to apply the technique of precoding-based interference alignment, designed by Cadambe and Jafar [30] for wireless interference channels. We adapt this technique to our problem and refer to it as *precoding-based network alignment*, or *PBNA* for short: precoding occurs only at source nodes, and all the intermediate nodes in the network perform random network coding. One advantage of PBNA is complexity: it significantly simplifies network code design since the nodes in the middle of the network perform random network coding. Another advantage is that PBNA can achieve the optimal symmetrical rate achieved by any precoding-based linear schemes, as shown in [33].

An important difference between the SISO scenario and the wireless interference channel is that there may be algebraic dependencies, which we refer to as *coupling relations*, between elements of the transfer matrix, which we refer to as *transfer functions*. These are introduced by the network topology and may affect the achievable rate of PBNA [34]. Such algebraic

dependencies are not present in the wireless interference channel, where channel gains are independent from each other such that the precoding-based interference alignment scheme of [30] can achieve 1/2 rate per session almost surely. Therefore, traditional interference alignment techniques, developed for the wireless interference channel, cannot be directly applied to networks with network coding but (i) they need to be properly adapted in the new setting, and (ii) their achievability conditions need to be characterized in terms of the network topology.

We make the following contributions:

- *Graph-Related Properties:* We identify some important graph-related properties of transfer functions, including the Linearization Property and the Square-Term Property, which play important roles in identifying the minimal set of coupling relations that may potentially affect the achievable rate of PBNA.
- *Achievability Conditions:* We identify the minimal set of coupling relations between transfer functions, the presence of which will potentially affect the achievable rate of PBNA.
- *Interpretation of Coupling Relations:* We further interpret these coupling relations in terms of network topology.
- *Algorithm to Check Coupling Relations:* We present polynomial-time algorithms for checking the existence of these coupling relations.

The rest of this chapter is organized as follows. In Section 2.2, we review related work. In Section 2.3, we present the problem setup and formulation. In Section 2.4, we present our proposed precoding-based interference alignment (PBNA) scheme for the network setting. In Section 2.5, we present an overview of our main results. In Section 2.6, we discuss in depth the achievability conditions of PBNA. In Section **??**, we provide polynomial-time algorithms

to check the presence of the coupling relations that may affect the achievable rate of PBNA. Section 2.7 summarizes this chapter. In Appendices A-D, we present detailed proofs for the lemmas and the theorems presented in this paper.

2.2 Related Work

2.2.1 Network Coding for Two Unicast Sessions

Network coding across two unicasts is one case that has been best understood up to now. Wang and Shroff provided a graph-theoretical characterization of sufficient and necessary condition for the achievability of symmetrical rate of one for two multicast sessions, of which two unicasts is a special case, over networks with integer edge capacities [35–37]. They showed that linear network code is sufficient to achieve this symmetrical rate. Wang et al. [38] further pointed out that there are only two possible capacity regions for the network studied in [37]. They also showed that for layered linear deterministic networks, there are exactly five possible capacity regions. Kamath et al. [39] provided a edge-cut outer bound for the capacity region of two unicasts over networks with arbitrary edge capacities.

2.2.2 Network Coding for More Than Two Unicast Sessions

For network coding across more than two unicasts, there is only limited progress. It is known that there exist networks in which network coding significantly outperforms routing schemes in terms of transmission rate [12]. However, there exist only approximation methods to characterize the rate region for this setting [15]. Moreover, it is known that finding linear network codes for this setting is NP-hard [19]. Therefore, only sub-optimal and heuristic methods exist to construct linear network code for this setting. For example, Ratnakar

et al. [22] considered coding pairs of flows using poison-antidote butterfly structures and packing a network using these butterflies to improve throughput; Traskov et al. [21] further presented a linear programming-based method to find butterfly substructures in the network; Ho et al. [40] developed online and offline back pressure algorithms for finding approximately throughput-optimal network codes within the class of network codes restricted to XOR coding between pairs of flows; Effros et al. [24] described a tiling approach for designing network codes for wireless networks with multiple unicast sessions on a triangular lattice; Kim et al. [27] presented an evolutionary approach to construct linear code. Unfortunately, most of these approaches don't provide any guarantee in terms of performance. Moreover, most of these approaches are concerned about finding network codes by jointly considering code assignment and network topology at the same time. In contrast, PBNA is oblivious to network topology in the sense that the design of encoding/decoding schemes is separated from network topology, and is predetermined regardless of network topology. The separation of code design from network topology greatly simplifies the code design of PBNA.

The part of our work that identifies coupling relations is related to some recent work on network coding. Ebrahimi and Fragouli [41] found that the structure of a network polynomial, which is the product of the determinants of all transfer matrices, can be described in terms of certain subgraph structures; Zeng et al. [42] proposed the Edge-Reduction Lemma which makes connections between cut sets and the row and column spans of the transfer matrices.

2.2.3 Interference Alignment

The original concept of precoding-based interference alignment was first proposed by Cadambe and Jafar [30] to achieve the optimal degree of freedom (DoF) for K-user wireless interference channel. After that, various approaches to interference alignment have been proposed. For example, Nazer et al. proposed ergodic interference alignment [43]; Bresler, Parekh and Tse

proposed lattice alignment [44]; Jafar introduced blind alignment [45] for the scenarios where the actual channel coefficient values are entirely unknown to the transmitters; Maddah-Ali and Tse proposed retrospective interference alignment [46] which exploits only delayed CSIT. Interference alignment has been applied to a wide variety of scenarios, including K-user wireless interference channel [30], compound broadcast channel [47], cellular networks [48], relay networks [49], and wireless networks supported by a wired backbone [50]. Recently, it was shown that interference alignment can be used to achieve exact repair in distributed storage systems [51] [52].

2.2.4 Network Alignment

The idea of PBNA was first proposed by Das et al., who also proposed a sufficient condition for PBNA to asymptotically achieve a symmetrical rate of 1/2 per session [32]. However, the sufficient achievability condition proposed in [32] contains an exponential number of constraints, and is very difficult to verify in practice. Later, Ramakrishnan et al. observed that whether PBNA can achieve a symmetrical rate of 1/2 per session depends on network topology [34], and conjectured that the condition proposed in [32] can be reduced to just six constraints. Han et al. [53] proved that this conjecture is true for the special case of three symbol extensions. They also identified some important properties of transfer functions, which are used in this paper. In [54], Meng et al. showed that the conjecture in [34] is false for more than three symbol extensions, and reduced the condition proposed in [32] to just 12 constraints by using two graph-related properties of transfer functions. Later, Meng et al. reduced the 12 constraints to a set of 9 constraints [33] by using a result from [53], and proved that they are also necessary conditions for PBNA to achieve 1/2 rate per session. They also provided an interpretation of all the constraints in terms of graph structure. At the same time and independently, a technical report by Han et al. [55] also provided a similar characterization.

2.3 Problem Formulation

2.3.1 Network Model

A network is represented by a directed acyclic graph $\mathcal{G} = (V, E)$, where V is the set of nodes and E the set of edges. We consider the simplest non-trivial communication scenario where there are three unicast sessions in the network. The ith ($i = 1, 2, 3$) unicast session is represented by a tuple $\omega_i = (s_i, d_i, \mathbf{X}_i)$, where s_i and d_i are the sender and the receiver of the ith unicast session, respectively; $\mathbf{X}_i = (X_i^{(1)}, X_i^{(2)}, \cdots, X_i^{(k_i)})^T$ is a vector of independent random variables, each of which represents a packet that s_i sends to d_i. Each sender s_i is connected to the network via a single edge σ_i, called a sender edge, and each receiver node d_i via a single edge τ_i, called a receiver edge. Each edge has unit capacity, *i.e.,* can carry one symbol of $\mathbb{F}_{2^m}$ in a time slot, and represents an error-free and delay-free channel. We group these unicast sessions into a set $\Omega = \{\omega_1, \omega_2, \omega_3\}$. We refer to the tuple $(\mathcal{G}, \Omega)$ as a single-input and single-output communication scenario, or a SISO scenario for short. An example of SISO scenario is shown in Fig. 2.1a. Clearly, in a SISO scenario, each sender can transmit at most one symbol to its corresponding receiver node in a time slot.

Given an edge $e = (u, v) \in E$, let $u = \text{head}(e)$ and $v = \text{tail}(e)$ denote the head and the tail of e, respectively. Given a node $v \in V$, let $\text{In}(v) = \{e \in E : \text{head}(e) = v\}$ denote the set of incoming edges at v, and $\text{Out}(v) = \{e \in E : \text{tail}(e) = v\}$ the set of outgoing edges at v. Given two distinct edges $e, e' \in E$, a directed path from e to e' is a subset of edges $P = \{e_1, e_2, \cdots, e_k\}$ such that $e_1 = e$, $e_k = e'$, and $\text{head}(e_i) = \text{tail}(e_{i+1})$ for $i \in \{1, 2, \cdots, k-1\}$. The set of directed paths from e to e' is denoted by $\mathcal{P}_{ee'}$. For $i, j \in \{1, 2, 3\}$, we also use $\mathcal{P}_{ij}$ to represent $\mathcal{P}_{\sigma_i \tau_j}$.

Each node in the network performs scalar linear network coding operations on the incoming symbols [5] [6]. The symbols transmitted in the network are elements of a finite field $\mathbb{F}_{2^m}$.

Let $\hat{X}_i$ be the symbol injected at the sender node s_i. Thus, for an edge $e = (u, v) \in E$, the symbol transmitted along e, denoted by Y_e, is a linear combination of the incoming symbols at u:

$$Y_e = \begin{cases} \hat{X}_i & \text{if } e = \sigma_i; \\ \sum_{e' \in \text{In}(u)} x_{e'e} Y_{e'} & \text{otherwise.} \end{cases} \tag{2.1}$$

where $x_{e'e}$ denotes the coding coefficient that is used to combine the incoming symbol $Y_{e'}$ into Y_e. Following the algebraic framework of [6], we treat the coding coefficients as variables. Let $\mathbf{x}$ denote the vector consisting of all the coding coefficients in the network, *i.e.,* $\mathbf{x} = (x_{e'e} : e', e \in E, \text{head}(e') = \text{tail}(e))$.

Due to the linear operations at each node, the network functions like a linear system such that the received symbol at τ_i is a linear combination of the symbols injected at sender nodes:

$$Y_{\tau_i} = m_{1i}(\mathbf{x})\hat{X}_1 + m_{2i}(\mathbf{x})\hat{X}_2 + m_{3i}(\mathbf{x})\hat{X}_3 \tag{2.2}$$

In the above formula, $m_{ji}(\mathbf{x})$ $(j = 1, 2, 3)$ is a multivariate polynomial in the ring $\mathbb{F}_2[\mathbf{x}]$, and is defined as follows [6]:

$$m_{ji}(\mathbf{x}) = \sum_{P \in \mathcal{P}_{ji}} t_P(\mathbf{x}) \tag{2.3}$$

Each $t_P(\mathbf{x})$ denotes a monomial in $m_{ji}(\mathbf{x})$, and is the product of all the coding coefficients along path P, *i.e.,* for a given path $P = \{e_1, e_2, \cdots, e_k\}$,

$$t_P(\mathbf{x}) = \prod_{i=1}^{k-1} x_{e_i e_{i+1}} \tag{2.4}$$

Thus, $t_P(\mathbf{x})$ represents the signal gain along a path P, and $m_{ji}(\mathbf{x})$ is simply the summation of the signal gains along all possible paths from σ_j to τ_i. We refer to $m_{ji}(\mathbf{x})$ as the *transfer*

function from σ_j to τ_i.

We make the following assumptions:

1. The nodes in $V - \{s_i, d_i : 1 \leq i \leq 3\}$ can only perform random linear network coding, i.e., there is no intelligence in the middle of the network. The variables in $\mathbf{x}$ all take values independently and uniformly at random from $\mathbb{F}_{2^m}$.

2. Except for the senders and the receivers, all other nodes in the network have zero memory, and therefore cannot store any received data.

3. The senders have no incoming edges, and the receivers have no outgoing edges.

4. The random variables in all $\mathbf{X}_i$'s are mutually independent. Each element of $\mathbf{X}_i$ has an entropy of m bits.

5. The transmissions within the network are all synchronized with respect to the symbol timing.

2.3.2 Transmission Process

The transmission process in the network continues for $N \in \mathbb{Z}_{>0}$ time slots, where $N \geq \max\{k_1, k_2, k_3\}$. Both N and k_i are parameters of the transmission scheme. We will show how to set these parameters in Section 2.4. Let $\mathbf{x}^{(t)} = (x_{e'e}^{(t)} : e', e \in E, \text{head}(e') = \text{tail}(e))$ denote the vector of coding coefficients for time slot t, where $x_{e'e}^{(t)}$ represents the coding coefficient used to combine the incoming symbol along e' into the symbol along e for time slot t. For an edge e, let $Y_e^{(t)}$ denote the symbol transmitted along e during time slot t, and $\mathbf{Y}_e = (Y_e^{(1)}, Y_e^{(2)}, \cdots, Y_e^{(N)})^T$ the vector of all the symbols transmitted along e during the N time slots. Define a vector of variables, $\xi = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}, \theta_1, \theta_2, \cdots, \theta_k)$, where

$\theta_1, \cdots, \theta_k$ are variables, which take values from $\mathbb{F}_{2^m}$, and are used in the encoding process at the senders.

Each sender s_i first encode $\mathbf{X}_i$ into a vector $\hat{\mathbf{X}}_i$ of N symbols:

$$\hat{\mathbf{X}}_i = \mathbf{V}_i \mathbf{X}_i \tag{2.5}$$

where $\mathbf{V}_i$ is an $N \times k_i$ matrix, each element of which is a rational function in $\mathbb{F}_{2^m}(\xi)$[2], and is called the *precoding matrix* at s_i. Define the following $N \times N$ diagonal matrix which includes all the transfer functions $m_{ji}(\mathbf{x}^{(t)})$ for the N time slots:

$$\mathbf{M}_{ji} = \begin{pmatrix} m_{ji}(\mathbf{x}^{(1)}) & 0 & \cdots & 0 \\ 0 & m_{ji}(\mathbf{x}^{(2)}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & m_{ji}(\mathbf{x}^{(N)}) \end{pmatrix} \tag{2.6}$$

Hence, the input-output equation of the network can be formulated in a matrix form as follows:

$$\begin{aligned} \mathbf{Y}_{\tau_i} &= \mathbf{M}_{1i}\hat{\mathbf{X}}_1 + \mathbf{M}_{2i}\hat{\mathbf{X}}_2 + \mathbf{M}_{3i}\hat{\mathbf{X}}_3 \\ &= \mathbf{M}_{1i}\mathbf{V}_1\mathbf{X}_1 + \mathbf{M}_{2i}\mathbf{V}_2\mathbf{X}_2 + \mathbf{M}_{3i}\mathbf{V}_3\mathbf{X}_3 \\ &= \mathbf{M}_i\mathbf{X} \end{aligned} \tag{2.7}$$

where $\mathbf{M}_i = (\mathbf{M}_{1i}\mathbf{V}_1 \quad \mathbf{M}_{2i}\mathbf{V}_2 \quad \mathbf{M}_{3i}\mathbf{V}_3)$, and $\mathbf{X} = (\mathbf{X}_1^T \quad \mathbf{X}_2^T \quad \mathbf{X}_3^T)^T$. Since the elements of $\mathbf{M}_{ji}$ $(1 \le j \le 3)$ and $\mathbf{V}_j$ are rational functions in $\mathbb{F}_{2^m}(\xi)$, the elements of $\mathbf{M}_i$ are also rational functions in terms of ξ.

[2]Given a field $\mathbb{F}$, $\mathbb{F}(x_1, \cdots, x_k)$ denotes the field consisting of all multivariate rational functions in terms of $(x_1, \cdots, x_k)$ over $\mathbb{F}$.

2.3.3 Precoding-Based Linear Scheme

In this paper, we consider the following transmission scheme, called precoding-based linear scheme:

Definition 2.3.1. Given a SISO scenario $(\mathcal{G}, \Omega)$, a *precoding-based linear scheme* for $(\mathcal{G}, \Omega)$ is a transmission scheme, where each sender s_i $(1 \leq i \leq 3)$ uses a precoding matrix $\mathbf{V}_i$ to encode source symbols, and the variables in ξ all take values independently and uniformly at random from $\mathbb{F}_{2^m}$. We use a tuple $\lambda = (\xi, \mathbf{V}_i : 1 \leq i \leq 3)$ to denote a precoding-based linear scheme.

From the above definition, it can be seen that a precoding-based linear scheme is a random linear network coding scheme. Given a precoding-based linear scheme, let P_{succ} denote the probability that the denominators of the precoding matrices are all evaluated to non-zero values, and all receivers can successfully decode their required source symbols from received symbols.

Definition 2.3.2. Given a precoding-based linear scheme $\lambda = (\xi, \mathbf{V}_i : 1 \leq i \leq 3)$, we say that it *achieves* the rate tuple $(\frac{k_1}{N}, \frac{k_2}{N}, \frac{k_3}{N})$, if $\lim_{m\to\infty} P_{succ} = 1$.

Given a precoding-based linear scheme, if the conditions of the above definition is satisfied, by choosing sufficiently large finite field $\mathbb{F}_{2^m}$, a random assignment of values to ξ will enable each receiver to successfully decode its required source symbols with high probability. In this sense, given sufficiently large $\mathbb{F}_{2^m}$, a precoding-based linear scheme works for most random realizations of ξ, but not all realizations.

Before proceeding, we introduce the following Schwartz-Zippel Theorem [56].

Theorem 2.3.1 (Schwartz-Zippel Theorem)**.** Let $Q(x_1, x_2, \cdots, x_n)$ be a non-zero multivariate polynomial of total degree d in the ring $\mathbb{F}[x_1, x_2, \cdots, x_n]$, where $\mathbb{F}$ is a field. Fix a finite

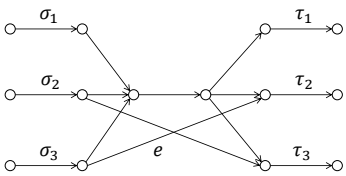


Figure 2.2: An illustrative example for precoding-based linear scheme.

set $S \subseteq \mathbb{F}$. Let $r_1, r_2, \cdots, r_n$ be chosen independently and uniformly at random from S. Then,

$$Pr(Q(r_1, r_2, \cdots, r_n) = 0) \leq \frac{d}{|S|}$$

Example 2.3.1. We use an example to illustrate the above concepts. Consider the network in Fig. 2.2. Note that under the network model considered in the paper, interference is almost unavoidable at the receivers. Consider a receiver d_i. Without loss of generality, assume that the $(1,1)$ element of $\mathbf{V}_j$ $(i \neq j)$ is a non-zero rational function $f_{11}(\xi)$. Thus, the $(1,1)$ element of $\mathbf{M}_{ji}\mathbf{V}_j$ is a non-zero rational function $m_{ji}(\mathbf{x}^{(1)})f_{11}(\xi)$. Due to Theorem 2.3.1, the probability that $m_{ji}(\mathbf{x}^{(1)})f_{11}(\xi)$ is evaluated to zero under a random assignment of values to ξ approaches to zero as $m \to \infty$. Hence, the probability that $\mathbf{M}_{ji}\mathbf{V}_j = \mathbf{0}$ approaches zero as $m \to \infty$. This means that interference is almost unavoidable at d_i.

Next, we present a precoding-based linear scheme that achieves a symmetric rate of $\frac{1}{3}$ per unicast session. Let $N = 3$, and $k_1 = k_2 = k_3 = 1$. Consider the following precoding matrix $\mathbf{V}_1 = (\theta_1^{(1)}\ \theta_1^{(2)}\ \theta_1^{(3)})$. According to Eq. (2.7), the output vector at d_i is $\mathbf{Y}_{\tau_i} = \mathbf{M}_i\mathbf{X}$, where $\mathbf{M}_i$ is as follows:

$$\mathbf{M}_i = \begin{pmatrix} m_{1i}(\mathbf{x}^{(1)})\theta_1^{(1)} & m_{2i}(\mathbf{x}^{(1)})\theta_2^{(1)} & m_{3i}(\mathbf{x}^{(1)})\theta_3^{(1)} \\ m_{1i}(\mathbf{x}^{(2)})\theta_1^{(2)} & m_{2i}(\mathbf{x}^{(2)})\theta_2^{(2)} & m_{3i}(\mathbf{x}^{(2)})\theta_3^{(2)} \\ m_{1i}(\mathbf{x}^{(3)})\theta_1^{(3)} & m_{2i}(\mathbf{x}^{(3)})\theta_2^{(3)} & m_{3i}(\mathbf{x}^{(3)})\theta_3^{(3)} \end{pmatrix}$$

It can be verified that $\det(\mathbf{M}_i)$ is a non-zero polynomial in $\mathbb{F}_{2^m}(\xi)$[3]. Let d be the total degree of $\det(\mathbf{M}_i)$. Due to Theorem 2.3.1, we have:

$$\begin{aligned} P_{succ} \geq & Pr(\det(\mathbf{M}_i) \neq 0) \\ = & 1 - Pr(\det(\mathbf{M}_i) = 0) \geq 1 - \frac{d}{2^m} \end{aligned}$$

Since $\lim_{m\to\infty}(1 - \frac{d}{2^m}) = 1$, it follows that $\lim_{m\to\infty} P_{succ} = 1$. Hence, the above precoding-based linear scheme achieves a symmetric rate $\frac{1}{3}$ per unicast session. As we will show in Section 2.6, using precoding-based alignment scheme, which is a special case of precoding-based linear scheme, each unicast session can achieve a symmetric rate $\frac{1}{2}$ per unicast session, which is the optimal symmetric rate achieved by any precoding-based linear schemes. ■

Table 2.1 summarizes the notations used in this paper, in which $e', e \in E$ and $1 \leq i, j, k \leq 3$.

2.4 Applying Precoding-Based Interference Alignment to Networks

In this section, we first present how to utilize precoding-based interference alignment technique to find a precoding-based linear scheme for $(\mathcal{G}, \Omega)$. Then, we present achievability conditions for PBNA. We then introduce the concept of "coupling relations," which are essential in determining the achievability of PBNA.

Throughout this section, we assume that all the senders are connected to all the receivers via directed paths, *i.e.,* $m_{ij}(\mathbf{x})$ is a non-zero polynomial for all $1 \leq i, j \leq 3$. This is the most challenging case, since each receiver may suffer interference from the other two senders.

[3]It can be seen that each row of $\mathbf{M}_i$ is of the form $(m_{1i}(\mathbf{x})\theta_1 \quad m_{2i}(\mathbf{x})\theta_2 \quad m_{3i}(\mathbf{x})\theta_3)$. Since $m_{1i}(\mathbf{x})\theta_1$, $m_{2i}(\mathbf{x})\theta_2$ and $m_{3i}(\mathbf{x})\theta_3$ are linearly independent, according to Lemma 2.4.2 (see Subsection 2.4.2), $\det(\mathbf{M}_i)$ is a non-zero polynomial.

Table 2.1: Summary of Notations

Notations	Meanings
$\omega_i = (s_i, d_i)$	The ith unicast session, where s_i and d_i are the sender and receiver of ω_i respectively.
σ_i, τ_i	The sender edge and the receiver edge for ω_i.
$\mathbf{X}_i$	A vector that holds all the source symbols transmitted from s_i to d_i.
$\mathbb{F}_{2^m}$	The finite field which forms the support for all the symbols transmitted in the network.
$x_{e'e}$	The coding coefficient used to combine the incoming symbol along e' to the symbol along e.
$\mathbf{x}$	The vector consisting of all the coding coefficients in the network.
$\mathcal{P}_{e'e}$	The set of directed paths from e' to e.
$\mathcal{P}_{ji}$	The set of directed paths from σ_j to τ_i.
$t_P(\mathbf{x})$	The product of coding coefficients along path P. It represents a monomial in a transfer function.
$m_{ji}(\mathbf{x})$	The transfer function from σ_j to τ_i.
$\mathbf{x}^{(t)}$	The vector consisting of all the coding coefficients in the network for time slot t.
ξ	A vector that holds all the coding coefficients in the network for the whole transmission process, and the variables used in the encoding process at all the senders.
$\mathbf{V}_i$	The precoding matrix used to encode the symbols sent by s_i.
$\mathbf{M}_{ji}$	A diagonal matrix, in which the element at coordinate (l, l) is the transfer function $m_{ji}(\mathbf{x}^{(l)})$.
$\mathscr{A}_i$, $\mathscr{B}_i$	The alignment condition and the rank condition for ω_i.
$\mathbf{V}_i^*$	The precoding matrix proposed in [30] (see Eq. (2.12)-(2.14)).
$\mathbf{P}_i$, $\mathbf{T}$	The diagonal matrices used in the reformulated alignment conditions Eq. (2.10) and the reformulated rank conditions $\mathscr{B}_1' \sim \mathscr{B}_3'$.
$\mathbf{I}_n$	The $n \times n$ identity matrix.
$p_i(\mathbf{x})$, $\eta(\mathbf{x})$	The rational functions that form the elements along the diagonals of $\mathbf{P}_i$ and $\mathbf{T}$ respectively
α_{ijk}	The last edge that forms a cut-set between σ_i and $\{\tau_j, \tau_k\}$ in a topological ordering of the edges in the network.
β_{ijk}	The first edge that forms a cut-set between $\{\sigma_j, \alpha_{ijk}\}$ and τ_k in a topological ordering of the edges in the network.
$\mathcal{C}_{e'e}$	The set of edges that forms a cut-set between e' and e.
$\mathcal{C}_{ij}$	The set of edges that forms a cut-set between σ_i and τ_j.
$\gcd(f(x), g(x))$	The greatest common divisor of two polynomials $f(x)$ and $g(x)$.

This case also models most practical communication scenarios, in which it is common that all the senders are connected to all the receivers. The other setting, where some sender s_i is disconnected from some receiver d_j $(i \neq j)$, *i.e.*, $m_{ij}(\mathbf{x})$ is a zero polynomial, is easier to deal with, since there is less interference at receivers. We defer the later case to Section 2.6, where we show that this case can be handled similarly as the first case.

2.4.1 Precoding-Based Network Alignment Scheme

In this section, we present how to apply interference alignment to networks to construct a precoding-based linear scheme for $(\mathcal{G}, \Omega)$. The basic idea is that under linear network coding, the network behaves like a wireless interference channel[4], which is shown below:

$$U_i = H_{1i}W_1 + H_{2i}W_2 + H_{3i}W_3 + N_i \quad i = 1, 2, 3 \tag{2.8}$$

where W_j, H_{ji}, U_i, and N_i $(j = 1, 2, 3)$ are all complex numbers, representing the transmitted signal at sender j, the channel gain from sender j to receiver i, the received signal at receiver j, and the noise term respectively. As we can see from Eq. (2.2), in a network equipped with linear network coding, $\hat{X}_j$'s $(j \neq i)$ play the roles of interfering signals, and transfer functions the roles of channel gains. This analogy enables us to borrow some techniques, such as precoding-based interference alignment [30], which is originally developed for the wireless interference channel, to the network setting.

A precoding-based network alignment scheme is defined as follows:

Definition 2.4.1. Given a SISO scenario $(\mathcal{G}, \Omega)$, $n \in \mathbb{Z}_{>0}$, and $s \in \{0, 1\}$, a precoding-based network alignment scheme with $2n + s$ symbol extensions, or a PBNA for short, is a precoding-based linear scheme $\lambda = (\xi, \mathbf{V}_i : 1 \leq i \leq 3)$, which satisfies the following conditions:

1. $\mathbf{V}_1$ is a $(2n + s) \times (n + s)$ matrix with rank $n + s$ on $\mathbb{F}_{2^m}(\xi)$, and $\mathbf{V}_2, \mathbf{V}_3$ are both $(2n + s) \times n$ matrices with rank n on $\mathbb{F}_{2^m}(\xi)$.

[4]The wireless interference channel that we consider here has only one sub-channel.

2. The following equations are satisfied [30]:

$$\mathscr{A}_1 : \text{span}(\mathbf{M}_{21}\mathbf{V}_2) = \text{span}(\mathbf{M}_{31}\mathbf{V}_3)$$

$$\mathscr{A}_2 : \text{span}(\mathbf{M}_{32}\mathbf{V}_3) \subseteq \text{span}(\mathbf{M}_{12}\mathbf{V}_1)$$

$$\mathscr{A}_3 : \text{span}(\mathbf{M}_{23}\mathbf{V}_2) \subseteq \text{span}(\mathbf{M}_{13}\mathbf{V}_1)$$

where for a matrix $\mathbf{E}$, span($\mathbf{E}$) denotes the linear space spanned by the column vectors contained in $\mathbf{E}$.

3. The variables in ξ all take values independently and uniformly at random from $\mathbb{F}_{2^m}$.

Definition 2.4.2. Given a SISO scenario $(\mathcal{G}, \Omega)$, and a rate tuple $(R_1, R_2, R_3) \in \mathbb{Q}^3_{>0}$, we say that (R_1, R_2, R_3) is asymptotically achievable through PBNA, if there exists a sequence $(\lambda_n)_{n=1}^{\infty}$, where each λ_n is a PBNA for $(\mathcal{G}, \Omega)$, such that each λ_n achieves a rate tuple $\mathbf{r}_n \in \mathbb{Q}^3_{>0}$, and $\lim_{n\to\infty} \mathbf{r}_n = (R_1, R_2, R_3)$.

In the above definition, $\mathscr{A}_i$ $(1 \leq i \leq 3)$ is called the alignment condition for ω_i. It guarantees that the undesired symbols or interferences at each receiver are mapped into a single linear space, such that the dimension of received symbols or the number of unknowns is decreased.

2.4.2 Achievability Conditions of PBNA

The following lemma provides sufficient conditions for PBNA schemes to achieve the rate tuple $(\frac{n+s}{2n+s}, \frac{n}{2n+s}, \frac{n}{2n+s})$.

Lemma 2.4.1. Assume that all the senders and all the receivers are connected via directed paths. Consider a PBNA $\lambda = (\xi, \mathbf{V}_i : 1 \leq i \leq 3)$. It achieves the rate tuple $(\frac{n+2}{2n+s}, \frac{n}{2n+s}, \frac{n}{2n+s})$,

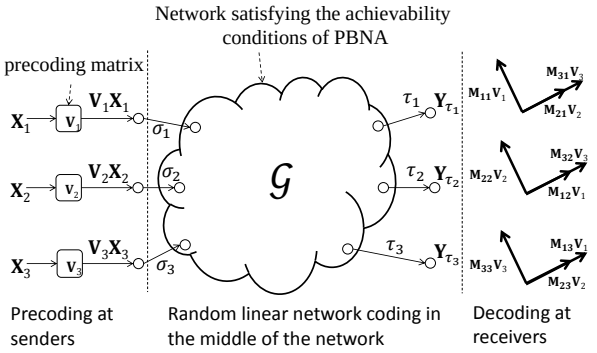


Figure 2.3: Applying precoding-based interference alignment to a network which satisfies the rank conditions of PBNA as per Lemma 2.4.1. At each sender edge σ_i $(i = 1, 2, 3)$, the input vector $\mathbf{X}_i$ is first encoded into $2n + s$ symbols through the precoding matrix $\mathbf{V}_i$; then the encoded symbols are transmitted through the network in $2n + s$ time slots via random linear network coding in the middle of the network; at each receiver edge τ_i, the undesired symbols are aligned into a single linear space, which is linearly indepdent from the linear space spanned by the desired signals, such that the receiver can decode all the desired symbols.

if the following conditions are satisfied [30]:

$$\mathscr{B}_1 : \text{rank}(\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{21}\mathbf{V}_2) = 2n + s$$

$$\mathscr{B}_2 : \text{rank}(\mathbf{M}_{12}\mathbf{V}_1 \quad \mathbf{M}_{22}\mathbf{V}_2) = 2n + s$$

$$\mathscr{B}_3 : \text{rank}(\mathbf{M}_{13}\mathbf{V}_1 \quad \mathbf{M}_{33}\mathbf{V}_3) = 2n + s$$

Proof. Suppose $\mathscr{B}_1 \sim \mathscr{B}_3$ are satisfied. Define the following matrices:

$$\mathbf{D}_1 = (\mathbf{M}_{11}\mathbf{V}_1 \quad \mathbf{M}_{21}\mathbf{V}_2)^{-1}$$

$$\mathbf{D}_2 = (\mathbf{M}_{12}\mathbf{V}_1 \quad \mathbf{M}_{22}\mathbf{V}_2)^{-1}$$

$$\mathbf{D}_3 = (\mathbf{M}_{13}\mathbf{V}_1 \quad \mathbf{M}_{33}\mathbf{V}_3)^{-1}$$

Let $f_i(\xi)$ denote the product of the denominators of all the elements in $\mathbf{V}_i$, and $g_i(\xi)$ the product of the denominators of all the elements in $\mathbf{D}_i$. Thus, $f_i(\xi), g_i(\xi)$ are both non-zero polynomials in $\mathbb{F}_{2^m}[\xi]$. Define $q(\xi) = \prod_{i=1}^{3} f_i(\xi)g_i(\xi)$. Let d denote the total degree of $q(\xi)$. Suppose ξ_0 is an assignment of values to ξ such that $q(\xi_0) \neq 0$. Hence, the denominators

of the elements in $\mathbf{V}_i$'s and $\mathbf{D}_i$'s are evaluated to non-zeros. Moreover, $\mathbf{X}_i$ is a sub-vector of $\mathbf{D}_i|_{\xi_0}\mathbf{Y}_{\tau_i}$, where $\mathbf{D}_i|_{\xi_0}$ is a matrix acquired through evaluating each element of $\mathbf{D}_i$ under the assignment $\xi = \xi_0$. Thus, all the receivers can decode their required source symbols. Hence, the probability P_{succ} that all the receivers can decoded their required source symbols satisfies the following inequalities:

$$P_{succ} \geq Pr(q(\xi) \neq 0) = 1 - Pr(q(\xi) = 0) \geq 1 - \frac{d}{2^m}$$

where the last inequality follows from Theorem 2.3.1. Since $\lim_{m\to\infty}(1 - \frac{d}{2^m}) = 1$, we have $\lim_{m\to\infty} P_{succ} = 1$. Hence, λ achieves $(\frac{n+s}{2n+s}, \frac{n}{2n+s}, \frac{n}{2n+s})$. ■

In Lemma 2.4.1, $\mathscr{B}_i$ $(1 \leq i \leq 3)$ are called the rank condition for ω_i. $\mathscr{B}_i$ guarantees that d_i can decode its required source symbol with high probability when the the size of $\mathbb{F}_{2^m}$ is sufficiently large. In Fig. 2.3, we use a figure to illustrate how to apply PBNA to a network which satisfies the rank conditions.

We can further simplify the alignment conditions as follows. First, we reformulate $\mathscr{A}_1 \sim \mathscr{A}_3$ as follows:

$$\mathscr{A}_1' : \mathbf{M}_{21}\mathbf{V}_2 = \mathbf{M}_{31}\mathbf{V}_3\mathbf{A}$$
$$\mathscr{A}_2' : \mathbf{M}_{32}\mathbf{V}_3 = \mathbf{M}_{12}\mathbf{V}_1\mathbf{B}$$
$$\mathscr{A}_3' : \mathbf{M}_{23}\mathbf{V}_2 = \mathbf{M}_{13}\mathbf{V}_1\mathbf{C}$$

where $\mathbf{A}$ is an $n \times n$ invertible matrix, and $\mathbf{B}$, $\mathbf{C}$ are both $(n+s) \times n$ matrices with rank n. A direct consequence of $\mathscr{A}_2'$ and $\mathscr{A}_3'$ is that the precoding matrices are not independent from each other: Both $\mathbf{V}_2$ and $\mathbf{V}_3$ are determined by $\mathbf{V}_1$ through the following equations:

$$\mathbf{V}_2 = \mathbf{M}_{13}\mathbf{M}_{23}^{-1}\mathbf{V}_1\mathbf{C} \quad \mathbf{V}_3 = \mathbf{M}_{12}\mathbf{M}_{32}^{-1}\mathbf{V}_1\mathbf{B} \tag{2.9}$$

Substituting the above equations into $\mathscr{A}_1'$, the three alignment conditions can be further consolidated into a single equation:

$$\mathbf{T}\mathbf{V}_1\mathbf{C} = \mathbf{V}_1\mathbf{B}\mathbf{A} \tag{2.10}$$

where $\mathbf{T} = \mathbf{M}_{13}\mathbf{M}_{21}\mathbf{M}_{32}\mathbf{M}_{12}^{-1}\mathbf{M}_{23}^{-1}\mathbf{M}_{31}^{-1}$. Eq. (2.10) suggests that alignment conditions introduce constraint on $\mathbf{V}_1$. Thus, in general, we cannot choose $\mathbf{V}_1$ freely.

Finally, using Eq. (2.9) and Eq. (2.10), the rank conditions are transformed into the following equivalent equations:

$$\begin{aligned}
\mathscr{B}_1' &: \quad \operatorname{rank}(\mathbf{V}_1 \quad \mathbf{P}_1\mathbf{V}_1\mathbf{C}) = 2n + s \\
\mathscr{B}_2' &: \quad \operatorname{rank}(\mathbf{V}_1 \quad \mathbf{P}_2\mathbf{V}_1\mathbf{C}) = 2n + s \\
\mathscr{B}_3' &: \quad \operatorname{rank}(\mathbf{V}_1 \quad \mathbf{P}_3\mathbf{V}_1\mathbf{C}\mathbf{A}^{-1}) = 2n + s
\end{aligned}$$

where $\mathbf{P}_1 = \mathbf{M}_{13}\mathbf{M}_{21}\mathbf{M}_{11}^{-1}\mathbf{M}_{23}^{-1}$, $\mathbf{P}_2 = \mathbf{M}_{13}\mathbf{M}_{22}\mathbf{M}_{12}^{-1}\mathbf{M}_{23}^{-1}$, and $\mathbf{P}_3 = \mathbf{M}_{21}\mathbf{M}_{33}\mathbf{M}_{23}^{-1}\mathbf{M}_{31}^{-1}$. Recalling each $\mathbf{M}_{kl}$ $(1 \leq k, l \leq 3)$ is a diagonal matrix (see Eq. (2.6)) with the elements along the diagonal being of the form $m_{kl}(\mathbf{x})$, $\mathbf{P}_i$ and $\mathbf{T}$ are both diagonal matrices. Define the following functions:

$$\begin{aligned}
p_1(\mathbf{x}) &= \frac{m_{13}(\mathbf{x})m_{21}(\mathbf{x})}{m_{11}(\mathbf{x})m_{23}(\mathbf{x})} \quad p_2(\mathbf{x}) = \frac{m_{13}(\mathbf{x})m_{22}(\mathbf{x})}{m_{12}(\mathbf{x})m_{23}(\mathbf{x})} \\
p_3(\mathbf{x}) &= \frac{m_{21}(\mathbf{x})m_{33}(\mathbf{x})}{m_{23}(\mathbf{x})m_{31}(\mathbf{x})} \quad \eta(\mathbf{x}) = \frac{m_{13}(\mathbf{x})m_{21}(\mathbf{x})m_{32}(\mathbf{x})}{m_{12}(\mathbf{x})m_{23}(\mathbf{x})m_{31}(\mathbf{x})}
\end{aligned} \tag{2.11}$$

It can been seen that $p_i(\mathbf{x})$ and $\eta(\mathbf{x})$ form the elements along the diagonals of $\mathbf{P}_i$ and $\mathbf{T}$ respectively.

Next, we reformulate the rank conditions in terms of $p_i(\mathbf{x})$ and $\eta(\mathbf{x})$. To this end, we need to know the internal structure of $\mathbf{V}_1$. We distinguish the following two cases:

Case I: $\eta(\mathbf{x})$ is non-constant, and thus $\mathbf{T}$ is not an identity matrix. For this case, Eq. (2.10) becomes non-trivial, and we cannot choose $\mathbf{V}_1$ freely. We use the following precoding matrices proposed by Cadambe and Jafar [30]:

$$\mathbf{V}_1^* = (\mathbf{w} \quad \mathbf{Tw} \quad \cdots \quad \mathbf{T}^n\mathbf{w}) \tag{2.12}$$

$$\mathbf{V}_2^* = \mathbf{M}_{13}\mathbf{M}_{23}^{-1}(\mathbf{w} \quad \mathbf{Tw} \quad \cdots \quad \mathbf{T}^{n-1}\mathbf{w}) \tag{2.13}$$

$$\mathbf{V}_3^* = \mathbf{M}_{12}\mathbf{M}_{32}^{-1}(\mathbf{Tw} \quad \mathbf{T}^2\mathbf{w} \quad \cdots \quad \mathbf{T}^n\mathbf{w}) \tag{2.14}$$

where $\mathbf{w}$ is a column vector of $2n+1$ ones. The above precoding matrices correspond to the configuration where $s = 1$, $\mathbf{A} = \mathbf{I}_n$, $\mathbf{C}$ consists of the left n columns of $\mathbf{I}_{n+1}$, and $\mathbf{B}$ the right n columns of $\mathbf{I}_{n+1}$. It is straightforward to verify that the above precoding matrices satisfy the alignment conditions.

We consider the following matrix,

$$\mathbf{H} = \begin{pmatrix} f_1(\mathbf{y}_1) & f_2(\mathbf{y}_1) & \cdots & f_r(\mathbf{y}_1) \\ f_1(\mathbf{y}_2) & f_2(\mathbf{y}_2) & \cdots & f_r(\mathbf{y}_2) \\ \cdots & \cdots & \cdots & \cdots \\ f_1(\mathbf{y}_r) & f_2(\mathbf{y}_r) & \cdots & f_r(\mathbf{y}_r) \end{pmatrix}$$

where $f_i(\mathbf{y})$ $(i = 1, 2, \cdots, r)$ is a rational function in terms of a vector of variables $\mathbf{y} = (y_1, \cdots, y_k)$ in $\mathbb{F}_{2^m}(\mathbf{y})$, and the jth row of $\mathbf{H}$ is simply a repetition of the vector $(f_1(\mathbf{y}), \cdots, f_r(\mathbf{y}))$, with $\mathbf{y}$ being replaced by a vector of variables $\mathbf{y}_j = (y_{j1}, \cdots, y_{jk})$. Due to the particular structure of $\mathbf{H}$, the problem of checking whether $\mathbf{H}$ is full rank can be simplified to checking whether $f_1(\mathbf{y}), \cdots, f_r(\mathbf{y})$ are linearly independent, as stated in the following lemma. Here, $f_1(\mathbf{y}), \cdots, f_r(\mathbf{y})$ are said to be linearly independent, if for any scalars $a_1, \cdots, a_r \in \mathbb{F}_q$, which are not all zeros, $a_1 f_1(\mathbf{y}) + \cdots + a_r f_r(\mathbf{y}) \neq 0$.

Lemma 2.4.2. $\det(\mathbf{H}) \neq 0$ if and only if $f_1(\mathbf{y}), \cdots, f_r(\mathbf{y})$ are linearly independent.

Proof. See Theorem 1 of [53]. ■

An important observation is that using the precoding matrices defined in Eq. (2.12)-(2.14), all of the matrices involved in $\mathscr{B}'_1, \mathscr{B}'_2, \mathscr{B}'_3$ have the same form as $\mathbf{H}$. Specifically, each row of the matrix in $\mathscr{B}'_i$ is of the form:

$$(1 \quad \eta(\mathbf{x}) \quad \cdots \quad \eta^n(\mathbf{x}) \quad p_i(\mathbf{x}) \quad \cdots \quad p_i(\mathbf{x})\eta^{n-1}(\mathbf{x})) \tag{2.15}$$

where for $1 \leq j \leq n+1$, the jth element is $\eta^{j-1}(\mathbf{x})$, and for $n+2 \leq j \leq 2n+1$, the jth element is $p_i(\mathbf{x})\eta^{j-n-2}(\mathbf{x})$. Hence, using Lemma 2.4.2, we can quickly derive:

Lemma 2.4.3. Assume that all the senders are connected to all the receivers via directed paths, and $\eta(\mathbf{x})$ is non-constant. Consider a PBNA $\lambda_n = (\xi, \mathbf{V}_i : 1 \leq i \leq 3)$, where $\mathbf{V}_i$ is defined in Eq. (2.12)-(2.14). λ_n achieves the rate tuple $(\frac{n+1}{2n+1}, \frac{n}{2n+1}, \frac{n}{2n+1})$, if for each $1 \leq i \leq 3$, the following condition is satisfied:[5]

$$p_i(\mathbf{x}) \notin \mathcal{S}_n = \left\{ \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} : f(z), g(z) \in \mathbb{F}_q[z], f(z)g(z) \neq 0, \right. \\ \left. \gcd(f(z), g(z)) = 1, d_f \leq n, d_g \leq n-1 \right\} \tag{2.16}$$

Proof. If Eq. (2.16) is satisfied, the rational functions in Eq. (2.15) are linearly independent. Therefore, due to Lemma 2.4.2, condition $\mathscr{B}'_i$ is satisfied. Hence, due to Lemma 2.4.1, $(\frac{n+1}{2n+1}, \frac{n}{2n+1}, \frac{n}{2n+1})$ is achieved by λ_n. ■

Note that each rational function $\frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}_n$ represents a constraint on $p_i(\mathbf{x})$, i.e., $p_i(\mathbf{x}) \neq \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))}$, the violation of which invalidates the use of the PBNA for achieving the rate tuple $(\frac{n+1}{2n+1}, \frac{n}{2n+1}, \frac{n}{2n+1})$ through the precoding matrices defined in Eq. (2.12)-(2.14). Also note

[5]Notation: For two polynomials $f(x)$ and $g(x)$, let $\gcd(f(x), g(x))$ denote their greatest common divisor, and d_f the degree of $f(x)$.

that Eq. (2.16) only guarantees that PBNA achieves a symmetrical rate close to one half. In order for each unicast session to asymptotically achieve a transmission rate of one half, we simply combine the conditions of Lemma 2.4.3 for all possible values of n, and get the following result:

Theorem 2.4.1. Assume that all the senders are connected to all the receivers via directed paths, and $\eta(\mathbf{x})$ is non-constant. The symmetrical rate $\frac{1}{2}$ is asymptotically achievable through PBNA, if for each $1 \leq i \leq 3$,

$$p_i(\mathbf{x}) \notin \mathcal{S}' = \left\{ \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} : f(z), g(z) \in \mathbb{F}_q[z], f(z)g(z) \neq 0, \right. \\ \left. \gcd(f(z), g(z)) = 1 \right\} \tag{2.17}$$

Proof. Consider the PBNA scheme λ_n defined in Lemma 2.4.3. If Eq. (2.17) is satisfied, Eq. (2.16) is satisfied, and thus λ_n achieves the rate tuple $(\frac{n+1}{2n+1}, \frac{n}{2n+1}, \frac{n}{2n+1})$. Since $\lim_{n\to\infty}(\frac{n+1}{2n+1}, \frac{n}{2n+1}, \frac{n}{2n+1}) = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$. This implies that the symmetrical rate $\frac{1}{2}$ is asymptotically achievable through PBNA. ■

Case II: $\eta(\mathbf{x})$ is constant, and thus $\mathbf{T}$ is an identity matrix. For this case, Eq. (2.10) becomes trivial. In fact, we set $n = 1$, $s = 0$, and $\mathbf{BA} = \mathbf{C}$, and hence Eq. (2.10) can be satisfied by any *arbitrary* $\mathbf{V}_1$. Specifically, we use the following precoding matrices:

$$\mathbf{V}_1 = (\theta_1 \quad \theta_2)^T \tag{2.18}$$

$$\mathbf{V}_2 = \mathbf{M}_{13}\mathbf{M}_{23}^{-1}(\theta_1 \quad \theta_2)^T \tag{2.19}$$

$$\mathbf{V}_3 = \mathbf{M}_{12}\mathbf{M}_{32}^{-1}(\theta_1 \quad \theta_2)^T \tag{2.20}$$

where θ_1, θ_2 are variables. The above precoding matrices correspond to the configuration where $\mathbf{A} = \mathbf{B} = \mathbf{C} = \mathbf{I}_2$. Using the above precoding matrices, $\mathscr{A}_1 \sim \mathscr{A}_3$ all become equalities,

i.e., the interfering signals are perfectly aligned into a single linear space. Meanwhile, using these precoding matrices, each row of the matrix in $\mathscr{B}_i'$ is of the following form:

$$(\theta \quad p_i(\mathbf{x})\theta) \tag{2.21}$$

Hence, using Lemma 2.4.2, we can quickly derive:

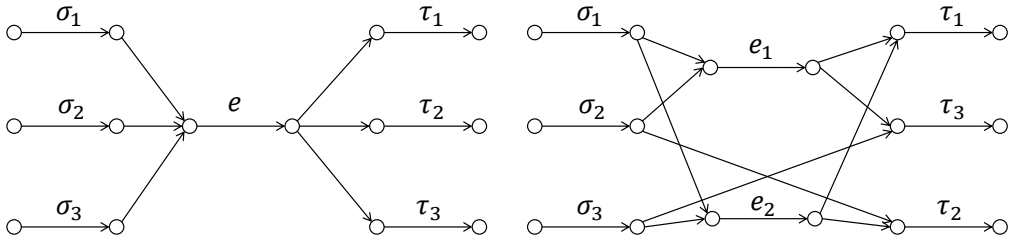
Theorem 2.4.2. Assume that all the senders are connected to all the receivers via directed paths, and $\eta(\mathbf{x})$ is constant. Consider the PBNA scheme $\lambda = (\xi, \mathbf{V}_i : 1 \leq i \leq 3)$, where the precoding matrices are defined in Eq. (2.18)-(2.20). Then λ achieves the symmetrical rate $\frac{1}{2}$, if for each $1 \leq i \leq 3$, $p_i(\mathbf{x})$ is non-constant.

Proof. If $p_i(\mathbf{x})$ is not constant, the functions in Eq. (2.21) are linearly independent, and therefore $\mathscr{B}_i'$ is satisfied due to Lemma 2.4.2. Thus, $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ is achieved by λ according to Lemma 2.4.1. ■

As shown in the above theorem, if $\eta(\mathbf{x})$ is constant, each unicast session can achieve one half rate in exactly two time slots by using PBNA.

2.4.3 Coupling Relations and Achievability of PBNA

In the previous section, we reformulated the achievability conditions of PBNA in terms of the functions $p_i(\mathbf{x})$ and $\eta(\mathbf{x})$. One critical question is: What is the connection between the reformulated conditions and network topology? We start by illustrating that through examples of networks whose structure violates these conditions. Let's first consider the network shown in Fig. 2.4a. Due to the bottleneck e, it can be easily verified that $p_1(\mathbf{x}) = p_2(\mathbf{x}) = p_3(\mathbf{x}) = \eta(\mathbf{x}) = 1$, and thus the conditions of Theorem 2.4.2 are violated. Moreover, consider the network shown in Fig. 2.4b. It can be easily verified that for this network,



(a) $p_1(\mathbf{x}) = p_2(\mathbf{x}) = p_3(\mathbf{x}) = \eta(\mathbf{x}) = 1$) (b) $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$

Figure 2.4: Examples of realizable coupling relations: The left network realizes the coupling relations $p_i(\mathbf{x}) = \eta(\mathbf{x}) = 1$ such that the conditions of Theorem 2.4.2 are violated; in the right network, $\eta(\mathbf{x}) \neq 1$, but $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$, which violates the conditions of Theorem 2.4.1.

$\eta(\mathbf{x}) \neq 1$, and $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$. Thus the conditions of Theorem 2.4.1 are violated. Moreover, by exchanging $\sigma_1 \leftrightarrow \sigma_2$ and $\tau_1 \leftrightarrow \tau_2$, we obtain another example, where $p_2(\mathbf{x}) = 1 + \eta(\mathbf{x})$, and thus the conditions of Theorem 2.4.1 are again violated. While the key feature of the first example can be easily identified, it is not obvious what are the defining features of the second example. Nevertheless, both examples demonstrate an important difference between networks and wireless interference channel: In networks, due to the internal structure of transfer functions, network topology might introduce dependence between different transfer functions, e.g., $p_1(\mathbf{x}) = 1$ or $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$; in contrast, in wireless channel, channel gains are algebraically independent almost surely.

The above dependence relations can be seen as special cases of coupling relations, as defined below.

Definition 2.4.3. A *coupling relation* is an equation in the following form:

$$f(m_{i_1 j_1}(\mathbf{x}), m_{i_2 j_2}(\mathbf{x}), \cdots, m_{i_k j_k}(\mathbf{x})) = 0 \tag{2.22}$$

where $f(z_1, z_2, \cdots, z_k)$ is a polynomial in $\mathbb{F}_{2^m}[z_1, \cdots, z_k]$, $1 \leq i_l, j_l \leq 3$ for $1 \leq l \leq k$. If there exists a network $\mathcal{G}$ such that the transfer functions $m_{i_1 j_1}(\mathbf{x})$, $m_{i_2 j_2}(\mathbf{x})$, $\cdots$, $m_{i_k j_k}(\mathbf{x})$ satisfy the above equation, we say that the coupling relation Eq. (2.22) is *realizable*, or $\mathcal{G}$

realizes the coupling relation Eq. (2.22).

As shown in Theorem 2.4.1, each rational function $\frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}'$ represents a coupling relation $p_i(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))}$.

The existence of coupling relations greatly complicates the achievability problem of PBNA. As shown previously, most of the coupling relations, such as $p_1(\mathbf{x}) = 1$ and $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$, are harmful to PBNA, because their presence violates the conditions of Theorems 2.4.1 and 2.4.2. The only exception is $\eta(\mathbf{x}) = 1$, which does help simplify the construction of precoding matrices, and thus is beneficial to PBNA. Indeed, as shown in Theorem 2.4.2, this coupling relation allows interferences to be perfectly aligned at each receiver, and each unicast session can achieve one half rate in exactly two time slots. Unfortunately, as we will see in Section **??**, this coupling relation requires that the network possesses particular structures, which are absent in most networks. For this reason, we will mainly focus on the case $\eta(\mathbf{x}) \neq 1$, which is applicable for most networks.

One interesting observation is that not all coupling relations are realizable. For example, consider the coupling relation $p_1(\mathbf{x}) = \eta^3(\mathbf{x})$, where both $p_1(\mathbf{x})$ and $\eta(\mathbf{x})$ are non-constants. Let $p_1(\mathbf{x}) = \frac{u(\mathbf{x})}{v(\mathbf{x})}$, $\eta(\mathbf{x}) = \frac{s(\mathbf{x})}{t(\mathbf{x})}$ denote the *unique forms* [6] of $p_1(\mathbf{x})$ and $\eta(\mathbf{x})$ respectively. Consider a coding variable $x_{ee'}$ that appears in both $\frac{u(\mathbf{x})}{v(\mathbf{x})}$ and $\frac{s(\mathbf{x})}{t(\mathbf{x})}$. Because the maximum degree of each coding variable in a transfer function is at most one, according to Eq. (2.11), the maximum of the degrees of $x_{ee'}$ in $u(\mathbf{x})$ and $v(\mathbf{x})$ is at most two. However, it can be easily seen that the maximum of the degrees of $x_{ee'}$ in $s^3(\mathbf{x})$ and $t^3(\mathbf{x})$ is at least three. Therefore, it is impossible that $p_1(\mathbf{x}) = \eta^3(\mathbf{x})$. This example suggests that there exists significant redundancy in the conditions of Theorem 2.4.1. More formally, it raises the following important question:

[6] For a non-zero rational function $h(\mathbf{y}) \in \mathbb{F}_q(\mathbf{y})$, its unique form is defined as $h(\mathbf{y}) = \frac{f(\mathbf{y})}{g(\mathbf{y})}$, where $f(\mathbf{y}), g(\mathbf{y}) \in \mathbb{F}_q[\mathbf{y}]$ and $\gcd(f(\mathbf{y}), g(\mathbf{y})) = 1$.

Q1: Which coupling relations $p_i(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}'$ are realizable?

The answer to this question allows us to reduce the set $\mathcal{S}'$ defined in Theorem 2.4.1 to its minimal size. For $i = 1, 2, 3$, we define the following set, which represents the minimal set of coupling relations we need to consider:

$$\mathcal{S}'_i = \left\{ \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}' : p_i(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \text{ is realizable} \right\} \tag{2.23}$$

Then the next important question is:

Q2: Given $p_i(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}'_i$, what are the defining features of the networks for which this coupling relation holds?

As we will see in the rest of this paper, the answers to Q1 and Q2 both lie in a deeper understanding of the properties of transfer functions. Intuitively, because each transfer function is defined on a graph, it usually possesses special properties. The graph-related properties not only allow us to reduce $\mathcal{S}'$ to the minimal set $\mathcal{S}'_i$, but also enable us to identify the defining features of the networks which realize the coupling relations represented by $\mathcal{S}'_i$.

In the derivation of Theorem 2.4.1, we only consider the precoding matrices defined in Eq. (2.12)-(2.14). However, the choices of precoding matrices are not limited to these matrices. In fact, as we will see in Section 2.6, given different $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$, we can derive different precoding matrix $\mathbf{V}_1$ such that Eq. (2.10) is satisfied. This raises the following interesting question:

Q3: Assume some coupling relation $p_i(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}'_i$ is present in the network. Is it still possible to utilize PBNA via other precoding matrices instead of those defined in Eq. (2.12)-(2.14)?

As we will see in Section 2.6, the answer to this question is negative. The basic idea is that

each precoding matrix $\mathbf{V}_1$ that satisfies Eq. (2.10) can be transformed into the precoding matrix in Eq. (2.12) through a transform equation $\mathbf{V}_1^* = \mathbf{G}^{-1}\mathbf{V}_1\mathbf{F}^{-1}$, where $\mathbf{G}$ is a diagonal matrix and $\mathbf{F}$ a full-rank matrix (See Lemma 2.6.3). Using this transform equation, we can prove that if the precoding matrices cannot be used due to the presence of a coupling relation, then any precoding matrices cannot be used.

2.5 Overview of Results

In this section, we state our main results. Proofs are deferred to Appendices.

2.5.1 Sufficient and Necessary Conditions for PBNA to Achieve Symmetrical Rate $\frac{1}{2}$

Since the construction of $\mathbf{V}_1$ depends on whether $\eta(\mathbf{x})$ is constant, we distinguish two cases.

Case 1: $\eta(\mathbf{x})$ Is Not Constant

Theorem 2.5.1 (The Main Theorem)**.** Assume that all the senders are connected to all the receivers via directed paths, and $\eta(\mathbf{x})$ is not constant. 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:

$$\begin{aligned} m_{11}(\mathbf{x}) \neq & \frac{m_{13}(\mathbf{x})m_{21}(\mathbf{x})}{m_{23}(\mathbf{x})}, \frac{m_{12}(\mathbf{x})m_{31}(\mathbf{x})}{m_{32}(\mathbf{x})}, \\ & \frac{m_{13}(\mathbf{x})m_{21}(\mathbf{x})}{m_{23}(\mathbf{x})} + \frac{m_{12}(\mathbf{x})m_{31}(\mathbf{x})}{m_{32}(\mathbf{x})} \end{aligned} \tag{2.24}$$

$$m_{22}(\mathbf{x}) \neq \frac{m_{12}(\mathbf{x})m_{23}(\mathbf{x})}{m_{13}(\mathbf{x})}, \frac{m_{32}(\mathbf{x})m_{21}(\mathbf{x})}{m_{31}(\mathbf{x})}, \frac{m_{12}(\mathbf{x})m_{23}(\mathbf{x})}{m_{13}(\mathbf{x})} + \frac{m_{32}(\mathbf{x})m_{21}(\mathbf{x})}{m_{31}(\mathbf{x})} \tag{2.25}$$

$$m_{33}(\mathbf{x}) \neq \frac{m_{23}(\mathbf{x})m_{31}(\mathbf{x})}{m_{21}(\mathbf{x})}, \frac{m_{13}(\mathbf{x})m_{32}(\mathbf{x})}{m_{12}(\mathbf{x})}, \frac{m_{23}(\mathbf{x})m_{31}(\mathbf{x})}{m_{21}(\mathbf{x})} + \frac{m_{13}(\mathbf{x})m_{32}(\mathbf{x})}{m_{12}(\mathbf{x})} \tag{2.26}$$

Proof. See Appendix B. ■

Eq. (2.24)-(2.26) can be reformulated into the following equivalent conditions:

$$p_1(\mathbf{x}) \notin \mathcal{S}_1' = \left\{ 1, \eta(\mathbf{x}), \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})} \right\} \tag{2.27}$$

$$p_2(\mathbf{x}) \notin \mathcal{S}_2' = \{1, \eta(\mathbf{x}), 1+\eta(\mathbf{x})\} \tag{2.28}$$

$$p_3(\mathbf{x}) \notin \mathcal{S}_3' = \{1, \eta(\mathbf{x}), 1+\eta(\mathbf{x})\} \tag{2.29}$$

Note that in Theorem 2.5.1, we reduce the conditions of Theorem 2.4.1 to its minimal size, such that each $\mathcal{S}_i'$ as defined in Eq. (2.27)-(2.29) represents the minimal set of coupling relations that are realizable. Moreover, as we will see later, each of these coupling relations has a unique interpretation in terms of the network topology. The interpretations further provide polynomial-time algorithms to check the existence of these coupling relations.

The conditions of the Main Theorem can be understood from the perspective of the interference channel. As shown in Section 2.4.1, under linear network coding, the network behaves as a 3-user wireless interference channel, where the channel coefficients $m_{ij}(\mathbf{x})$ are all non-zeros. Let $\mathbf{H}$ denote the matrix with the (i,j)-element being $m_{ij}(\mathbf{x})$. It is easy to see that the first two inequalities in Eq. (2.24)-(2.26) can be rewritten as $M_{kl}(\mathbf{H}) \neq 0$ for some

$k \neq l$, where $M_{kl}(\mathbf{H})$ denotes the (k,l)-Minor of $\mathbf{H}$. For example, $m_{11}(\mathbf{x}) \neq \frac{m_{13}(\mathbf{x})m_{21}(\mathbf{x})}{m_{23}(\mathbf{x})}$ is equivalent to $M_{32}(\mathbf{H}) \neq 0$, and $m_{11}(\mathbf{x}) \neq \frac{m_{12}(\mathbf{x})m_{31}(\mathbf{x})}{m_{32}(\mathbf{x})}$ is equivalent to $M_{23}(\mathbf{H}) \neq 0$. Suppose that there exists $M_{kl}(\mathbf{H}) = 0$ for some $k \neq l$. For such a channel, it is known that the sum-rate achieved by the three unicast sessions cannot be more than 1 in the information theoretical sense (see Lemma 1 of [57]), i.e., no precoding-based linear scheme can achieve a rate beyond 1/3 per user. Therefore, given that all senders are connected to all receivers, the condition $M_{kl}(\mathbf{H}) \neq 0$ is information theoretically necessary for achievable rate 1/2 per session. Hence, the first two inequalities of Eq. (2.24)-(2.26) are simply the information theoretic necessary conditions, so they must hold for any precoding-based linear schemes.

Case II: $\eta(\mathbf{x})$ Is Constant

In this case, we can choose $\mathbf{V}_1$ freely by setting $\mathbf{BA} = \mathbf{C}$. As stated in the following theorem, each unicast session can achieve one half rate in exactly two time slots.

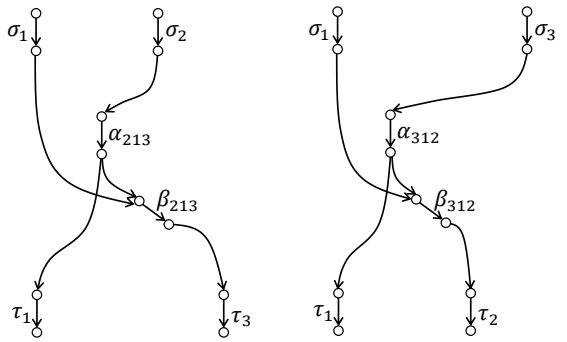
Theorem 2.5.2. Assume that all the senders are connected to all the receivers via directed paths, and $\eta(\mathbf{x})$ is constant. The three unicast sessions can achieve the rate tuple $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ in exactly two time slots through PBNA if and only if the following conditions are satisfied:

$$m_{11}(\mathbf{x}) \neq \frac{m_{13}(\mathbf{x})m_{21}(\mathbf{x})}{m_{23}(\mathbf{x})} \tag{2.30}$$

$$m_{22}(\mathbf{x}) \neq \frac{m_{12}(\mathbf{x})m_{23}(\mathbf{x})}{m_{13}(\mathbf{x})} \tag{2.31}$$

$$m_{33}(\mathbf{x}) \neq \frac{m_{23}(\mathbf{x})m_{31}(\mathbf{x})}{m_{21}(\mathbf{x})} \tag{2.32}$$

Proof. See Section 2.6.3. ■



(a) α_{213} and β_{213} (b) α_{312} and β_{312}

Figure 2.5: A graphical illustration of the four edges, α_{213}, β_{213}, α_{312}, and β_{312}, which are important in defining the networks that realize $\eta(\mathbf{x}) = 1$.

Eq. (2.30)-(2.32) can be reformulated into the following equivalent conditions:

$$p_i(\mathbf{x}) \neq 1 \quad \forall 1 \leq i \leq 3$$

2.5.2 Topological Interpretations of the Feasibility Conditions

As we have seen, the following coupling relations are important for the achievability of PBNA: 1) $\eta(\mathbf{x}) = 1$; 2) $p_i(\mathbf{x}) = 1$ and $p_i(\mathbf{x}) = \eta(\mathbf{x})$ where $i = 1, 2, 3$; 3) $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$, $p_i(\mathbf{x}) = 1 + \eta(\mathbf{x})$, where $i = 2, 3$. As we will see, the networks that realize these coupling relations have special topological properties. We defer all the proofs to Appendix C.

We assume that all the edges in E are arranged in a topological ordering such that if $\text{head}(e) = \text{tail}(e')$, e must precede e' in this ordering.

Definition 2.5.1. Given two subsets of edges S and D, we define an edge e as a *bottleneck* between S and D if the removal of e will disconnect every directed path from S to D.

Given $1 \leq i, j, k \leq 3$, let α_{ijk} denote the last bottleneck between σ_i and $\{\tau_j, \tau_k\}$ in this topological ordering, and β_{ijk} the first bottleneck between $\{\sigma_j, \alpha_{ijk}\}$ and τ_k.

As shown below, the four edges, α_{213}, β_{213}, α_{312}, and β_{312}, are important in defining the networks that realize $\eta(\mathbf{x}) = 1$. A graphical illustration of the four edges is shown in Fig. 2.5.

Theorem 2.5.3. $\eta(\mathbf{x}) = 1$ if and only if $\alpha_{213} = \alpha_{312}$ and $\beta_{213} = \beta_{312}$.

In [53], the authors independently discovered a similar result. Consider the example shown in Fig. 2.4a. It is easy to see that in this example, $\alpha_{213} = \alpha_{312} = \beta_{213} = \beta_{312} = e$, and thus $\eta(\mathbf{x}) = 1$. In Fig. 2.6a, we show another example, where $\alpha_{213} = \alpha_{312} = e_1$, $\beta_{213} = \beta_{312} = e_2$, and thus $\eta(\mathbf{x}) = 1$.

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 2.5.4. The following statements hold:

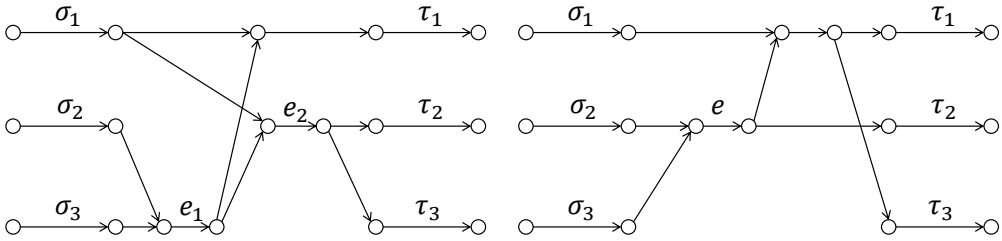
1. $p_1(\mathbf{x}) = 1$ if and only if the minimum cut between $\{\sigma_1, \sigma_2\}$ and $\{\tau_1, \tau_3\}$ equals one; $p_1(\mathbf{x}) = \eta(\mathbf{x})$ if and only if the minimum cut between $\{\sigma_1, \sigma_3\}$ and $\{\tau_1, \tau_2\}$ equals one.

2. $p_2(\mathbf{x}) = 1$ if and only if the minimum cut between $\{\sigma_1, \sigma_2\}$ and $\{\tau_2, \tau_3\}$ equals one; $p_2(\mathbf{x}) = \eta(\mathbf{x})$ if and only if the minimum cut between $\{\sigma_2, \sigma_3\}$ and $\{\tau_1, \tau_2\}$ equals one.

3. $p_3(\mathbf{x}) = 1$ if and only if the minimum cut between $\{\sigma_2, \sigma_3\}$ and $\{\tau_1, \tau_3\}$ equals one; $p_3(\mathbf{x}) = \eta(\mathbf{x})$ if and only if the minimum cut between $\{\sigma_1, \sigma_3\}$ and $\{\tau_2, \tau_3\}$ equals one.

For instance, in Fig. 2.4a, the cut-set with minimum capacity between $\{\sigma_2, \sigma_3\}$ and $\{\tau_1, \tau_2\}$ contains only one edge e, and thus $p_2(\mathbf{x}) = \eta(\mathbf{x})$.

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. As shown below, two edges are important in defining



(a) $\eta(\mathbf{x}) = 1$ (b) $p_2(\mathbf{x}) = \eta(\mathbf{x})$

Figure 2.6: Additional examples of coupling relations

the networks that realizes the third coupling relation in Eq. (2.27)-(2.29), *e.g.,* α_{213} and α_{312} are used to define the networks that realize $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$, and so on.

Theorem 2.5.5. The following statements hold:

1. $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$ if and only if the following conditions are satisfied: a) α_{312} is a bottleneck between σ_1 and τ_2; b) α_{213} is a bottleneck between σ_1 and τ_3; c) α_{312} is parallel with α_{213}; d) $\{\alpha_{312}, \alpha_{213}\}$ forms a cut-set between σ_1 from τ_1.

2. $p_2(\mathbf{x}) = 1 + \eta(\mathbf{x})$ if and only if the following conditions are satisfied: a) α_{123} is a bottleneck between σ_2 and τ_3; b) α_{321} is a bottleneck between σ_2 and τ_1; c) α_{123} is parallel with α_{321}; d) $\{\alpha_{123}, \alpha_{321}\}$ forms a cut-set between σ_2 from τ_2.

3. $p_3(\mathbf{x}) = 1 + \eta(\mathbf{x})$ if and only if the following conditions are satisfied: a) α_{231} is a bottleneck between σ_3 and τ_1; b) α_{132} is a bottleneck between σ_3 and τ_2; c) α_{231} is parallel with α_{132}; d) $\{\alpha_{231}, \alpha_{132}\}$ forms a cut-set between σ_3 from τ_3.

Consider the network as shown in Fig. 2.4b. It is easy to see that $e_2 = \alpha_{312}$ and $e_1 = \alpha_{213}$, and all the conditions in 1) of Theorem 2.5.5 are satisfied. Therefore, this network realizes the coupling relation $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$. Note that these three coupling relations are mutually exclusive when $\eta(\mathbf{x})$ is not constant. If any two of these coupling relation were to occur in the same network, then it would induce a graph structure that forces $\eta(\mathbf{x})$ to be a constant [53].

2.6 Achievability Conditions of PBNA

In this section, we first present two graph-related properties, namely Linearization Property and Square-Term Property, which play important roles in the proof of the sufficiency of the conditions of Theorem 2.5.1. Then, we explain the main ideas behind Theorem 2.5.1, and the proof of 2.5.2. Consistent with Section 2.5, we distinguish two cases based on whether $\eta(\mathbf{x})$ is constant. The full proof is provided in Appendix B.

2.6.1 Graph-Related Properties

Since the transfer functions are defined on graphs, they exhibit special graph-related properties introduced by the graph structure. In the following discussion, we consider the general form of $p_i(\mathbf{x})$ as below

$$h(\mathbf{x}) = \frac{m_{ab}(\mathbf{x})m_{pq}(\mathbf{x})}{m_{aq}(\mathbf{x})m_{pb}(\mathbf{x})} \tag{2.33}$$

where $a, b, p, q = 1, 2, 3$ and $a \neq p, b \neq q$. Moreover, by the definition of transfer function, the numerator and denominator of $h(\mathbf{x})$ can be expanded respectively as follows:

$$m_{ab}(\mathbf{x})m_{pq}(\mathbf{x}) = \sum\nolimits_{(P_1,P_2)\in\mathcal{P}_{ab}\times\mathcal{P}_{pq}} t_{P_1}(\mathbf{x})t_{P_2}(\mathbf{x})$$
$$m_{aq}(\mathbf{x})m_{pb}(\mathbf{x}) = \sum\nolimits_{(P_3,P_4)\in\mathcal{P}_{aq}\times\mathcal{P}_{pb}} t_{P_3}(\mathbf{x})t_{P_4}(\mathbf{x})$$

Hence, each path pair in $\mathcal{P}_{ab} \times \mathcal{P}_{pq}$ contributes a term in $m_{ab}(\mathbf{x})m_{pq}(\mathbf{x})$, and each path pair in $\mathcal{P}_{aq} \times \mathcal{P}_{pb}$ contributes a term in $m_{aq}(\mathbf{x})m_{pb}(\mathbf{x})$.

The first graph-related property, namely Linearization Property, is stated in the following lemma. According to this property, if $p_i(\mathbf{x})$ is not constant, it can be transformed into its simplest non-trivial form, i.e., a linear function or the inverse of a linear function, through a

partial assignment of values to $\mathbf{x}$.

Lemma 2.6.1 (Linearization Property)**.** Assume $h(\mathbf{x})$ is not constant. Let $h(\mathbf{x}) = \frac{u(\mathbf{x})}{v(\mathbf{x})}$ such that $\gcd(u(\mathbf{x}), v(\mathbf{x})) = 1$. Then, we can assign values to $\mathbf{x}$ other than a variable $x_{ee'}$ such that $u(\mathbf{x})$ and $v(\mathbf{x})$ are transformed into either $u(x_{ee'}) = c_1 x_{ee'} + c_0$, $v(x_{ee'}) = c_2$ or $u(x_{ee'}) = c_2, v(x_{ee'}) = c_1 x_{ee'} + c_0$, where c_0, c_1, c_2 are constants in $\mathbb{F}_{2^m}$, and $c_1 c_2 \neq 0$.

Proof. See Appendix A. ■

The second property, namely Square-Term Property, is presented in the following lemma. According to this property, the coefficient of $x_{ee'}^2$ in the numerator of $h(\mathbf{x})$ equals its counterpart in the denominator of $h(\mathbf{x})$. Thus, if $x_{ee'}^2$ appears in the numerator of $h(\mathbf{x})$ under some assignment to $\mathbf{x}$, it must also appear in the denominator of $h(\mathbf{x})$, and vice versa.

Lemma 2.6.2 (Square-Term Property)**.** Given a coding variable $x_{ee'}$, let $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ be the coefficients of $x_{ee'}^2$ in $m_{ab}(\mathbf{x})m_{pq}(\mathbf{x})$ and $m_{aq}(\mathbf{x})m_{pb}(\mathbf{x})$ respectively. Then $f_1(\mathbf{x}) = f_2(\mathbf{x})$.

Proof. See Appendix A. ■

2.6.2 $\eta(\mathbf{x})$ Is Not Constant

In this subsection, we first present a simple method to quickly identify a class of networks, for which PBNA can asymptotically achieve symmetric rate $\frac{1}{2}$. Then, we sketch the outline of the proof for the sufficiency of Theorem 2.5.1. Next, we explain the main idea behind the proof for the necessity of Theorem 2.5.1.

A Simple Method Based on Theorem 2.4.1

As shown in Theorem 2.4.1, the set $\mathcal{S}'$ contains an exponential number of rational functions, and thus it is very difficult to check the conditions of Theorem 2.4.1 in practice. Interestingly, the theorem directly yields a simple method to quickly identify a class of networks for which PBNA is feasible. The major idea of the method is to exploit the asymmetry between $p_i(\mathbf{x})$ and $\eta(\mathbf{x})$ in terms of effective variables. Here, given a rational function $f(\mathbf{y})$, we define a variable as an effective variable of $f(\mathbf{y})$ if it appears in the unique form of $f(\mathbf{y})$. Let $\mathcal{V}(f(\mathbf{y}))$ denote the set of effective variables of $f(\mathbf{y})$. Intuitively, this asymmetry allows us more freedom to control the values of $p_i(\mathbf{x})$ and $\eta(\mathbf{x})$ such that they can change independently, which makes the network behave more like a wireless channel. The formal description of the method is presented below:

Corollary 2.6.1. Assume all $m_{ij}(\mathbf{x})$'s $(i, j = 1, 2, 3)$ are non-zeros, and $\eta(\mathbf{x})$ is not constant. Each unicast session can asymptotically achieve one half rate through PBNA if for $i = 1, 2, 3$, $p_i(\mathbf{x}) \neq 1$ and $\mathcal{V}(\eta(\mathbf{x})) \neq \mathcal{V}(p_i(\mathbf{x}))$.

Proof. If the above conditions are satisfied, we must have $p_i(\mathbf{x}) \neq \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}'$. Thus, the theorem holds. ■

Consider the networks shown in Fig. 2.7a and Fig. 2.2, which we replicate in Fig. 2.7b for easy review. As shown in these examples, due to edge e, $\eta(\mathbf{x})$ contains effective variables $x_{\sigma_3 e}, x_{e\tau_2}$, which are absent in the unique form of $p_i(\mathbf{x})$ $(i = 1, 2, 3)$. Thus, by Corollary 2.6.1, each unicast session can asymptotically achieve one half rate through PBNA. However, Corollary 2.6.1 doesn't subsume all possible networks for which PBNA can achieve one half rate. For instance, in Fig. 2.7c, we show a counter example, where $\mathcal{V}(\eta(\mathbf{x})) = \mathcal{V}(p_1(\mathbf{x}))$, and thus Corollary 2.6.1 is not applicable. Nevertheless, it is easy to verify the network satisfies the conditions of Theorem 2.5.1, and thus PBNA can still achieve one half rate.

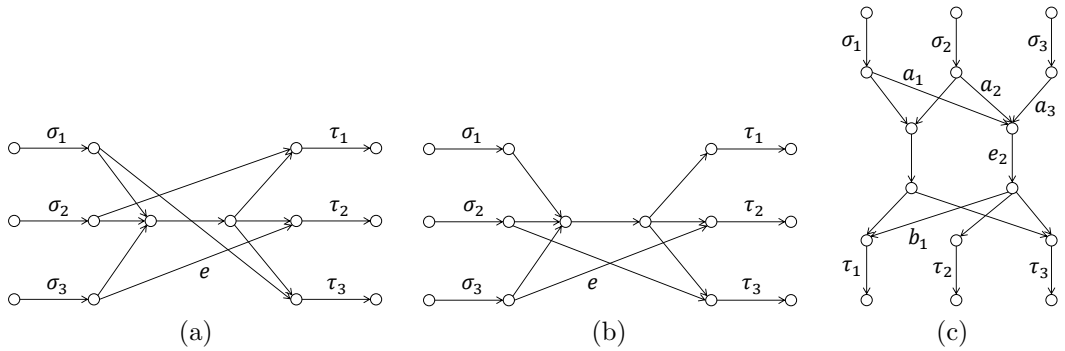


Figure 2.7: Illustration of type III networks. (i) It can be seen that for all the three examples, PBNA can achieve one half rate. (ii) The three examples can be verified by using different methods: for (a) and (b), due to edge e, $\eta(\mathbf{x})$ contains coding variables $x_{\sigma_3 e}, x_{e\tau_2}$, which are absent in the unique forms of $p_1(\mathbf{x}), p_2(\mathbf{x})$ and $p_3(\mathbf{x})$, and thus Corollary 2.6.1 applies to both cases; Corollary 2.6.1 doesn't apply to (c), but PBNA can still achieve a symmetric rate $\frac{1}{2}$ for this network according to Theorem 2.5.1. (iii) For both (a) and (b), routing can only achieve a symmetrical rate $\frac{1}{3}$; for (c), PBNA and routing can both achieve a symmetrical rate $\frac{1}{2}$.

Sufficiency of Theorem 2.5.1

As shown in Section 2.4, not all coupling relations $p_i(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}'$ are realizable due to the special properties of transfer functions. Indeed, since the transfer functions are defined on graphs, they exhibit special properties due to the graph structure. As we will see, these properties are essential in identifying the minimal sub-set of realizable coupling relations. In fact, we only need two such properties, namely Linearization Property and Square-Term Property, which are presented below.

Now, we sketch the outline for the proof of the sufficiency of Theorem 2.5.1. The proof consists of three steps:

First, we use the Linearization Property and a simple degree-counting technique to reduce

$\mathcal{S}'$ to the following set $\mathcal{S}''_1$:

$$\mathcal{S}''_1 = \left\{ \frac{a_0 + a_1\eta(\mathbf{x})}{b_0 + b_1\eta(\mathbf{x})} \in \mathcal{S}' : a_0, a_1, b_0, b_1 \in \mathbb{F}_q \right\} \tag{2.34}$$

Next, we iterate through all possible configurations of a_0, a_1, b_0, b_1, and utilize the Linearization Property and the Square-Term Property to further reduce $\mathcal{S}''_1$ to just four rational functions:

$$\mathcal{S}''_2 = \left\{ 1, \eta(\mathbf{x}), 1 + \eta(\mathbf{x}), \frac{\eta(\mathbf{x})}{1 + \eta(\mathbf{x})} \right\} \tag{2.35}$$

Finally, we use a recent result from [53] to rule out the fourth redundant rational function in $\mathcal{S}''_2$, resulting in the minimal set $\mathcal{S}'_i$ defined in Theorem 2.5.1. The detailed proof is deferred to Appendix B.

Necessity of the Theorem 2.5.1

We first show how to get a precoding matrix $\mathbf{V}_1$ that satisfies Eq. (2.12). The construction of $\mathbf{V}_1$ involves solving a system of linear equations defined on $\mathbb{F}_{2^m}(\xi)(z)$:

$$\mathbf{r}(z)(z\mathbf{C} - \mathbf{BA}) = 0 \tag{2.36}$$

In the above equation, $\mathbf{r}(z) = (r_1(z), \cdots, r_{n+s}(z))$, where $r_i(z) \in \mathbb{F}_{2^m}(\xi)(z)$ for $1 \leq i \leq n+s$. Assume $\mathbf{r}_0(z)$ is a non-zero solution to Eq. (2.36). Substitute z with $\eta(\mathbf{x})$, and we have $\eta(\mathbf{x})\mathbf{r}_0(\eta(\mathbf{x}))\mathbf{C} = \mathbf{r}_0(\eta(\mathbf{x}))\mathbf{BA}$. Finally, construct the following precoding matrix

$$\mathbf{V}_1^T = (\mathbf{r}_0^T(\eta(\mathbf{x}^{(1)})) \quad \mathbf{r}_0^T(\eta(\mathbf{x}^{(2)})) \quad \cdots \quad \mathbf{r}_0^T(\eta(\mathbf{x}^{(2n+s)}))) \tag{2.37}$$

Apparently, $\mathbf{V}_1$ satisfies Eq. (2.10). Hence, each non-zero solution to Eq. (2.36) corresponds to a row of $\mathbf{V}_1$ satisfying Eq. (2.10). Conversely, it is straightforward to see that each row of $\mathbf{V}_1$ satisfying Eq. (2.10) corresponds to a solution to Eq. (2.36).

As we will prove in Appendix B, $\text{rank}(z\mathbf{C} - \mathbf{BA}) = n$. If $s = 0$, $z\mathbf{C} - \mathbf{BA}$ becomes an invertible square matrix, and Eq. (2.36) only has zero solution. Thus, in order for Eq. (2.12) to have a non-zero solution, s must equal 1.

As an example, consider the case where $s = 1$, $n = 2$, and $2^m = 4$. Let α be the primitive element of $\mathbb{F}_4$ such that $\alpha^3 = 1$ and $\alpha^2 + \alpha + 1 = 0$. Moreover, let $\mathbf{A} = \mathbf{I}_2$ and

$$\mathbf{C} = \begin{pmatrix} 1 & \alpha \\ \alpha & 1 \\ \alpha^2 & 1 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} \alpha^2 & \alpha \\ 1 & 1 \\ 1 & \alpha \end{pmatrix}$$

It's easy to verify that $\mathbf{r}(z) = (\alpha^2 z^2 + \alpha, z + \alpha, z^2 + \alpha z + \alpha^2)$ satisfies Eq. (2.36). Thus, we substitute z with $\eta(\mathbf{x}^j)$ and construct $\mathbf{V}_1^T = (\mathbf{r}^T(\eta(\mathbf{x}^1)) \;\; \mathbf{r}^T(\eta(\mathbf{x}^2)) \;\; \cdots \;\; \mathbf{r}^T(\eta(\mathbf{x}^5)))$. Apparently, Eq. (2.10) is satisfied. From this example, we can see that given different $\mathbf{A}, \mathbf{B}, \mathbf{C}$, we can construct different precoding matrix $\mathbf{V}_1$, and thus the choices of precoding matrices are not limited to those defined in Eq. (2.12)-(2.14). An interesting observation is that the above precoding matrix $\mathbf{V}_1$ is closely related to Eq. (2.12) through a transform equation: $\mathbf{V}_1 = \mathbf{V}_1^* \mathbf{F}$, where

$$\mathbf{F} = \begin{pmatrix} \alpha & \alpha & \alpha^2 \\ 0 & 1 & \alpha \\ \alpha^2 & 0 & 1 \end{pmatrix}$$

Actually, this observation can be generalized to the following Lemma.

Lemma 2.6.3. Assume $s = 1$. Any $\mathbf{V}_1$ satisfying Eq. (2.10) is related to $\mathbf{V}_1^*$ through the

following transform equation

$$\mathbf{V}_1 = \mathbf{G}\mathbf{V}_1^*\mathbf{F} \tag{2.38}$$

where $\mathbf{V}_1^*$ is defined in Eq. (2.12), $\mathbf{F}$ is an $(n+1)\times(n+1)$ matrix, and $\mathbf{G}$ is a $(2n+1)\times(2n+1)$ diagonal matrix, with the (i,i) element being $f_i(\eta(\mathbf{x}^i))$, where $f_i(z)$ is an arbitrary non-zero rational function in $\mathbb{F}_{2^m}(\xi)(z)$. Moreover, the $(n+1)$th row of $\mathbf{FC}$ and the 1st row of $\mathbf{FBA}$ are both zero vectors.

Proof. See Appendix B. ■

Using Lemma 2.6.3, we can prove that if a coupling relation $p_i(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}'$ is present in the network, any PBNA cannot achieve one half rate per unicast session. This implies that the conditions of Theorem 2.5.1 are also necessary for PBNA to achieve one half rate per unicast session. We defer the detailed proof to Appendix B.

2.6.3 $\eta(\mathbf{x})$ Is Constant

Proof of Theorem 2.5.2. In the proof of Theorem 2.4.2, we've proved the sufficiency of Theorem 2.5.2. If $p_i(\mathbf{x}) = 1$, $\mathbf{P}_i$ becomes an identity matrix. We will show that it is impossible for PBNA to achieve one half rate for each unicast session. We only prove the case for $i = 1$. The other cases $i = 2, 3$ can be proved similarly, and are omitted. The matrix in the reformulated rank condition $\mathcal{B}_1'$ becomes $(\mathbf{V}_1 \quad \mathbf{V}_1\mathbf{C})$. Since $\text{rank}(\mathbf{V}_1\mathbf{C}) = n$, there are n columns in $\mathbf{V}_1$ that are linearly dependent of the columns in $\mathbf{V}_1\mathbf{C}$. Thus, it is impossible for PBNA to achieve one half rate for ω_1. ■

In Fig. 2.8, we show an example of this case. Note that the network in Fig. 2.8 has rich connectivity such that each sender is connected to its corresponding receiver via a disjoint

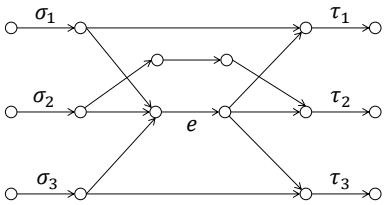


Figure 2.8: An example where $\eta(\mathbf{x}) = 1$ and $p_i(\mathbf{x}) \neq 1$ for $i \in \{1, 2, 3\}$, and thus each unicast session can achieve one half rate in exactly two time slots due to Theorem 2.5.2. For this example, routing achieves symmetric rate of one.

directed path. Thus, there is no coding opportunity that can be exploited, and routing is sufficient to achieve rate 1 per unicast session, which is the maximum symmetric rate achieved by any network coding schemes. Hence, this class of networks is of less significance than the class of networks considered in Theorem 2.5.1.

2.6.4 Some s_i Is Disconnected from Some d_j $(i \neq j)$

In this case, since the number of interfering signals is reduced, at least one alignment condition can be removed, and thus the restriction on $\mathbf{V}_1$ imposed by Eq. (2.10) vanishes. Therefore, we can choose $\mathbf{V}_1$ freely, and the feasibility conditions of PBNA can be greatly simplified. For example, assume $m_{21}(\mathbf{x}) = 0$ and all other transfer functions are non-zeros. Hence, the alignment condition for the first unicast session vanishes. Using a scheme similar to above, we set $\mathbf{V}_1 = (\theta_1 \quad \theta_2)^T$, $\mathbf{V}_2 = \mathbf{M}_{13}\mathbf{M}_{23}^{-1}(\theta_1 \quad \theta_2)^T$ and $\mathbf{V}_3 = \mathbf{M}_{12}\mathbf{M}_{32}^{-1}(\theta_1 \quad \theta_2)^T$, and thus the interferences at τ_2 and τ_3 are all perfectly aligned. It is easy to see that $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ is feasible through PBNA if and only if $p_i(\mathbf{x})$ is not constant for every $i = 1, 2, 3$. Using similar arguments, we can discuss other cases.

2.7 Summary

In this chapter, we consider the problem of network coding for the SISO scenarios with three unicast sessions. We consider a network model, in which the middle of the network performs random linear network coding. We apply precoding-based interference alignment [30] to this network setting. We show that network topology may introduce algebraic dependence ("coupling relations") between different transfer functions, which can potentially affect the rate achieved by PBNA. Using two graph-related properties and a recent result from [53], we identify the minimal set of coupling relations that are realizable in networks. Moreover, we show that each of these coupling relations has a unique interpretation in terms of network topology. Based on these interpretations, we present a polynomial-time algorithm to check the existence of these coupling relations.

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.

There are still many problems that remain to be solved regarding applying interference alignment techniques to the 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. The extensions to other network scenarios beyond SISO with more than three unicast sessions are highly non-trivial. Finally, the current paper applies precoding at the sources only, while intermediate nodes performed simply random network coding; an open direction for future work is alignment by network

code design in the middle of the network as well.

Chapter 3

Multicast-Packing Coding Scheme for Multiple Unicast Sessions

3.1 Introduction

In this chapter, we focus on network coding across multiple unicast sessions over linear networks, where the nodes in the network can perform linear network coding operations. It has been shown that determining whether there exists a linear network coding scheme for multiple unicasts is NP-hard [19]. Thus, constructive and sub-optimal approaches, such as [21–24, 27], have been proposed and shown to improve over routing.

In this chapter, we introduce a constructive inter-session network coding scheme for multiple unicast sessions, illustrated in the following example.

Example 3.1.1. Let us consider the network $\mathcal{N}$ shown in Fig. 3.1a. In this network, five unicast sessions coexist in the network, where each edge has unit capacity. The ith unicast session ($1 \leq i \leq 5$) is denoted by $\omega_i = (s_i, d_i)$, where s_i and d_i are the sender and the receiver of ω_i, respectively. The set of unicast sessions are represented by $\Omega = \{\omega_i : 1 \leq i \leq 5\}$.

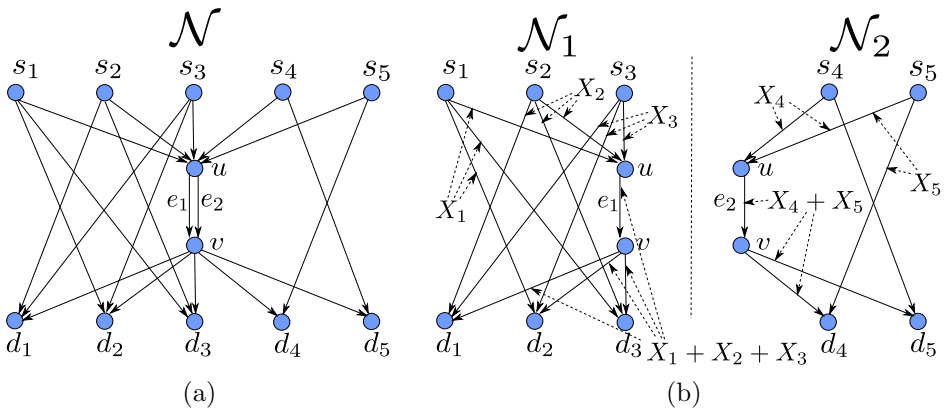


Figure 3.1: A motivating example. In the network $\mathcal{N}$ shown in (a), five unicast sessions coexist. In (b), these unicast sessions are partitioned into two disjoint subsets, $\Omega_1 = \{\omega_1, \omega_2, \omega_3\}$ and $\Omega_2 = \{\omega_4, \omega_5\}$, and the network $\mathcal{N}$ is partitioned into two sub-graphs $\mathcal{N}_1$ and $\mathcal{N}_2$. Note that the unicast sessions in Ω_1 use only $\mathcal{N}_1$, while the unicast sessions in Ω_2 use only $\mathcal{N}_2$. Then, we construct linear network coding schemes for Ω_1 and Ω_2 separately, as shown in (b), where X_i $(1 \leq i \leq 5)$ denotes the source symbol transmitted by s_i. Note that the constructed network coding schemes are network coding schemes for two multicast scenarios over $\mathcal{N}_1$ and $\mathcal{N}_2$.

Let us partition Ω into two disjoint sets, $\Omega_1 = \{\omega_1, \omega_2, \omega_3\}$ and $\Omega_2 = \{\omega_4, \omega_5\}$. Also, $\mathcal{N}$ is partitioned into two sub-graphs $\mathcal{N}_1$ and $\mathcal{N}_2$. The unicast sessions Ω_1 and Ω_2 use only their respective subgraphs to transmit symbols, *i.e.,* the unicast sessions in Ω_1 (Ω_2) uses only $\mathcal{N}_1$ ($\mathcal{N}_2$). Then, we construct our codes to be the network codes for two multicast scenarios: In $\mathcal{N}_1$, d_1, d_2 and d_3 can decode all the source symbols transmitted by s_1, s_2 and s_3; in $\mathcal{N}_2$, d_4 and d_5 can decode all the source symbols transmitted by s_4 and s_5. These network codes also serve as network codes for the original multiple unicast sessions.

Note that several partitions of Ω, other than Ω_1 and Ω_2 discussed in this example, are also possible. Part of the contribution of this paper is how to find good partitions. ■

The example demonstrates the approach we follow in this paper. First, we partition the set of unicast flows into disjoint subsets of unicast flows. Second, we map each subset of unicast flows to a multicast session with the same set of receivers, and linear network codes are constructed for these multicast sessions by a deterministic [9] or a random approach [10].

These linear network codes collectively serve as a linear network coding scheme for the original unicast sessions, which we refer to as *Multicast-Packing Coding Scheme* or MPC for short.

MPC has the following strengths. First, the MPC approach, *i.e.,* partitioning the unicast sessions to subsets of unicast sessions and mapping each subset to a multicast network coding problem, is general enough to be applied to any directed acyclic graph. Second, given a partition of the set of the unicast sessions, we use a linear program to quickly analyze the performance, *e.g.,* maximum common rate and minimum cost, achieved by MPC. In contrast, previous constructive approaches are difficult to analyze due to the lack of succinct mathematical formulations. For example, the integer linear programming (ILP) approach [21] is difficult to analyze since it needs to consider all possible butterfly structures in the network. On the other hand, the evolutionary approach [27], does not have a mathematical formulation. Third, in order to find the best MPC, we only need to search the space of all partitions of the set of unicast sessions, *independently of the network size.* This is clearly more efficient and scalable than other constructive approaches, whose combinatorial optimization involved the network graph in addition to the set of sessions. For example, the approach in [21] uses integer linear programming to select the best set of butterflies considering all pairs of unicast sessions, but also all possible coding points on the network topology. The evolutionary approach [27] involves a random walk in the space of local coding vectors, which does not scales well with the network size. Although independent of the network size, our search problem is still exponential in the number of unicast sessions. This is why we utilize a suboptimal, yet efficient, simulating annealing technique to find good partitions of the unicast sessions. Simulation results over appropriately chosen scenarios demonstrate the above points.

The rest of this chapter is organized as follows. In Section 3.2, we present problem setup. In Section 3.3, we present the formal definition of MPC, and the rate region achieved by MPC.

In Section 3.4, we present a simulated annealing algorithm to find good partitions for MPC. In Section 3.5, we present the evaluation of MPC. Section 3.6 concludes this chapter.

3.2 Problem Setup

3.2.1 Network Model

A network is represented by a weighted directed acyclic graph $\mathcal{N} = (V, E, h)$, where V and E denote the node set and the edge set, respectively, and $h : E \to \mathbb{R}_{\geq 0}$ is a function such that for $e \in E$, $h(e)$ equals the capacity of e. We allow multiple edges between two nodes, and hence $E \subseteq V \times V \times \mathbb{Z}_+$, where the last integer enumerates edges between two nodes. The edges are denoted by (u, v, i). If no confusion arises, we simply use (u, v) to represent edges. We denote the tail and the head of an edge e by $\text{head}(e)$ and $\text{tail}(e)$, respectively. The sets of incoming and outgoing edges at a node v are denoted by $\text{In}(v)$ and $\text{Out}(v)$ respectively, *i.e.,* $\text{In}(v) = \{e \in E : \text{head}(e) = v\}$ and $\text{Out}(v) = \{e \in E : \text{tail}(e) = v\}$. There are multiple unicast sessions in the network. We use a set $\Omega = \{\omega_i = (s_i, d_i) : 1 \leq i \leq |\Omega|\}$ to represent the multiple unicast sessions, where ω_i denotes the ith unicast session, and s_i, d_i are the sender and the receiver of ω_i respectively. Given $\Omega' \subseteq \Omega$, $S(\Omega')$ and $D(\Omega')$ denote the set of senders and the set of receivers involved in Ω', respectively. For $1 \leq i \leq |\Omega|$, $\mathbf{X}_i$ denotes the vector of source symbols that s_i transmits to d_i. Given a vector $\mathbf{A}$, $|\mathbf{A}|$ denotes the dimension of $\mathbf{A}$.

We make the following assumptions to simplify our analysis.

- The symbols transmitted in the network all belong to a finite field $\mathbb{F}_q$.
- The source symbols transmitted by all the senders in $S(\Omega)$ are mutually independent random variables uniformly distributed over $\mathbb{F}_q$.

- Each edge in the network represents an error-free and delay-free channel.
- The senders are all different nodes, and so are the receivers.
- $\text{In}(s_i) = \text{Out}(d_i) = \emptyset$, for $1 \leq i \leq |\Omega|$.

3.2.2 Linear Network Coding Scheme

We define a linear network coding scheme as follows.

Definition 3.2.1. Let t be a positive integer such that $\lceil t \times h(e) \rceil \geq 1$ for each edge $e \in E$ with positive capacity. Denote $k(e) = \lceil t \times h(e) \rceil$. A *linear network coding scheme* of length t for the multiple unicast sessions Ω consists of the following components:

1. For each sender $s_i \in S(\Omega)$ and each $e \in \text{Out}(s_i)$ with positive capacity, a $k(e) \times |\mathbf{X}_i|$ encoding matrix $\mathbf{E}_e$ over $\mathbb{F}_q$.
2. For each $v \in V - (S(\Omega) \cup D(\Omega))$ and each $e \in \text{Out}(v)$ with positive capacity, a $k(e) \times (\sum_{e' \in \text{In}(v)} k(e'))$ encoding matrix $\mathbf{E}_e$ over $\mathbb{F}_q$.
3. For each $1 \leq i \leq |\Omega|$, an $|\mathbf{X}_i| \times (\sum_{e' \in \text{In}(v)} k(e'))$ decoding matrix $\mathbf{D}_i$.

We use a tuple $\lambda = (\mathbf{E}_e : e \in E, h(e) > 0;\ \mathbf{D}_i : 1 \leq i \leq |\Omega|)$ to represent the above linear network coding scheme.

In a linear network coding scheme, for an edge e with positive capacity, the vector of the symbols transmitted along e, denoted by $\mathbf{Y}_e$, is a function of $(\mathbf{X}_i : 1 \leq i \leq |\Omega|)$ defined recursively as follows:

$$\mathbf{Y}_e = \begin{cases} \mathbf{E}_e \mathbf{X}_i & \text{if } e \in \text{Out}(s_i); \\ \mathbf{E}_e(\mathbf{Y}_{e'} : e' \in \text{In}(v), h(e') > 0) & \text{if } e \in \text{Out}(v), v \in V - (S(\Omega) \cup D(\Omega)). \end{cases}$$

Definition 3.2.2. Given a rate vector $\mathbf{R} = (R_i : 1 \leq i \leq |\Omega|) \in \mathbb{R}_{\geq 0}^{|\Omega|}$, we say that $\mathbf{R}$ is *achievable* by linear network coding schemes if for any $\epsilon \in \mathbb{R}_{>0}$, there exists a linear network coding scheme $\lambda = (\mathbf{E}_e : e \in E, h(e) > 0;\ \mathbf{D}_i : 1 \leq i \leq |\Omega|)$ of length t such that the following conditions are satisfied:

1. For each $1 \leq i \leq |\Omega|$, $\mathbf{X}_i = \mathbf{D}_i(\mathbf{Y}_e : e \in \mathrm{In}(d_i), h(e) > 0)$.
2. For each $1 \leq i \leq |\Omega|$, $\frac{|\mathbf{X}_i|}{t} > R_i - \epsilon$.

The *rate region* achieved by linear network coding schemes, denoted by $\mathcal{R}_{lnc}$, is the set of the rate vectors $\mathbf{R}$'s achievable by linear network coding schemes.

3.3 Packing Multicast for Multiple Unicast Sessions

3.3.1 Multicast-Packing Coding Scheme (MPC)

In this chapter, we present the detailed description of MPC, *i.e.,* mapping multiple unicast sessions to multicast sessions, when the partition of the original multiple unicast sessions is given. The problem of how to finding such a partition is considered in Section 3.4. We use a tuple (s, D), where $s \in V$ and $D \subseteq V - \{s\}$, to represent a multicast session such that the nodes in D all require the source symbols transmitted by s. A *multicast* scenario is represented by a set of multicast sessions, *i.e.,* $\Gamma = \{(s_i, D) : 1 \leq i \leq |\Gamma|\}$, where the nodes in D require all the source symbols transmitted by all s_i's.

Definition 3.3.1. A *partition* of the multiple-unicast scenario Ω is a set of non-empty disjoint subsets of Ω, $\mathcal{G} = \{\Omega_i : 1 \leq i \leq |\mathcal{G}|\}$, such that $\Omega = \bigcup_{i=1}^{|\mathcal{G}|} \Omega_i$.

Definition 3.3.2. Given a partition $\mathcal{G}$, an *allocation of network capacities* with respect to $\mathcal{G}$ is represented by a set of functions $\mathcal{H} = \{h_i : E \to \mathbb{R}_{\geq 0} : 1 \leq i \leq |\mathcal{G}|\}$, which satisfies the

following condition:

$$\sum_{i=1}^{|\mathcal{G}|} h_i(e) \leq h(e) \quad \forall e \in E. \tag{3.1}$$

Given $\mathcal{H}$, we define $\mathcal{N}_i = (V, E, h_i)$ as a sub-capacitated network, the edge capacities of which are defined by h_i. Given $\mathcal{G}$ and $\mathcal{H}$, we can view each Ω_i as a multiple-unicast scenario of smaller scale that works "separately" in the sub-capacitated network $\mathcal{N}_i$.

Example 3.3.1. For example, in Fig. 3.1, the allocation of network capacities are as follows. If e is an outgoing edge of s_1, s_2, s_3, an incoming edge of d_1, d_2, d_3, or $e = e_1$, $h_1(e) = 1$; otherwise, $h_1(e) = 0$. If e is an outgoing edge of s_4, s_5, an incoming edge of d_4, d_5, or $e = e_2$, $h_2(e) = 1$; otherwise, $h_2(e) = 0$. ■

Suppose $\mathcal{G}$ and $\mathcal{H}$ are already given. We construct a linear network coding scheme for Ω as follows. For each $\Omega_i \in \mathcal{G}$, we construct a multicast scenario, $\Gamma_i = \{(s_j, D(\Omega_i)) : s_j \in S(\Omega_i)\}$, over the network $\mathcal{N}_i = (V, E, h_i)$, such that the receivers in $D(\Omega_i)$ can decode the source symbols transmitted by all the senders in $S(\Omega_i)$. A linear network coding scheme can then be constructed for this multicast scenario. These linear network coding schemes collectively serve as a linear network coding scheme, namely a multicast-packing coding scheme, for the original multiple unicast sessions Ω. More formally, we define a multicast-packing coding scheme as follows:

Definition 3.3.3. Suppose $\mathcal{G}$ is a partition of Ω, and $\mathcal{H}$ an allocation of network capacities with respect to $\mathcal{G}$. Let t be a positive integer such that for $1 \leq i \leq |\Omega|$ and $e \in \{e' \in E : h_i(e') > 0\}$, $\lceil t \times h_i(e) \rceil \geq 1$. Denote $k_i(e) = t \times h_i(e)$. A multicast-packing coding scheme (or a MPC for short) of length t with respect to $(\mathcal{G}, \mathcal{H})$ consists of the following components:

1. For each $1 \leq i \leq |\mathcal{G}|$, $s_j \in S(\Omega_i)$, and $e \in \mathrm{Out}(s_j)$ such that $h_i(e) > 0$, a $k_i(e) \times |\mathbf{X}_j|$ encoding matrix $\mathbf{E}_{i,e}$;

2. For each $1 \leq i \leq |\mathcal{G}|$, $v \in V - (S(\Omega) - D(\Omega))$ and $e \in \mathrm{Out}(v)$ such that $h_i(e) > 0$, a $k_i(e) \times (\sum_{e' \in \mathrm{In}(v)} k_i(e'))$ encoding matrix $\mathbf{E}_{i,e}$;

3. For each $1 \leq i \leq |\mathcal{G}|$ and $\omega_j \in \Omega_i$, an $|\mathbf{X}_j| \times (\sum_{e' \in \mathrm{In}(d_j)} k_i(e'))$ decoding matrix $\mathbf{D}_i$.

We use a tuple $\gamma = (\mathbf{E}_{i,e}, \mathbf{D}_i : 1 \leq i \leq |\mathcal{G}|, e \in E, h_i(e) > 0)$ to denote the above MPC.

In the MPC defined above, for $1 \leq i \leq |\mathcal{G}|$, and an edge $e \in \{e' \in E : h_i(e') > 0\}$, the vector of the symbols transmitted along e for the unicast sessions in Ω_i, denoted by $\mathbf{Y}_{i,e}$, is a function of $(\mathbf{X}_j : \omega_j \in \Omega_i)$ defined recursively as follows:

$$\mathbf{Y}_{i,e} = \begin{cases} \mathbf{E}_{i,e}\mathbf{X}_j & \text{if } e \in \mathrm{Out}(s_j), s_j \in S(\Omega_i); \\ \mathbf{E}_{i,e}(\mathbf{Y}_{i,e'} : e' \in \mathrm{In}(v), h_i(e') > 0) & \text{if } e \in \mathrm{Out}(v), v \in V - (S(\Omega) \cup D(\Omega)). \end{cases}$$

Definition 3.3.4. Given a rate vector $\mathbf{R} = (R_i : 1 \leq i \leq |\Omega|) \in \mathbb{R}_{\geq 0}^{|\Omega|}$, we say that $\mathbf{R}$ is achievable by MPC if for any $\epsilon \in \mathbb{R}_{>0}$, there exists a partition $\mathcal{G}$ of Ω, an allocation $\mathcal{H}$ of network capacities with respect to $\mathcal{G}$, and an MPC $\gamma = (\mathbf{E}_{i,e}, \mathbf{D}_i : 1 \leq i \leq |\mathcal{G}|, e \in E, h_i(e) > 0)$ of length t with respect to $(\mathcal{G}, \mathcal{H})$ such that the following conditions are satisfied:

1. For $1 \leq i \leq |\mathcal{G}|$, $(\mathbf{X}_j : \omega_j \in \Omega_i) \subseteq \mathrm{span}(\mathbf{Y}_{i,e} : e \in \mathrm{In}(d_i), h_i(e) > 0)$, and $\mathbf{X}_j = \mathbf{D}_j(\mathbf{Y}_{i,e} : e \in \mathrm{In}(d_i), h_i(e) > 0)$ for each $\omega_j \in \mathcal{G}$.

2. For $1 \leq j \leq |\Omega|$, $\frac{|\mathbf{X}_j|}{t} > R_i - \epsilon$.

The region achieved by MPC, denoted by $\mathcal{R}_{mpc}$, is the set of all the rate vectors achieved by MPC.

Remark. We can use the following method to construct an MPC. We add a super sender s and connect it to each $s_j \in S(\Omega_i)$ via $|\mathbf{X}_j|$ parallel edges, each of which has unit capacity

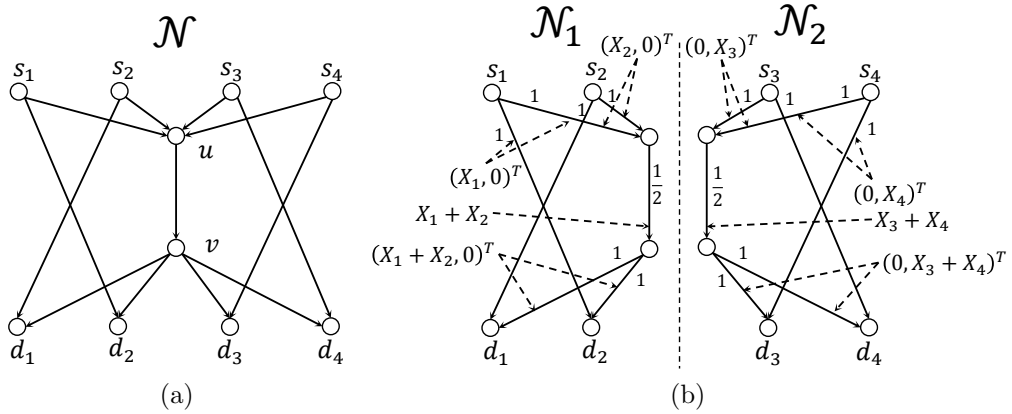


Figure 3.2: An example of MPC. The network is shown in (a), where each edge has unit capacity, and 4 unicast sessions coexist in the network. In (b), we show the two sub-capacitated networks $\mathcal{N}_1$ and $\mathcal{N}_2$ for a partition $\mathcal{G} = \{\Omega_1 = \{\omega_1, \omega_2\}, \Omega_2 = \{\omega_3, \omega_4\}\}$, where the numbers beside the edges marks an allocation of network capacities. In (b), we also show an MPC of length 2, which achieves one half rate for each unicast session.

and carries a distinct source symbol in $\mathbf{X}_j$. Thus, we transform the multicast scenario with multiple multicast sessions into a multicast scenario with a single multicast session. Hence, we can employ the random approach of [10] or the deterministic approach of [9] to construct linear network coding scheme for this multicast scenario.

Example 3.3.2. Consider the example network as shown in Fig. 3.2a. In this example, each edge has unit capacity, and four unicast sessions coexist in the network, *i.e.,* $\Omega = \{\omega_i = (s_i, d_i) : 1 \leq i \leq 4\}$. Each sender s_i sends only one source symbol X_i to d_i, *i.e.,* $\mathbf{X}_i$ is a scalar. We consider a partition $\mathcal{G} = \{\Omega_1 = \{\omega_1, \omega_2\}, \Omega_2 = \{\omega_3, \omega_4\}\}$. The allocation of network capacities is as follows. If e is an outgoing edge of s_1, s_2 or an incoming edge of d_1, d_2, $h_1(e) = 1$; if $e = (u, v)$, $h_1(e) = \frac{1}{2}$; otherwise, $h_1(e) = 0$. If e is an outgoing edge of s_3, s_4 or an incoming edge of d_3, d_4, $h_2(e) = 1$; if $e = (u, v)$, $h_2(e) = \frac{1}{2}$; otherwise, $h_2(e) = 0$. In Fig. 3.2b, we depict the sub-capacitated networks $\mathcal{N}_1$ and $\mathcal{N}_2$, where we overlook all the edges with zero capacities. We also show an MPC of length 2 in Fig. 3.2b. For example,

the encoding-matrices and the decoding matrices for Ω_1 are as follows:

$$\mathbf{E}_{1,(s_1,d_2)} = \mathbf{E}_{1,(s_1,u)} = (1 \quad 0)^T \quad \mathbf{E}_{1,(s_2,d_1)} = \mathbf{E}_{1,(s_2,u)} = (1 \quad 0)^T$$
$$\mathbf{E}_{1,(u,v)} = (1 \quad 0 \quad 1 \quad 0) \quad \mathbf{E}_{1,(v,d_1)} = \mathbf{E}_{1,(v,d_2)} = (1 \quad 0)^T$$
$$\mathbf{D}_1 = \mathbf{D}_2 = (1 \quad 0 \quad 1 \quad 0)$$

Note that both d_1 and d_2 can decode the two source symbols X_1 and X_2. Similarly, we can write down the encoding matrices and the decoding matrices for Ω_2. Hence, the linear network coding scheme is an MPC. Clearly, this MPC achieves one half rate for each unicast session. ■

Proposition 3.3.1. The MPC as shown in Fig. 3.1b and Fig. 3.2b achieve the maximal symmetrical rate achieved by any linear/nonlinear network coding schemes.

Proof. See Appendix E.1. ■

The choice of $\mathcal{G}$ and $\mathcal{H}$ is subject to various practical goals, which we explain in detail in Section 3.3.3.

3.3.2 Achievability of Multicast-Packing Code

In this section, we characterize the rate region of MPC for a given partition of the unicast sessions. We first introduce the following concept. Given $S \subset V$ and $d \in V - S$, an $S - d$ flow over $\mathcal{N}$ is a function $f : E \to \mathbb{R}_{\geq 0}$ which satisfies the following conditions:

1. For each edge $e \in E$, $0 \leq f(e) \leq h(e)$.

2. For each node $v \in V - (S \cup \{d\})$, the following flow conservation law must be satisfied:

$$\sum\nolimits_{e \in \mathrm{In}(v)} f(e) = \sum\nolimits_{e \in \mathrm{Out}(v)} f(e).$$

The value of flow f at $v \in S$ is defined as $\mathrm{val}(f, v) = \sum_{e \in \mathrm{Out}(v)} f(e) - \sum_{e \in \mathrm{In}(v)} f(e)$. A $S - d$ cut is a partition (W, U) of V such that $S \subseteq W$, $d \in U$, and $W \cup U = V$. The cut-set of cut (W, U) is defined as $C(W, U) = \{e \in E : \mathrm{tail}(e) \in W, \mathrm{head}(e) \in U\}$. The capacity of cut (W, U) is defined as the capacity of its cut-set. Let $\mathrm{mincut}(S, d, \mathcal{N})$ denote the minimum capacity of all $S - d$ cuts over $\mathcal{N}$.

The following theorem fully characterizes the rate region achieved by MPCs with respect to $(\mathcal{G}, \mathcal{H})$.

Theorem 3.3.1. Assume the size of finite field $\mathbb{F}_q$ is greater than $|\Omega|$. Let $\mathbf{R} = (R_1, \cdots, R_{|\Omega|}) \in \mathbb{R}_{\geq 0}^{|\Omega|}$. Then the following statements are equivalent:

1. $\mathbf{R}$ is achievable through MPC with respect to $(\mathcal{G}, \mathcal{H})$.
2. For each $\Omega_i \in \mathcal{G}$ and each $d_j \in D(\Omega_i)$, there exists a $S(\Omega_i) - d_j$ flow f_{ij} over $\mathcal{N}_i = (V, E, h_i)$ such that $\mathrm{val}(f_{ij}, s_l) = R_l$ for each $s_l \in S(\Omega_i)$.
3. For each $\Omega_i \in \mathcal{G}$ and each $d_j \in D(\Omega_i)$, the following condition is satisfied:

$$\sum_{s_l \in U} R_l \leq \mathrm{mincut}(U, d_j, \mathcal{N}_i) \quad \forall U \subseteq S(\Omega_i), U \neq \emptyset \tag{3.2}$$

Proof. See Appendix E.3. ■

Example 2 - continued. Let us consider again the example provided in Fig. 3.2a to explain Theorem 3.3.1. For $\Omega_i = \Omega_1$ and $d_j = d_1$, we construct a $\{s_1, s_2\} - d_1$ flow f_{11} over $\mathcal{N}_1 = (V, E, h_1)$ as follows: For $e \in \{(s_1, u), (u, v), (v, d_1), (s_2, d_1)\}$, $f_{11}(e) = 0.5$; otherwise,

$f_{11}(e) = 0$. For $\Omega_i = \Omega_1$ and $d_j = d_2$, we construct a $\{s_1, s_2\} - d_2$ flow f_{12} as follows: For $e \in \{(s_2, u), (u, v), (v, d_2), (s_1, d_2)\}$, $f_{12}(e) = 0.5$; otherwise, $f_{12}(e) = 0$. It is easy to see that $\text{val}(f_{1i}, s_j) = 0.5$ for $i = 1, 2$ and $j = 1, 2$. Similarly, we can verify the case for $\Omega_i = \Omega_2$ and $s_j = d_3, d_4$. This indicates that the MPC can achieve a symmetrical rate $\frac{1}{2}$. ■

3.3.3 Linear Program for MPC

In this section, we formulate a linear program to calculate the performance achieved by MPCs for a given partition of the unicast sessions. Theorem 3.3.1 yields a set of linear constraints to describe the rate region achieved by multicast-packing code for a given partition $\mathcal{G}$. In addition to Eq. (3.1), we add the following linear constraints:

- For each $\Omega_i \in \mathcal{G}$, $d_j \in D(\Omega_i)$ and $s_l \in S(\Omega_i)$, the value of the $S(\Omega_i) - d_j$ flow f_{ij} at s_l equals R_l:

$$R_l = \sum\nolimits_{e \in \text{Out}(s_l)} f_{ij}(e) - \sum\nolimits_{e \in \text{In}(s_l)} f_{ij}(e). \tag{3.3}$$

- For each $\Omega_i \in \mathcal{G}$ and $d_j \in D(\Omega_i)$, f_{ij} must satisfy the flow conservation law at each $v \in V - (S(\Omega_i) \cup \{d_j\})$:

$$\sum\nolimits_{e \in \text{Out}(v)} f_{ij}(e) = \sum\nolimits_{e \in \text{In}(v)} f_{ij}(e). \tag{3.4}$$

- For each $\Omega_i \in \mathcal{G}$, $d_j \in D(\Omega_i)$ and $e \in E$,

$$0 \leq f_{ij}(e) \leq h_i(e). \tag{3.5}$$

Remark. It can be easily seen that if each Ω_i only contains one unicast session, the above linear constraints are reduced to those of a routing scheme, in which each node only forwards

the symbols it receives. Hence, routing can be viewed as a special case of MPC.

In practice, the above constraints can be combined with additional constraints and various objectives to form a linear program. In this paper, we consider the following two objectives:

- *Maximum common rate*: We require that the transmission rate of each unicast session must be at least a rate $R(\mathcal{G})$. In addition to Eq. (3.1) and Eq. (3.3)-(3.5), we add the following linear constraint for each $s_l \in S(\Omega)$,

$$R_l \geq R(\mathcal{G}). \tag{3.6}$$

  The objective is simply:

$$\text{Maximize} \quad R(\mathcal{G}). \tag{3.7}$$

- *Minimum cost*: We require that the transmission rate of each unicast flow ω_l must be at least a fixed value q_l. In addition to Eq. (3.1) and Eq. (3.3)-(3.5), we add the following constraint for each $s_l \in S(\Omega)$,

$$R_l \geq q_l. \tag{3.8}$$

  Let $a : E \to \mathbb{R}_{\geq 0}$ be a function such that $a(e)$ denotes the cost of occupying unit capacity along e. The objective is simply:

$$\text{Minimize} \quad \sum\nolimits_{e \in E} \sum\nolimits_{i=1}^{|\Omega|} a(e) h_i(e). \tag{3.9}$$

Remark. As it is seen, the allocation of network capacities $\mathcal{H}$ are decision variables in the above linear programs (see Eq. (3.1)). Thus, the solution to these linear programs not only allows us to evaluate the performance achieved by MPC for a given partition $\mathcal{G}$, but also

includes $\mathcal{H}$ as part of the LP solution. From practical perspective, we only need to find the best partition $\mathcal{G}$ such that the MPC constructed from the LP solution achieves the best performance among all MPCs for Ω. Yet, when $|\Omega|$ becomes large, finding such partition as an LP solution is computationally expensive. Therefore, we present a practical partitioning algorithm based on simulated annealing techniques in the next section.

3.4 Simulated Annealing Algorithm to Find Good Partitions

In this section, we present a practical partitioning algorithm to approximate the best partition of Ω by employing simulated annealing technique [58]. The running process of the algorithm is divided into stages, each of which is associated with a positive value T (also called *temperature* [58]). During each stage, it performs a random walk in the space of partitions of Ω. The probability that it moves from the current partition $\mathcal{G}$ to another partition $\mathcal{G}_1$ is

$$Pr(r, r_1, T) = \begin{cases} 1 & \text{if } r_1 \text{ is better than } r; \\ \exp(-\frac{\kappa|r_1 - r|}{T}) & \text{otherwise.} \end{cases}$$

where r, r_1 denote the values of the objective function corresponding to $\mathcal{G}$ and $\mathcal{G}_1$, respectively. At the end of each stage, we reduce T by a constant factor. Note that in case $\mathcal{G}_1$ is worse than $\mathcal{G}$, there is still probability that the algorithm will move to $\mathcal{G}_1$. This strategy prevents the algorithm from being stuck at a sub-optimal partition, which is typical of a greedy strategy.

The algorithm consists of the following parts:

Initialization (lines 1-4): The algorithm starts with a trivial partition $\mathcal{G}$, in which each Ω_i

Algorithm 1: Algorithm to find good partition

```
1  G ← {{ω_i} : 1 ≤ i ≤ |Ω|} ;                                  // Initialize partition
2  (r, H) ← solve(G) ;                                          // Solve LP
3  (r_opt, H_opt) ← (r, H); G_opt ← G ;                         // Store the result
4  T ← T_0 ;                                                    // Setting initial temperature
5  for i ← 1 to α do
6  |   j ← 1, k ← 1;
7  |   while j ≤ β and k ≤ ζ do
8  |   |   G_1 ← get(G) ;                                       // Get a new partition from G
9  |   |   (r_1, H_1) ← solve(G) ;                              // Solve LP
10 |   |   if r_1 is better than r_opt then
11 |   |   |   (r_opt, H_opt) ← (r_1, H_1);
12 |   |   |   G_opt ← G_1;
13 |   |   end
14 |   |   if oracle(r, r_1, T) = true then
15 |   |   |   r ← r_1, G ← G_1 ;                               // Move to the new partition
16 |   |   |   k ← k + 1 ;                                      // Record successful moves
17 |   |   end
18 |   |   j ← j + 1;
19 |   end
20 |   T ← T * η ;                                              // Decrease temperature by a factor
21 end
22 return (r_opt, H_opt, G_opt);

23 function get(G)
24   Select Ω_i randomly from G; select ω_l randomly from Ω_i;
25   Select Ω_j randomly from G ∪ {∅} such that Ω_i ≠ Ω_j;
26   Ω_i ← Ω_i − {ω_l}, Ω_j ← Ω_j ∪ {ω_l};
27   if Ω_i is empty then G ← G − {Ω_i};
28   return G;

29 function oracle(r, r_1, T)
30   if r_1 is better than r then return true;
31   Randomly select a number δ in the range [0, 1];
32   if δ < exp(−κ|r−r_1| / T) then return true;
33   else return false;
```

contains only one unicast flow ω_i (line 1). An LP solver is invoked to compute the solution $(r, \mathcal{H})$ to the linear program constructed from $\mathcal{G}$ (line 2), where r denotes the value of the objective function, and $\mathcal{H}$ the allocation of network capacities included in the solution. Then, T is initialized to T_0 (line 4).

The for-loop (lines 5-22): The major body of the algorithm is the for-loop. Each iteration of the for-loop corresponds to a stage. The major body of the for-loop is a while-loop (lines 7-19). At the beginning of the while-loop, the algorithm calls a function get to generate a

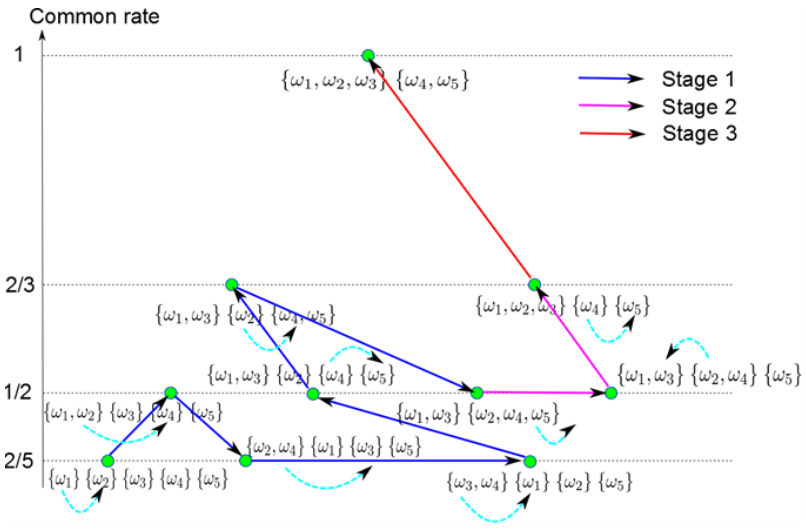


Figure 3.3: An example of the running process of the simulated-annealing algorithm for the network shown in Fig. 3.1. The vertical axis marks the maximal common rates achieved by MPCs with respect to different partitions. The dashed lines denote the operations performed by the get function, and the solid arrows represent transitions between partitions. For example, for the initial partition, the get function moves ω_1 from $\{\omega_1\}$ to $\{\omega_2\}$, resulting in the partition $\{\{\omega_1,\omega_2\},\{\omega_3\},\{\omega_4\},\{\omega_5\}\}$. The algorithm runs for three stages, with the transitions in each stage being marked in a different color. The algorithm finds the optimal partition $\{\{\omega_1,\omega_2,\omega_3\},\{\omega_4,\omega_5\}\}$ after three stages.

new partition $\mathcal{G}_1$ from $\mathcal{G}$ (line 8). The LP solver is invoked to compute the solution of the linear program constructed from $\mathcal{G}$ (line 9). If r_1 is better than the best objective found previously, the algorithm records this better solution (lines 10-13). A function oracle is then called to decide if the algorithm moves to the new partition $\mathcal{G}_1$ (lines 14-17). At the end of each stage, T is reduced by a factor η (line 20). The function get is used to generate a random partition from the given partition $\mathcal{G}$. It first randomly picks up two distinct subsets Ω_i, Ω_j, and a unicast session $\omega_l \in \Omega_i$. Then, it moves ω_l from Ω_i to Ω_j, and returns the final partition.

Fig. 3.3 shows an example of the running process of the algorithm for the example of Fig. 3.1. The algorithm starts from $\{\{\omega_1\},\{\omega_2\},\{\omega_3\},\{\omega_4\},\{\omega_5\}\}$, and the corresponding MPC achieves a maximal common rate of $\frac{2}{5}$. It then moves to the partition $\{\{\omega_1,\omega_2\},\{\omega_3\},\{\omega_4\},\{\omega_5\}\}$, with the maximal common rate increased to $\frac{1}{2}$. Due to the random nature of the anneal-

ing algorithm, at the next step, it moves to $\{\{\omega_2, \omega_4\}, \{\omega_1\}, \{\omega_3\}, \{\omega_5\}\}$, with the maximal common rate decreased to $\frac{2}{5}$. The whole process consists of three stages, marked in different colors. Stage 1 (marked in blue) has a higher temperature than stage 2 (marked in purple). Thus, the algorithm moves more violently in stage 1 than it does in stage 2. At the end of stage 3, the algorithm reaches the optimal partition.

To deal with large scenarios, we divide the space of partitions of Ω into disjoint sub-spaces, and assign each of them to a dedicated processor. Then, these processors run the algorithm in parallel by randomly moving in the assigned sub-spaces. At last, we choose the best partition returned by these processors.

3.5 Evaluation

3.5.1 Simulation Setup

We evaluate the performance of our approach via simulations. We used a network (see Fig. 3.4), which has been used by other researchers [21,27], for our simulations. It has been shown by previous work [21,27] that network coding exhibits better performance than routing only when shared bottlenecks are present. Thus, in our simulations, we focus on communication scenarios, where senders are separated from receivers by bottleneck links, *e.g.,* e_5, e_6 and e_7, that have lower bandwidths and higher costs than other links. By changing the positions of senders and receivers, this network allows us to investigate the influence of the positions of bottlenecks on the performance of MPCs. The outgoing edges of a_i's ($1 \leq i \leq 9$) and the incoming edges of b_i's all have infinite capacities and zero costs. Each multiple-unicast scenario Ω is a subset of $\{(a_i, b_j) : 1 \leq i, j \leq 9\}$ such that the senders and the receivers are all distinct. We considered the cases where $3 \leq |\Omega| \leq 7$. For each setting of $|\Omega|$, we randomly constructed 50 multiple-unicast scenarios. We considered two objectives, maximum common

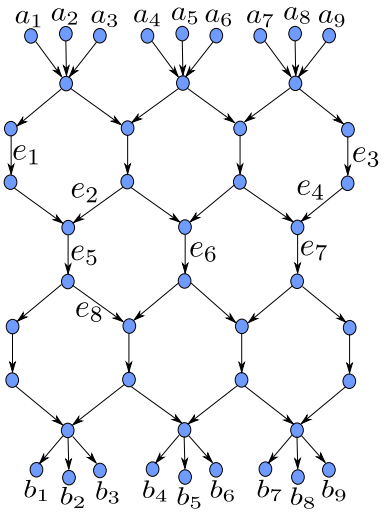


Figure 3.4: The network used for simulation.

rate and minimum cost. For the first objective, we considered two capacity settings for the edges other than the outgoing edges of a_i's and the incoming edges of b_i's:

Scenario 1: The edges all have unit capacities.

Scenario 2: $h(e_2) = h(e_5) = h(e_7) = 0.5, h(e_6) = 0.1$. The other edges have unit capacities. In this network, for the second objective, each edge has infinite capacity. We required that each unicast session must achieve at least unit rate. We considered two cost settings for the edges other than the outgoing edges of a_i's and the incoming edges of b_i's:

Scenario 3: $a(e_1) = a(e_2) = a(e_3) = a(e_4) = 10, a(e_6) = 100$. The other edges have unit costs.

Scenario 4: $a(e_3) = a(e_5) = a(e_7) = 10$, $a(e_6) = 100$. The other edges have unit costs.

Scenarios 2-4 model many practical transmission scenarios, where the end users are communicating with each other through long-distance links, which usually have limited bandwidths and higher costs than local links. The parameters for the simulated annealing algorithm were: $T_0 = 0.5$, $\alpha = 7$, $\beta = 92$, $\zeta = 46$, $\eta = 0.9$, $\kappa = 7$. We used GLPK 4.47 as the LP solver. All the simulations were run on a desktop computer with Intel core i3 CPU and 2GB memory.

Table 3.1: Simulation results.

$\|\Omega\|$	δ (%)	λ (%)	τ (sec)
3	36	27.78	5.07
4	20	100	4.74
5	10	28.75	11.23
6	46	89.13	9.75
7	34	18.37	13.08

(a) Max. common rate, Scenario 1

$\|\Omega\|$	δ (%)	λ (%)	τ (sec)
3	18	94.44	3.23
4	20	100	4.74
5	30	96.67	6.71
6	46	89.13	9.75
7	50	84	15.32

(b) Max. common rate, Scenario 2

$\|\Omega\|$	δ (%)	λ (%)	τ (sec)
3	26	27.94	3.47
4	38	21.46	6.64
5	44	26.04	10.20
6	60	26.73	14.01
7	66	25.36	17.78

(c) Min. cost, Scenario 3

$\|\Omega\|$	δ (%)	λ (%)	τ (sec)
3	18	30.59	3.91
4	20	29.51	7.11
5	30	27.22	10.8
6	46	24.69	12.16
7	50	23.5	17.76

(d) Min. cost, Scenario 4

3.5.2 Simulation Results

Let q_m denote the objective obtained by the partitioning algorithm, q_r the optimal objective achieved by routing, and λ_{succ} the number of scenarios in which q_m is better than q_r. We define the following metrics to evaluate the results: i) Performance gain, $\lambda = \frac{|q_m - q_r|}{q_r} \times 100\%$; ii) ratio of scenarios with gains, $\delta = \frac{\lambda_{succ}}{50} \times 100\%$; iii) the average running time τ of the partitioning algorithm. We averaged the performance gains over all scenarios with gains. The simulation results are shown in Table 3.1. We make the following observations:

Except for Scenario 1, the ratio of scenarios for which MPC achieves gains over routing all increases with $|\Omega|$. Moreover, under Scenarios 2-4, MPC outperforms routing in almost half of the scenarios when $|\Omega| = 6, 7$. Under Scenario 3, MPC outperforms routing for more than 60% of the scenarios when $|\Omega| = 6, 7$. For these scenarios, MPC is more scalable than routing in the sense that the chance of obtaining performance gains through MPC increases with $|\Omega|$.

Under Scenarios 1-2, we observed that MPC achieves considerably better performance than routing for some scenarios. Under Scenario 2, MPC nearly doubles the performance of

routing for the scenarios with gains. Under Scenarios 3-4, MPC still achieves a performance gain over routing ranging from 23% to 30% for the scenarios with gains. Since the ILP-based approach [21] converges very slow even for four unicast flows, we compare MPC with the evolutionary approach [27]. We consider the particular scenario presented in [27], where $\Omega = \{(a_1, b_7), (a_2, b_1), (a_7, b_2), (a_8, b_5)\}$ and the cost setting is the same as Scenario 3. For this scenario, MPC achieves a cost of 148, whereas the best cost achieved by the evolutionary approach over 30 runs of simulations is 156 [27].

The simulated annealing algorithm is efficient in finding good partitions. For most scenarios, the running time of the algorithm never exceeds 17 seconds. This is mainly because the algorithm only needs to search in the space of the partitions of Ω. Note that, even for $|\Omega| = 5$, the integer linear program in [21] contains around 68700 and 1400 variables, and around 67500 and 1700 constraints. This makes the converging speed of the integer linear program very slow, and may fail to converge in a reasonable time. In contrast, for $|\Omega| = 7$ and $|\mathcal{G}| = 6$, our linear program contains only 750 variables and 791 constraints, and each stage of the algorithm takes no more than 2 seconds for most scenarios. The evolutionary approach in [27] performs a random walk in the space of the coding vectors in the network. With each generation taking around one second, the total running time is around 100 seconds for 100 generations. In contrast, the simulated annealing algorithm preforms a random walk in the space of the partitions of Ω, which is much smaller than the space of the coding vectors. This greatly reduces the random steps the algorithm takes. The simulation results fully demonstrate the efficiency of the simulated annealing algorithm in finding good partitions.

3.6 Summary

In this chapter, we propose a novel approach, MPC, to construct linear network coding schemes for multiple unicast sessions. We propose a set of linear constraints to describe the

rate region achieved by MPC for a given partition of the multiple unicast sessions. These linear constraints can be combined with various objectives and additional constraints to form linear programs to calculate the performance achieved by MPC. The succinct formulation of these linear programs allow us to quickly analyze the performance of MPC. We further present a practical partitioning algorithm to find good partitions such that the resulting MPC approximates the best performance among all MPCs. Simulation results demonstrate the performance of MPC and the efficiency of the partitioning algorithm.

Chapter 4

Routing-Optimal Networks for Multiple Unicast Sessions

4.1 Introduction

In this chapter, we consider the nonlinear network model, in which each node can perform non-linear network coding operations. In general, non-linear network coding schemes can achieve better rate than linear network coding schemes [13]. Yet, there exist networks, for which routing is sufficient to achieve any rate vector achieved by any linear/nonlinear network coding schemes. We refer to these networks as *routing-optimal* networks. We attempt to answer the following questions: 1) What are the distinct topological features of these networks? 2) Why do these features make a network routing-optimal? The answers to these questions will not only explain which kind of networks can or cannot benefit from network coding, but will also deepen our understanding on how network topologies affect the rate region of network coding.

A major challenge is that there is currently no effective method to calculate the rate region of

network coding. Some researchers proposed to use information inequalities to approximate the rate region [13, 14]. However, except for very simple networks, it is very difficult to use this approach since there is potentially an exponential number of inequalities that need to be considered. [17, 18] provides a formula to calculate the rate region by finding all possible entropy functions, which are vectors of an exponential number of dimensions, thus very difficult to solve even for simple networks.

In this chapter, we employ a graph theoretical approach in conjunction with information inequalities to identify topological features of routing-optimal networks. Our high-level idea is as follows. Consider a network code. For each unicast session, we choose a cut-set C between source and sink, and a set $\mathcal{P}$ of paths from source to sink such that each path in $\mathcal{P}$ passes through an edge in C. Since the information transmitted from the source is totally contained in the information transmitted along the edges in C, we can think of distributing the source information along the edges in C (details will be explained later). Moreover, we consider a routing scheme in which the traffic transmitted along each path $P \in \mathcal{P}$ is exactly the source information distributed over the edge in C that is traversed by P. Such a routing scheme achieves the same rate vector as the network code. However, since the edges might be shared among multiple unicast sessions, such a routing scheme might not satisfy the edge capacity constraints. This suggests that the cut-sets and path-sets we choose for the unicast sessions should have special features. These are essentially the features we are looking for to describe routing-optimal networks.

We make the following contributions:

- We identify a class of networks, called *information-distributive* networks, which are defined by three topological features. The first two features capture how the edges in the cut-sets are connected to the sources and the sinks, and the third feature captures how the paths in the path-sets overlap with each other. Due to these features, given

a network code, there is always a routing scheme such that it achieves the same rate vector as the network code, and the traffic transmitted through the network is exactly the source information distributed over the cut-sets between the sources and the sinks.

- We prove that if a network is information-distributive, it is routing-optimal. We also show that the converse is not true. This indicates that the three features might be too restrictive in describing routing-optimal networks.

- We present examples of information-distributive networks taken from the index coding problem [31, 59] and single unicast with hard deadline constraint.

The rest of this chapter is organized as follows. In Section 4.2, we present related work. In Section 4.3, we present preliminaries. In Section 4.4, we present the detailed description of information-distributive networks. In Section 4.5, we present examples of information-distributive networks related to index coding problem, and single-unicast with hard deadline constraint. In Section 4.6, we show that information-distributive networks don't include all routing-optimal networks.

4.2 Related Work

Network coding was first proposed as an alternative technique to routing, with the expectation that it will achieve better rate than routing. However, it was shown that for certain networks, the benefits gained from network coding compared with routing are very limited. For example, Yin et al. considered bidirected networks with a multicast session, where for each edge (u, v), there exists an edge (v, u) with opposite direction [60]. They showed that the coding advantage, defined as the ratio between the maximal rate achieved by network coding and that achieved by routing, is upper-bounded by a link capacity parameter. In particular, if the two edges with opposite directions have the same capacity, the coding

advantage equals one, *i.e.,* network coding provides no advantage over routing. Some researchers consider network coding for multiple unicast sessions over undirected networks, where the capacity of an undirectional edge is shared by the two opposite channels between the two end nodes of the edge. It was conjectured that network coding provides no benefit over routing for undirected networks [61]. Jain et al. proved that the conjecture is true for a undirected bipartite network [62]. They further proved that for directed bipartite networks, the maximum rate achieved by network coding is upper bounded by a rational number. Xiahou et al. utilized a space information flow approach to prove that the conjecture holds for certain networks, and they further proved the coding gain is upper bounded for certain undirected networks [63]. Langberg and Médard proved that the coding gain is upper bounded by 3 for certain networks with strong connectivity [64]. Sengupta et al. showed that network coding doesn't provide any benefit over routing for a P2P network model, where the uplink capacity of each node is much lower than the downlink capacity [65].

4.3 Preliminaries

4.3.1 Network Model

The network is represented by an acyclic directed multi-graph $G = (V, E)$, where V and E are the set of nodes and the set of edges in the network respectively. Edges are denoted by $e = (u, v, i) \in V \times V \times \mathbb{Z}_{\geq 0}$, or simply by (u, v), where $v = \text{head}(e)$ and $u = \text{tail}(e)$. Each edge represents an error-free and delay-free channel with capacity rate of one. Let $\text{In}(v)$ and $\text{Out}(v)$ denote the set of incoming edges and the set of outgoing edges at node v.

There are $K \geq 1$ unicast sessions in the network. The ith unicast session is denoted by a tuple $\omega_i = (s_i, d_i)$, where s_i and d_i are the source and the sink of ω_i respectively. The message sent from s_i to d_i is assumed to be a uniformly distributed random variable Y_i with

finite alphabet $\mathcal{Y}_i = \{1, \cdots, \lceil 2^{nR_i} \rceil\}$, where R_i is the source information rate at s_i. All Y_i's are mutually independent. Given $1 \leq i \leq j \leq K$, denote $Y_{i:j} = \{Y_m : i \leq m \leq j\}$. We assume $\text{In}(s_i) = \text{Out}(d_i) = \emptyset$ for all $1 \leq i \leq K$.

Let $\text{mincut}(u, v, G)$ denote the minimum capacity of all cut-sets between two nodes u and v. Given two nodes u, v, let $\mathcal{P}_{uv}$ denote the set of directed paths from u to v. The *routing domain* of ω_i, denoted by G_i, is the sub-graph induced by the edges of the paths in $\mathcal{P}_{s_i d_i}$.

4.3.2 Routing Scheme

A *routing scheme* is a transmission scheme where each node only replicates and forwards the received messages onto its outgoing edges. Define the following linear constraints:

$$\sum_{P \in \mathcal{P}_{s_i d_i}} f_i(P) \geq R'_i \qquad \forall 1 \leq i \leq K \tag{4.1}$$

$$\sum_{i=1}^{K} \sum_{P \in \mathcal{P}_{s_i d_i}, e \in P} f_i(P) \leq 1 \qquad \forall e \in E \tag{4.2}$$

where $f_i(P) \in \mathbb{R}_{\geq 0}$ represents the amount of traffic routed through path P for ω_i. A rate vector $\mathbf{R} = (R'_i : 1 \leq i \leq K) \in \mathbb{R}^K_{\geq 0}$ is achievable by routing scheme if there exist $f_i(P)$'s such that (4.1) and (4.2) are satisfied. The rate region of routing scheme, denoted by $\mathcal{R}_r$, is the set of all rate vectors achievable by routing scheme.

4.3.3 Network Coding Scheme

A network coding scheme is defined as follows: [17]

Definition 4.3.1. An $(n, (\eta_e : e \in E), (R_i : 1 \leq i \leq K), (\delta_i : 1 \leq i \leq K))$ *network code* with block length n is defined by:

1. for each $1 \leq i \leq K$ and $e \in \text{Out}(s_i)$, a local encoding function: $\phi_e : \mathcal{Y}_i \to \{1, \cdots, \eta_e\}$;
2. for each $v \in V - \{s_i, d_i : 1 \leq i \leq K\}$ and $e \in \text{Out}(v)$, a local encoding function: $\phi_e : \prod_{e' \in \text{In}(v)} \{1, \cdots, \eta_{e'}\} \to \{1, \cdots, \eta_e\}$;
3. for each $1 \leq i \leq K$, a decoding function: $\psi_i : \prod_{e' \in \text{In}(d_i)} \{1, \cdots, \eta_{e'}\} \to \mathcal{Y}_i$;
4. for each $1 \leq i \leq K$, the decoding error for ω_i is $\delta_i = Pr(\tilde{\psi}_i(Y_{1:K}) \neq Y_i)$, where $\tilde{\psi}_i(Y_{1:K})$ is the value of ψ_i as a function of $Y_{1:K}$.

Given $e \in E$, let $U_e = \tilde{\phi}_e(Y_{1:K})$, where $\tilde{\phi}_e(Y_{1:K})$ is the value of ϕ_e as a function of $Y_{1:K}$, denote the random variable transmitted along e in a network code. For a subset $C \subseteq E$, denote $U_C = \{U_e : e \in C\}$.

Definition 4.3.2. A rate vector $\mathbf{R} = (R'_i : 1 \leq i \leq K) \in \mathbb{R}^K_{\geq 0}$ is *achievable* by network coding if for any $\epsilon > 0$, there exists for sufficiently large n, an $(n, (\eta_e : e \in E), (R_i : 1 \leq i \leq K), (\delta_i : 1 \leq i \leq K))$ network code such that the following conditions are satisfied:

$$\frac{1}{n} \log \eta_e \leq 1 + \epsilon \qquad \forall e \in E \tag{4.3}$$

$$R_i \geq R'_i - \epsilon \qquad \forall 1 \leq i \leq K \tag{4.4}$$

$$\delta_i \leq \epsilon \qquad \forall 1 \leq i \leq K \tag{4.5}$$

The *capacity region* achieved by network coding, denoted by $\mathcal{R}_{nc}$, is the set of all rate vectors $\mathbf{R}$ achievable by network coding.

Given a network code that satisfies (4.3)-(4.5), the following inequalities must hold:

$$\frac{1}{n} H(U_e) \leq \frac{1}{n} \log(\eta_e) \leq 1 + \epsilon \qquad \forall e \in E \tag{4.6}$$

$$\frac{1}{n} H(Y_i) = \frac{1}{n} \log(\lceil 2^{nR_i} \rceil) \geq R_i \geq R'_i - \epsilon \quad \forall 1 \leq i \leq K \tag{4.7}$$

$$\frac{1}{n} I(Y_i; U_{\text{In}(d_i)}) \geq (1 - \epsilon)(R'_i - \epsilon) - \frac{1}{n} \qquad \forall 1 \leq i \leq K \tag{4.8}$$

where (4.8) is due to Fano's Inequality:

$$\begin{aligned}&\frac{1}{n}I(Y_i;U_{\mathrm{In}(d_i)}) \geq \frac{1}{n}(H(Y_i) - \delta_i \log|\mathcal{Y}_i| - 1)\\ =&\frac{1}{n}(1-\delta_i)H(Y_i) - \frac{1}{n} \geq (1-\epsilon)(R_i' - \epsilon) - \frac{1}{n}\end{aligned}$$

4.3.4 Routing-Optimal Networks

Since routing scheme is a special case of network coding, $\mathcal{R}_r \subseteq \mathcal{R}_{nc}$.

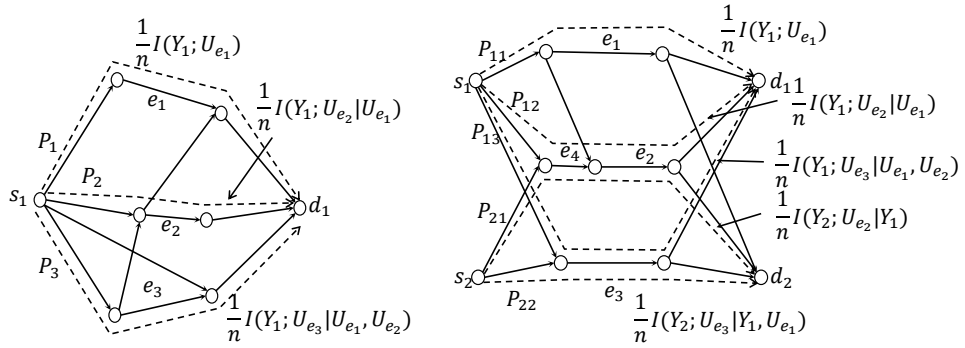
Definition 4.3.3. A network is said to be *routing-optimal*, if $\mathcal{R}_{nc} = \mathcal{R}_r$, *i.e.,* for such network, routing is sufficient to achieve the whole rate region of network coding.

4.4 A Class of Routing-Optimal Networks

In this section, we present a class of routing-optimal networks, called *information-distributive* networks. We first use examples to illustrate the topological features of these networks, and show why they make the networks routing-optimal. Then, we define these networks more rigorously.

4.4.1 Illustrative Examples

Example 4.4.1. We start with the simplest case of single unicast. It is well known that for this case, a network is always routing-optimal [1]. In this example, we re-investigate this case from a new perspective in order to highlight some of the important features that make it routing optimal. Let $m = \mathrm{mincut}(s_1, d_1, G)$, and $C = \{e_1, \cdots, e_m\}$ is a cut-set between s_1 and d_1. Assume $R_1' \in \mathcal{R}_{nc}$. Therefore, for $\epsilon = \frac{1}{k} > 0$ ($k \in \mathbb{Z}_{>0}$), there exists a network code



(a) Single unicast session (b) Two unicast sessions

Figure 4.1: Examples of information-distributive networks, where s_i, d_i $(1 \leq i \leq 2)$ are the source and the sink of the ith unicast session respectively. For each network, we also show a routing scheme that achieves the same rate vector as network coding scheme, where the dashed lines represent the paths that carry non-zero traffic. Beside each such path, we also mark the amount of traffic carried by the path.

such that (4.3)-(4.5) are satisfied. In the followings, all the random variables are defined in this network code.

One important feature of this network is that each path from s_1 to d_1 must pass through at least an edge in C. Thus, $U_{\mathrm{In}(d_1)}$ is a function of U_C. The following inequality holds:

$$I(Y_1; U_{\mathrm{In}(d_1)}) \leq I(Y_1; U_C) \tag{4.9}$$

The following equation holds:

$$I(Y_1; U_C) = \sum_{j=1}^{m} I(Y_1; U_{e_j} | U_{\{e_1,\cdots,e_{j-1}\}}) \tag{4.10}$$

Intuitively, we can interpret (4.10) as follows: $I(Y_1; U_{e_1})$ is the amount of information about Y_1 that can be obtained from U_{e_1}, $I(Y_1; U_{e_2}|U_{e_1})$ the amount of information about Y_1 that can be obtained from U_{e_2}, excluding those already obtained from U_{e_1}, and so on. Hence, (4.10) can be seen as a "distribution" of the source information over the edges in C. Moreover, for

each $1 \leq j \leq m$, we have:

$$I(Y_1; U_{e_j} | U_{\{e_1, \cdots, e_{j-1}\}}) \leq H(U_{e_j}) \tag{4.11}$$

Another important feature is that due to Menger's Theorem, there exist m edge-disjoint paths, $P_1, \cdots, P_m$, from s_1 to d_1 such that $e_j \in P_j$ for $1 \leq j \leq m$. Due to this feature, we can construct a routing scheme by simply letting each P_j transmit the information distributed on e_j:

$$f^{n,k}(P) = \begin{cases} \frac{1}{n} I(Y_1; U_{e_j} | U_{\{e_1, \cdots, e_{j-1}\}}) & \text{if } P = P_j, 1 \leq j \leq m \\ 0 & \text{otherwise.} \end{cases} \tag{4.12}$$

In Fig. 4.1a, we depict such a routing scheme. Clearly, due to (4.6) and (4.11), the above routing scheme satisfies the following inequalities:

$$f^{n,k}(P_j) \leq \frac{1}{n} H(U_{e_j}) \leq 1 + \frac{1}{k} \tag{4.13}$$

Moreover, due to (4.8)-(4.10), we have:

$$\begin{aligned} &\sum_{P \in \mathcal{P}_{s_1 d_1}} f^{n,k}(P) = \sum_{j=1}^{m} f^{n,k}(P_j) = \frac{1}{n} I(Y_1; U_C) \\ \geq &\frac{1}{n} I(Y_1; U_{\mathrm{In}(d_1)}) \geq \left(1 - \frac{1}{k}\right) \left(R_i' - \frac{1}{k}\right) - \frac{1}{n} \end{aligned} \tag{4.14}$$

Since $f^{n,k}(P_j)$ have an upper bound (see (4.13)), there exists a sub-sequence $(n_l, k_l)_{l=1}^{\infty}$ such that each sequence $(f^{n_l,k_l}(P_j))_{l=1}^{\infty}$ approaches a finite limit. Define the following routing

scheme:

$$f_1(P) = \begin{cases} \lim_{l\to\infty} f^{n_l,k_l}(P) & \text{if } P = P_j (1 \le j \le m); \\ 0 & \text{otherwise.} \end{cases}$$

Due to (4.13) and (4.14), the above routing scheme satisfies (4.1) and (4.2). Hence, $R'_1 \in \mathcal{R}_r$, which implies $\mathcal{R}_{nc} \subseteq \mathcal{R}_r$. Therefore, the network is routing-optimal. ■

As shown above, two features are essential in making a network with single-unicast routing-optimal. The first feature is the existence of a cut-set such that each path from the source to the sink must pass through an edge in the cut-set. Due to this feature, the source information contained in $U_{\mathrm{In}(d_1)}$ can be completely obtained from the messages transmitted through the cut-set C (see (4.9)). The second feature is the existence of edge-disjoint paths $P_1, \cdots, P_m$, each of which passes through exactly one edge in C. Due to this feature, a routing scheme can be constructed such that the traffic transmitted along the paths $P_1, \cdots, P_m$ is exactly the information distributed on the edges in C (see (4.12)). These two features together guarantee that the routing scheme achieves the same rate as network coding (see (4.13), (4.14)).

However, extending these features to multiple unicast sessions is not straightforward. One difference from single unicast is that $U_{\mathrm{In}(d_i)}$ may not be a function of U_C, where C is a cut-set between s_i and d_i, and thus (4.9) might not hold. Another difference is that the information from multiple unicast sessions might be distributed on an edge, and thus (4.11) might not hold. Moreover, the paths for multiple unicast sesssions might overlap with each other, and thus (4.13) might not hold. These differences suggest that the cut-sets and the paths, over which a routing scheme is to be constructed, should have additional features in order for the resulting routing scheme to achieve the same rate vector as network coding. We use an example to illustrate some of these features.

Example 4.4.2. Consider the network shown in Fig. 4.1b. Consider an arbitrary rate vector $\mathbf{R} = (R_1', R_2') \in \mathcal{R}_{nc}$. Therefore, for $\epsilon = \frac{1}{k}$ ($k \in \mathbb{Z}_{>0}$), there exists a network code that satisfies (4.3)-(4.5). In the sequel, all the random variables are defined in this network code.

For ω_1, we choose a cut-set $C_1 = \{e_1, e_2, e_3\}$ between s_1 and d_1, and a set of paths $\mathcal{P}_1 = \{P_{11}, P_{12}, P_{13}\}$ that pass through e_1, e_2, e_3 respectively; for ω_2, we choose a cut-set $C_2 = \{e_2, e_3\}$ between s_2 and d_2, and a set of paths $\mathcal{P}_2 = \{P_{21}, P_{22}\}$ that pass through e_2, e_3 respectively.

We first investigate C_1, C_2. One important feature is that each path from s_2 to d_1 passes through at least an edge in C_1. Thus, C_1 is also a cut-set between $\{s_1, s_2\}$ and d_1, and $U_{\text{In}(d_1)}$ is a function of U_{C_1}. Hence, we have:

$$I(Y_1; U_{\text{In}(d_1)}) \leq I(Y_1; U_{C_1}) \tag{4.15}$$

Moreover, $\text{Out}(s_1) \cup C_2$ is a cut-set between $\{s_1, s_2\}$ and d_2, and $U_{\text{Out}(s_1)}$ is a function of Y_1. Hence $U_{\text{In}(d_2)}$ is a function of Y_1, U_{C_2}, which implies:

$$I(Y_2; U_{\text{In}(d_2)}|Y_1) \leq I(Y_2; U_{C_2}|Y_1) \tag{4.16}$$

We distribute the source information over C_1, C_2 as follows:

$$\begin{aligned} I(Y_1; U_{C_1}) &= I(Y_1; U_{e_1}) + I(Y_1; U_{e_2}|U_{e_1}) \\ &\quad + I(Y_1; U_{e_3}|U_{\{e_1,e_2\}}) \\ I(Y_2; U_{C_2}|Y_1) &= I(Y_2; U_{e_2}|Y_1) + I(Y_2; U_{e_3}|Y_1, U_{e_2}) \end{aligned} \tag{4.17}$$

Another feature about C_1, C_2 is that edge e_1 is connected to only one source s_1, and thus U_{e_1} is a function of Y_1. As shown below, this feature guarantees that the information distributed

on an edge $e \in C_1 \cup C_2$ is completely contained in U_e. First, for e_1, it can be easily seen that:

$$I(Y_1; U_{e_1}) \leq H(U_{e_1}) \tag{4.18}$$

For e_2, we have:

$$\begin{aligned} & I(Y_1; U_{e_2}|U_{e_1}) + I(Y_2; U_{e_2}|Y_1) \\ \overset{(b)}{=} & I(Y_1; U_{e_2}|U_{e_1}) + I(Y_2; U_{e_2}|Y_1, U_{e_1}) \\ = & I(Y_1, Y_2; U_{e_2}|U_{e_1}) \leq H(U_{e_2}) \end{aligned} \tag{4.19}$$

where (b) is due to the fact that U_{e_1} is a function of Y_1, and thus, $I(Y_2; U_{e_2}|Y_1) = I(Y_2; U_{e_2}|Y_1, U_{e_1})$. Similarly, for e_3, we have:

$$\begin{aligned} & I(Y_1; U_{e_3}|U_{\{e_1,e_2\}}) + I(Y_2; U_{e_3}|Y_1, U_{e_2}) \\ \overset{(c)}{=} & I(Y_1; U_{e_3}|U_{\{e_1,e_2\}}) + I(Y_2; U_{e_3}|Y_1, U_{\{e_1,e_2\}}) \\ = & I(Y_1, Y_2; U_{e_3}|U_{\{e_1,e_2\}}) \leq H(U_{e_3}) \end{aligned} \tag{4.20}$$

where (c) is again due to the fact that U_{e_1} is a function of Y_1.

Next, we investigate $\mathcal{P}_1, \mathcal{P}_2$. One important feature is that if $P \in \mathcal{P}_1$ overlaps with $P' \in \mathcal{P}_2$, $P \cap C_1 = P' \cap C_2$. For example, P_{12} overlaps with P_{21}, and $P_{12} \cap C_1 = P_{21} \cap C_2 = \{e_2\}$. This feature ensures that the information distributed over C_1, C_2 can be further distributed over

the paths in $\mathcal{P}_1, \mathcal{P}_2$. To see this, we construct the following routing scheme:

$$f_1^{n,k}(P) = \begin{cases} \frac{1}{n} I(Y_1; U_{e_j} | U_{\{e_1, \cdots, e_{j-1}\}}) & \text{if } P = P_{1j}, 1 \leq j \leq 3 \\ 0 & \text{otherwise.} \end{cases}$$

$$f_2^{n,k}(P) = \begin{cases} \frac{1}{n} I(Y_2; U_{e_2} | Y_1) & \text{if } P = P_{21}; \\ \frac{1}{n} I(Y_2; U_{e_3} | Y_1, U_{e_2}) & \text{if } P = P_{22}; \\ 0 & \text{otherwise.} \end{cases}$$

Due to (4.18)-(4.20), we can derive that for each $e \in C_1 \cup C_2$,

$$\sum_{i=1}^{2} \sum_{P \in \mathcal{P}_{s_i d_i}, e \in P} f_i^{n,k}(P) \leq \frac{1}{n} H(U_e) \leq 1 + \frac{1}{k} \tag{4.21}$$

For e_4, we have:

$$\begin{aligned} &\sum_{i=1}^{2} \sum_{P \in \mathcal{P}_{s_i d_i}, e_4 \in P} f_i^{n,k}(P) \\ =& f_1^{n,k}(P_{12}) + f_2^{n,k}(P_{21}) \leq \frac{1}{n} H(U_{e_2}) \leq 1 + \frac{1}{k} \end{aligned}$$

Likewise, we can prove that (4.21) holds for all the other edges of the paths in $\mathcal{P}_1 \cup \mathcal{P}_2$. Due to (4.15)-(4.17), the following inequalities hold for $i = 1, 2$

$$\sum_{P \in \mathcal{P}_{s_i d_i}} f_i^{n,k}(P) \geq \left(1 - \frac{1}{k}\right) \left(R_i' - \frac{1}{n}\right) + \frac{1}{n} \tag{4.22}$$

By (4.21), there exists a sub-sequence $(n_l, k_l)_{l=1}^{\infty}$ such that for all $P \in \mathcal{P}_1 \cup \mathcal{P}_2$ and $i = 1, 2$,

the sub-sequence $(f_i^{n_l,k_l}(P))_{l=1}^{\infty}$ approaches a finite limit. Define a routing scheme:

$$f_i(P) = \begin{cases} \lim_{l\to\infty} f_i^{n_l,k_l}(P) & \text{if } P \in \mathcal{P}_i, i = 1, 2; \\ 0 & \text{otherwise.} \end{cases} \tag{4.23}$$

Due to (4.21) and (4.22), $f_i(P)$ satisfies (4.1) and (4.2). Hence, $\mathbf{R} \in \mathcal{R}_r$, and $\mathcal{R}_{nc} \subseteq \mathcal{R}_r$. The network is routing-optimal. ■

4.4.2 Information Distributive Networks

In this section, we present the definition of information-distributive networks. Similarly to single unicast, for each unicast session ω_i $(1 \leq i \leq K)$, we choose a cut-set C_i between s_i and d_i such that $|C_i| = \text{mincut}(s_i, d_i, G_i)$, and a set of paths $\mathcal{P}_i$ from s_i to d_i. The collection of these cut-sets, denoted by $\mathcal{W} = (C_i)_{i=1}^K$, is called a *cut-set sequence*, and the collection of these path-sets, denoted by $\mathcal{K} = (\mathcal{P}_i)_{i=1}^K$, is called a *path-set sequence*. For instance, in Example 4.4.2, we choose a cut-set sequence $\mathcal{W} = (C_i)_{i=1}^2$, where $C_1 = \{e_1, e_2, e_3\}$ is a cut-set between s_1 and d_1, and $C_2 = \{e_2, e_3\}$ is a cut-set between s_2 and d_2, and a path-set sequence $\mathcal{K} = (\mathcal{P}_i)_{i=1}^2$, where $\mathcal{P}_1$ is a path-set from s_1 to d_1, and $\mathcal{P}_2$ a path-set from s_2 to d_2. Moreover, we arrange the edges in each cut-set in $\mathcal{W}$ in some ordering. For instance, in Example 4.4.2, we arrange the edges in C_1 in the ordering $T_1 = (e_1, e_2, e_3)$, and the edges in C_2 in the ordering $T_2 = (e_2, e_3)$. Each such ordering is called a permutation of the edges in the corresponding cut-set. The collection of these permutations, denoted $\mathcal{T} = (T_i)_{i=1}^K$, is called a *permutation sequence*. For $e \in C_i$, let $T_i(e)$ denote the subset of edges before e in T_i. For $e \in E$, define $\mathcal{W}(e) = \{C_i \in \mathcal{W} : e \in C_i\}$, and $\alpha(e)$ the largest index of the source to which tail(e) is connected. The first feature is described below.

Next, we formalize the three features we have shown in Example 4.4.2. The first feature is described below.

Definition 4.4.1. Given a cut-set sequence $\mathcal{W}$, if for all $1 \leq i < j \leq K$, each path from s_j to d_i must pass through an edge in C_i, we say that $\mathcal{W}$ is *cumulative*.

This feature guarantees that the source information contained in the incoming messages at each sink d_i can be completely obtained from $Y_{1:i-1}, U_{C_i}$.

Lemma 4.4.1. Consider a network code as defined in Definition 4.3.1. If $\mathcal{W}$ is a cumulative cut-set sequence, then for each $1 \leq i \leq K$, Y_i is a function of $Y_{1:i-1}, U_{C_i}$, and the following inequality holds:

$$I(Y_i; U_{\mathrm{In}(d_i)}|Y_{1:i-1}) \leq I(Y_i; U_{C_i}|Y_{1:i-1}) \tag{4.24}$$

Proof. See Appendix F.2. ■

Given a cumulative cut-set sequence $\mathcal{W}$ and a permutation sequence $\mathcal{T}$ for $\mathcal{W}$, we can distribute the source information Y_i over the edges in C_i as follows:

$$I(Y_i; U_{C_i}|Y_{1:i-1}) = \sum_{e \in C_i} I(Y_i; U_e|Y_{1:i-1}, U_{T_i(e)}) \tag{4.25}$$

The second feature is presented below. Without loss of generality, let $\mathcal{W}(e) = \{C_{n_1}, \cdots, C_{n_k}\}$, where $1 \leq n_1 < \cdots < n_k \leq K$.

Definition 4.4.2. Given a cut-set sequence $\mathcal{W}$, we say that it is *distributive* if there exists a permutation sequence $\mathcal{T}$ for $\mathcal{W}$ such that for each $e \in \bigcup_{i=1}^{K} C_i$, the following conditions are satisfied: for all $1 \leq j < k$,

$$\alpha(e') \leq n_k \qquad \forall e' \in T_{n_{j+1}}(e) - T_{n_j}(e) \tag{4.26}$$

$$\alpha(e') \leq n_{j+1} - 1 \qquad \forall e' \in T_{n_j}(e) - T_{n_{j+1}}(e) \tag{4.27}$$

As shown in Example 4.4.2, let $T_1 = (e_1, e_2, e_3)$, and $T_2 = (e_2, e_3)$. For e_3, $\mathcal{W}(e_3) = \{C_1, C_2\}$, $T_2(e_3) - T_1(e_3) = \emptyset$, and thus, (4.26) is trivially satisfied; $T_1(e_3) - T_2(e_3) = \{e_1\}$, $\alpha(e_1) = 1$, and (4.27) is satisfied. Similarly, we can verify other edges. Hence, $\mathcal{W}$ is distributive.

The above two features ensure that the information from multiple unicast sessions that is distributed on an edge $e \in \bigcup_{i=1}^{K} C_i$ can be completely obtained from U_e.

Lemma 4.4.2. Consider a network code as defined in Definition 4.3.1. Given a cumulative cut-set sequence $\mathcal{W}$, if $\mathcal{W}$ is distributive, for each $e \in \bigcup_{i=1}^{K} C_i$, the following inequality holds:

$$\sum_{1 \le i \le K, e \in C_i} I(Y_i; U_e | Y_{1:i-1}, U_{T_i(e)}) \le H(U_e) \tag{4.28}$$

Proof. See Appendix F.2. ■

The third feature is presented below.

Definition 4.4.3. Given a path-set sequence $\mathcal{K}$ for $\mathcal{W}$, we say that $\mathcal{K}$ is *extendable*, if for all $1 \le i < j \le K$, $P_1 \in \mathcal{P}_i$ and $P_2 \in \mathcal{P}_j$ such that P_1 overlaps with P_2, $P_1 \cap C_i = P_2 \cap C_j$.

As shown in Example 4.4.2, let $\mathcal{K} = \{\mathcal{P}_1, \mathcal{P}_2\}$. Clearly, we have $P_{12} \cap P_{21} = \{e_2, e_4\}$, $P_{13} \cap C_1 = P_{21} \cap C_2 = \{e_2\}$, and $P_{13} \cap P_{22} = \{e_3\}$, $P_{13} \cap C_1 = P_{22} \cap C_2 = \{e_3\}$. Thus, $\mathcal{K}$ is extendable.

Definition 4.4.4. A network with multiple unicast sessions is said to be *information-distributive*, if there exist a cumulative and distributive cut-set sequence $\mathcal{W}$, and an extendable path-set sequence $\mathcal{K}$ for $\mathcal{W}$ in the network.

As shown in the next theorem, the three features together guarantee that the network is routing-optimal.

Theorem 4.4.1. If a network is information-distributive, it is routing-optimal.

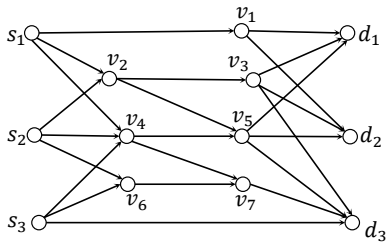


Figure 4.2: An example of information-distributive network with three unicast sessions.

Proof. See Appendix F.2. ■

Example 4.4.3. Consider the network shown in Fig. 4.2. Define the following cut-sets:

$$C_1 = \{(s,v_1),(v_2,v_3),(v_4,v_5)\} \quad C_2 = \{(v_2,v_3),(v_4,v_5)\} \quad C_3 = \{(v_6,v_7),(s_3,d_3)\}$$

Define $\mathcal{W} = (C_i)_{i=1}^3$. Define the following paths:

$$P_{11} = \{(s_1,v_1),(v_1,d_1)\} \quad P_{12} = \{(s_1,v_2),(v_2,v_3),(v_3,d_1)\}$$
$$P_{13} = \{(s_1,v_4),(v_4,v_5),(v_5,d_1)\} \quad P_{21} = \{(s_2,v_2),(v_2,v_3),(v_3,d_2)\}$$
$$P_{31} = \{(s_3,v_6),(v_6,v_7),(v_7,d_3)\} \quad P_{33} = \{(s_3,d_3)\}$$

Define $\mathcal{K} = \{\{P_{11},P_{12},P_{13}\},\{P_{21},P_{22}\},\{P_{31},P_{32}\}\}$. It can be verified that $\mathcal{W}$ is cumulative and distributive, and $\mathcal{K}$ is extendable. The network is information-distributive. ■

4.5 More Examples

4.5.1 Index Coding

We consider a multiple-unicast version of index coding problem [66]. In this problem, there are K terminals $t_1,\cdots,t_K$, a broadcast station s, and K source messages $X_1,\cdots,X_K$, all

available at s. All X_i's are mutually independent random variables uniformly distributed over alphabet $\mathcal{X}_i = \{1, \cdots, 2^m\}$. Each terminal requires X_i, and has acquired a subset of source messages $\mathcal{H}_i$ such that $X_i \notin \mathcal{H}_i$. s uses an encoding function $\phi : \prod_{i=1}^{K} \mathcal{X}_i \to \{1, \cdots, 2^l\}$ to encode the source messages, and broadcasts the encoded message to the terminals through an error-free broadcast channel. Each t_i uses a decoding function ψ_i to decode X_i by using the received message and the messages in $\mathcal{H}_i$. The encoding function ϕ and the decoding functions ψ_i's are collectively called an index code, and l is the length of this index code. The minimum length of an index code is denoted by l_{min}.

This index coding problem can be cast to a multiple-unicast network coding problem over a network $G_1 = (V_1, E_1)$, where $V_1 = \{s_i, d_i : 1 \leq i \leq K\} \cup \{u, v\}$, $E_1 = \{(s_i, u), (v, d_i) : 1 \leq i \leq K\} \cup \{(u, v)\} \cup \{(s_j, d_i) : X_j \in \mathcal{H}_i\}$. The K unicast sessions are $(s_1, d_1), \cdots, (s_K, d_K)$. It can be verified that there exists an index code of length l, if and only if $\mathbf{R} = (\frac{l}{m}, \cdots, \frac{l}{m})$ is achievable by network coding in G_1.

Let $C_i = \{(u, v)\}$, $P_i = \{(s_i, u), (u, v), (v, d_i)\}$. Define $\mathcal{W} = (C_i)_{i=1}^{K}$ and $\mathcal{K} = (\mathcal{P}_i)_{i=1}^{K}$, where $\mathcal{P}_i = \{P_i\}$. Since each C_i contains only one edge, $\mathcal{W}$ is distributive. Meanwhile, since all P_i's overlap at (u, v), $\mathcal{K}$ is extendable.

The following theorem states that if the optimal solution to the index coding problem is to let the broadcast station transmit raw packet, *i.e.,* no coding is needed, then the corresponding multiple-unicast network is information-distributive, and the converse is also true.

Theorem 4.5.1. $l_{min} = mK$ if and only if $\mathcal{W}$ is cumulative, *i.e.,* G_1 is information-distributive.

Proof. See Appendix F.3. ■

Example 4.5.1. In Fig. 4.3, we show an example of G_1, which corresponds to an index coding problem defined by: $\mathcal{H}_1 = \emptyset$, $\mathcal{H}_2 = \{X_1\}$, $\mathcal{H}_3 = \{X_1, X_2\}$, and $\mathcal{H}_4 = \{X_2, X_3\}$.

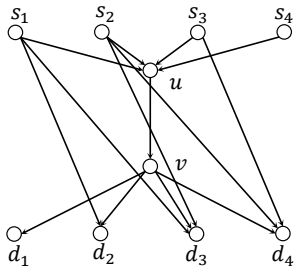


Figure 4.3: The equivalent network coding problem for an index coding problem. The network is information-distributive, and thus no coding is needed in the index coding problem.

Clearly, $\mathcal{W}$ is cumulative, and thus $l_{min} = mK$. ■

4.5.2 Single Unicast with Hard Deadline Constraint

In this example, we consider the network coding problem for a single-unicast session (s, d) over a network $G = (V, E)$, where each edge e is associated with a delay $d_e \in \mathbb{Z}_{>0}$, and each node has a memory to hold received data. Given a directed path P, let $d(P) = \sum_{e \in P} d_e$ denote its delay. For $e \in E$, let $\delta(e)$ denote the minimum delay of directed paths from s to tail(e). The data transmission in the network proceeds in time slots. The messages transmitted from s is represented by a sequence $(Y[t])_{t=0}^{K}$, where $Y[t]$ is a uniformly distributed random variable, and represents the message transmitted from s at time slot t. All $Y[t]$'s are mutually independent. We require that each $Y[t]$ must be received by d within τ time slots. Otherwise, it is regarded as useless, and is discarded. This problem was first proposed by [67] [68]. Recently, it has been shown that network coding can improve throughput by utilizing over-delayed information [69].

This problem can be cast to an equivalent network coding problem for multiple unicast sessions. We construct a time-extended graph $\tilde{G} = (\tilde{V}, \tilde{E})$ as follows: the node set is $\tilde{V} = \{s_t, d_t : 0 \leq t \leq K\} \cup \{v[t] : 0 \leq t \leq K + \tau\}$; for each $e = (u, v) \in E$ and $0 \leq t \leq K+\tau-d_e$, we add an edge $e[t] = (u[t], v[t+d_e])$ to $\tilde{E}$; for $u \in V$, and $0 \leq t \leq K+\tau-1$,

we add M edges from $u[t]$ to $u[t+1]$, where M is the amount of memory available at u; for each $0 \leq t \leq K$, we add J edges from s_t to $s[t]$ and J edges from $d[t+\tau]$ to d_t, where J is a sufficiently large integer. Thus, the original single unicast session (s, d) is cast to $K+1$ unicast sessions $(s_0, d_0), \cdots, (s_K, d_K)$ over $\tilde{G}$.

Let $\tilde{G}[t]$ denote the routing domain for (s_t, d_t), and $m = \text{mincut}(s_0, d_0, \tilde{G}[0])$. It can be seen that each $\tilde{G}[t]$ is simply a time-shifted version of $\tilde{G}[0]$. Given a subset of edges $U \subseteq \tilde{E}$, define $U[t] = \{(u[k+t], v[l+t]) : (u[k], v[l]) \in U\}$. Let $C = \{e_j[t_j] : 1 \leq j \leq m\}$ be a cut-set between s_0 and d_0 such that $e_j \in E$ for $1 \leq j \leq m$, and $\mathcal{P} = \{P_j, \cdots, P_m\}$ a set of edge disjoint paths from s_0 to d_0 such that $e_j[t_j] \in P_j$ for $1 \leq j \leq m$. Let $\mathcal{P}[t] = \{P[t] : P \in \mathcal{P}\}$. We consider the cut-set sequence $\mathcal{W} = (C[t])_{t=0}^{K}$, and the path-set sequence $\mathcal{K} = (\mathcal{P}[t])_{t=0}^{K}$.

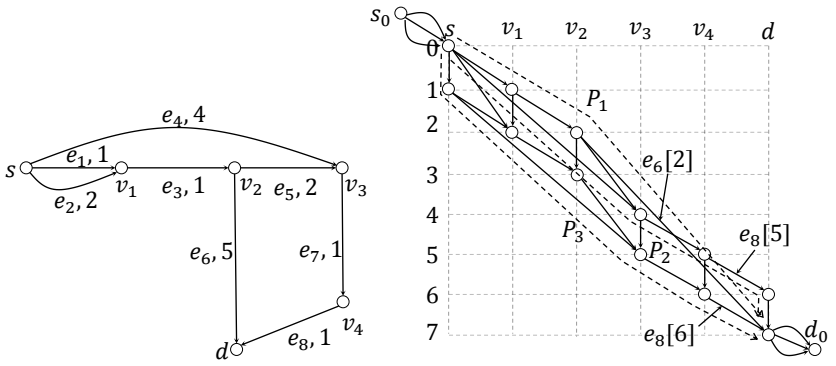
Lemma 4.5.1. $\mathcal{W}$ is cumulative.

Proof. See Appendix F.3. ■

Given $U \subseteq \tilde{E}$, a *recurrent* sequence of U is a sequence consisting of all the edges in U that are time-shifted versions of the same edge. $C[0]$ is said to be *distributive* if there is a re-indexing of the edges in $C[0]$ such that for each recurrent sequence $(e_p[t_{n_j}])_{j=1}^{k}$ of $C[0]$, the following conditions are satisfied:

1. for each $1 < j \leq k$, if $e_q[t_q] \in C[0]$ lies before $e_p[t_{n_j}]$, and $e_q[t_q - t_{n_j} + t_{n_{j-1}}] \notin C[0]$, then $t_q - \delta(e_q) \leq t_{n_j} - t_{n_{j-1}} - 1$;
2. for each $1 \leq j < k$, if $e_q[t_q] \in C[0]$ lies before $e_p[t_{n_j}]$, and $e_q[t_q + t_{n_{j+1}} - t_{n_j}] \notin C[0]$, then $t_q - \delta(e_q) \leq t_{n_j} - t_{n_1}$.

$\mathcal{P}$ is said to be *extendable* if for all $P_i, P_j \in \mathcal{P}$ and $e[k], e[l] \in \tilde{E}$ such that $e[k] \in P_i$ and $e[l] \in P_j$, $e_i = e_j$ and $t_i - t_j = k - l$.



(a) Original network (b) Routing domain for (s_0, d_0)

Figure 4.4: An example of single unicast with deadline constraint $\tau = 7$. (a) shows an network with a single unicast (s, d), where e_k, i denotes the alias of an edge and its corresponding delay respectively. (b) shows the routing-domain between s_0 and d_0 over the corresponding time-extended graph $\tilde{G}$, where the node at coordinate (v, t) is $v[t]$. In this routing-domain, $C[0] = \{e_8[5], e_6[2], e_8[6]\}$ is distributive, and $\mathcal{P} = \{P_1, P_2, P_3\}$ is extendable. Hence, $\tilde{G}$ is information-distributive, and therefore, routing-optimal.

Theorem 4.5.2. If $C[0]$ is distributive, and $\mathcal{P}$ is extendable, $\tilde{G}$ is information-distributive, and thus is routing-optimal.

Proof. See Appendix F.3. ■

Example 4.5.2. In Fig. 4.4a, we show an example of single unicast with delay constraint $\tau = 7$. In Fig. 4.4b, we show the routing domain $\tilde{G}[0]$ for (s_0, d_0). Let $C[0] = \{e_8[5], e_6[2], e_8[6]\}$, and $\mathcal{P} = \{P_1, P_2, P_3\}$, where P_1, P_2, P_3 are marked as black dashed lines in Fig. 4.4b. It can be verified that $C[0]$ is distributive, and $\mathcal{P}$ is extendable. Thus, according to Theorem 4.5.2, $\tilde{G}$ is information-distributive. ■

4.6 The Converse is Not True

Note that information-distributive networks don't subsume all possible routing-optimal networks. In the following, we show an example of such a network.

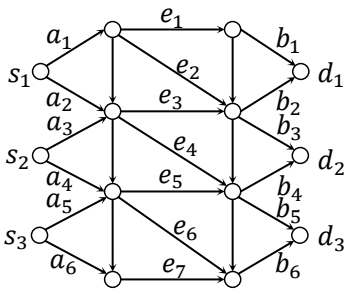


Figure 4.5: A routing-optimal network that is not information-distributive.

Example 4.6.1. Consider the network as shown in Fig. 4.5. We first show that it is not information-distributive. Define the following paths:

$$P_{11} = \{a_1, e_1, b_1\}, P_{12} = \{a_2, e_3, b_2\}$$

$$P_{21} = \{a_3, e_3, b_3\}, P_{22} = \{a_4, e_5, b_4\}$$

$$P_{31} = \{a_5, e_5, b_5\}, P_{32} = \{a_6, e_6, b_6\}$$

For $1 \leq i \leq 3$, let $\mathcal{P}_i = \{P_{i1}, P_{i2}\}$, and $\mathcal{K} = (\mathcal{P}_i)_{i=1}^3$. Since each source has only two outgoing edges, $\mathcal{K}$ is the only-possible path-set sequence. It can be verified that for all cumulative and distributive cut-set sequences, $\mathcal{K}$ is not extendable. For instance, let $C_1 = \{a_1, e_3\}$, $C_2 = \{e_3, b_4\}$, and $C_3 = \{e_5, b_6\}$. Clearly, the cut-set sequence $\mathcal{W} = (C_i)_{i=1}^3$ is cumulative and distributive. However, it can be seen that P_{22} overlaps with P_{31}, but $P_{22} \cap C_2 = \{b_4\}$, and $P_{31} \cap C_3 = \{e_5\}$. Hence, $\mathcal{K}$ doesn't satisfy the condition of Definition 4.4.3. Similarly, we can verify other cases. Thus, the network is not information-distributive.

Nevertheless, we can show that the network is routing-optimal. Consider an arbitrary rate vector $\mathbf{R} = (R_1', R_2', R_3') \in \mathcal{R}_{nc}$. For $\epsilon = \frac{1}{k}$ $(k \geq 2)$, there exists a network code of length n such that (4.6)-(4.8) are satisfied.

Define the following cut-sets, and permutations of edges:

$$C_1 = \{e_1, e_2, e_3\}\ C_2 = \{e_3, e_4, e_5\}\ C_3 = \{e_5, e_6, e_7\}$$

$$T_1 = (e_1, e_2, e_3)\ \ T_2 = (e_3, e_4, e_5)\ \ T_3 = (e_5, e_6, e_7)$$

Define the following permutations:

$$T_1' = (b_1, b_2)\ T_2' = (b_3, b_4)\ T_3' = (b_5, b_6)$$

Let $\mathcal{W} = (C_i)_{i=1}^3$, and $\mathcal{T} = (T_i)_{i=1}^3$. Clearly, $\mathcal{W}$ satisfies the condition of Definition 4.4.1. Thus, according to Lemma 4.4.1, for $i = 1, 2, 3$, the following inequality holds:

$$I(Y_i; U_{\mathrm{In}(d_i)} | Y_{1:i-1}) \leq I(U_{C_i}; Y_i | Y_{1:i-1}) \tag{4.29}$$

Moreover, since $\mathcal{T}$ satisfies the conditions of Definition 4.4.2. By Lemma 4.4.2, for $e \in \bigcup_{i=1}^3 C_i$, the following inequality holds:

$$\sum_{i=1}^{3} \sum_{e \in C_i} I(Y_i; U_e | Y_{1:i-1}, U_{T_i(e)}) \leq H(U_e) \tag{4.30}$$

Define the following paths:

$$P_{11} = \{a_1, e_1, b_1\}, P_{12} = \{a_1, e_2, b_2\}, P_{13} = \{a_2, e_3, b_2\}$$

$$P_{21} = \{a_3, e_3, b_3\}, P_{22} = \{a_3, e_4, b_4\}, P_{23} = \{a_4, e_5, b_4\}$$

$$P_{31} = \{a_5, e_5, b_5\}, P_{32} = \{a_5, e_6, b_6\}, P_{33} = \{a_6, e_7, b_6\}$$

Let $\mathcal{P}_i = \{P_{i1}, P_{i2}, P_{i3}\}$. Define a routing scheme as follows:

$$f_i^{n,k}(P) = \begin{cases} \frac{1}{n} I(Y_i; U_{P\cap C_i} | Y_{1:i-1}, U_{T_i(e)}) & \text{if } P \in \mathcal{P}_i \\ 0 & \text{otherwise.} \end{cases}$$

Note the following inequalities hold for $i = 1, 2, 3$:

$$\begin{aligned} &\frac{1}{n} H(Y_i) \geq \sum_{P\in\mathcal{P}_i} f_i^{n,k}(P) \\ =&\frac{1}{n} I(Y_i; U_{C_i} | Y_{1:i-1}) \overset{(a)}{\geq} \frac{1}{n} I(Y_i; U_{\mathrm{In}(d_i)} | Y_{1:i-1}) \\ =&\frac{1}{n} \sum_{e\in\mathrm{In}(d_i)} I(Y_i; U_e | Y_{1:i-1}, U_{T_i'(e)}) \\ \overset{(b)}{\geq}&\frac{1}{n}(1-\frac{1}{k}) H(Y_i) - \frac{1}{n} \geq (1-\frac{1}{k})(R_2' - \frac{1}{k}) - \frac{1}{n} \end{aligned} \tag{4.31}$$

where (a) holds because $U_{\mathrm{In}(d_i)}$ is a function of $U_{C_2}, Y_{1:i-1}$; (b) is due to Fano's Inequality. For $i = 1, 2, 3$, $e' \in \bigcup_{i=1}^3 C_i$, and $e \in \mathrm{In}(d_i)$, define the following notations:

$$\begin{aligned} &y_i^{n,k} = \frac{1}{n} H(Y_i) \quad u_{e'}^{n,k} = \frac{1}{n} H(U_{e'}) \\ &g_{i,e}^{n,k} = \frac{1}{n} I(Y_i; U_e | Y_{1:i-1}, U_{T_i'(e)}) \end{aligned}$$

Thus, (4.31) can be rewritten in a concise form as:

$$\begin{aligned} &y_i^{n,k} \geq \sum_{P\in\mathcal{P}_i} f_i^{n,k}(P) \geq \sum_{e\in\mathrm{In}(d_i)} g_{i,e}^{n,k} \\ \geq&(1-\frac{1}{k}) y_i^{n,k} - \frac{1}{n} \geq (1-\frac{1}{k})(R_2' - \frac{1}{k}) - \frac{1}{n} \end{aligned} \tag{4.32}$$

Due to (4.31), it can be seen that:

$$\begin{aligned}&\frac{1}{2}y_i^{n,k}-1\leq(1-\frac{1}{k})y_i^{n,k}-\frac{1}{n}\leq\frac{1}{n}I(Y_i;U_{\mathrm{In}(d_i)}|Y_{1:i-1})\\ \leq&\frac{1}{n}\sum_{e\in\mathrm{In}(d_i)}H(U_e)\leq 2(1+\frac{1}{k})\leq 3\end{aligned}$$

This means that $y_i^{n,k}\leq 8$. Clearly, all $y_i^{n,k}$'s, $u_{e'}^{n,k}$'s, $g_{i,e}^{n,k}$'s and $f_i^{n,k}(P)$'s have upper bounds. Thus, there exists a sub-sequence $(n_l,k_l)_{l=1}^{\infty}$ such that $y_i^{n_l,k_l}$, $u_{e'}^{n_l,k_l}$, $g_{i,e}^{n_l,k_l}$ and $f_i^{n_l,k_l}(P)$ approach finite limits when $l\to\infty$. Define the following notations:

$$y_i=\lim_{l\to\infty}y_i^{n_l,k_l}\quad u_{e'}=\lim_{l\to\infty}u_{e'}^{n_l,k_l}\quad g_{i,e}=\lim_{l\to\infty}g_{i,e}^{n_l,k_l}$$

Clearly, the following inequalities holds:

$$u_{e'}\leq 1\quad g_{i,e}\leq u_e\leq 1$$

Define the following routing scheme:

$$f_i(P)=\begin{cases}\lim_{l\to\infty}f_i^{n_l,k_l}(P) & \text{if } P\in\mathcal{P}_i\\ 0 & \text{otherwise.}\end{cases}$$

We will prove that this routing scheme satisfies (4.1) and (4.2). According to (4.31), we see that $\sum_{P\in\mathcal{P}_i}f_i(P)\geq R_2'$, and thus, (4.1) is satisfied. Moreover, due to (4.30), (4.2) is satisfied for $e\in\bigcup_{i=1}^{3}C_i$. For a_3, we have:

$$\begin{aligned}&f_2^{n,k}(P_{21})+f_2^{n,k}(P_{22})=\frac{1}{n}I(Y_2;U_{\{e_3,e_4\}}|Y_1)\overset{(c)}{\leq}\frac{1}{n}I(Y_2;U_{a_3}|Y_1)\\ \leq&H(U_{a_3})\leq 1+\frac{1}{k}\end{aligned}$$

where (c) is due to the fact that $U_{\{e_3,e_4\}}$ is a function of U_{a_3},Y_1. Thus, $f_2(P_{21})+f_2(P_{22})\leq 1$,

and (4.2) is satisfied for a_3. Using similar arguments, we can prove that (4.2) is satisfied for a_1, a_5. Now consider b_4. Due to (4.32), the following equations hold:

$$y_2 = g_{2,b_3} + g_{2,b_4} = f_2(P_{21}) + f_2(P_{22}) + f_2(P_{23}) \tag{4.33}$$

Meanwhile, since U_{b_3} is a function of U_{e_3}, Y_1, the following equations hold:

$$f_2^{n,k}(P_{21}) = \frac{1}{n} I(Y_2; U_{e_3}|Y_1) \geq \frac{1}{n} I(Y_2; U_{b_3}|Y_1) = g_{2,b_3}^{n,k}$$

Hence, $f_2(P_{21}) \geq g_{2,b_3}$. Combining with (4.33), we have:

$$f_2(P_{22}) + f_2(P_{23}) \leq g_{2,b_3} \leq 1$$

Hence, (4.2) holds for b_3. Similarly, we can prove that (4.2) holds for b_2, b_6. It can be easily seen that for all the other edges, (4.2) also holds. Therefore, we have proved that $\mathbf{R} \in \mathcal{R}_r$. This means that $\mathcal{R}_{nc} \subseteq \mathcal{R}_r$, and the network is routing-optimal. ■

4.7 Summary

In this chapter, we present a class of routing-optimal networks, called information-distributive networks, defined by three topological features. Due to these features, there is always a routing scheme that achieves the same rate vector as network coding such that the traffic transmitted through the network is the information distributed over the cut-sets between the sources and the sinks in the corresponding network coding scheme. We then present some examples of information-distributive networks related to index coding and single unicast with hard deadline constraint.

Chapter 5

Conclusion

In this thesis, we consider the inter-session network coding problem for multiple unicast sessions over directed acyclic graphs. In particular, we investigate the problem of the effects of network structure on the rate region achieved by network coding. Based on the types of coding operations allowed at each node, we consider three network models: (i) dummy networks, in which each node can only perform random linear network coding; (ii) linear networks, in which each node can perform linear network coding (not necessarily random); (iii) nonlinear networks, in which each node can perform nonlinear network coding, *i.e.,* the coding operations at each node are unlimited.

For the dummy network model, we apply a precoding-based interference alignment approach, which we refer to as precoding-based network alignment (or PBNA for short), to the network setting. We show that network structures might introduce dependency relations between transfer functions, also called coupling relations, which might affect the achievable rate of PBNA. We observe that since these transfer functions are defined on networks, they usually possess special properties, called graph-related properties. Using these graph-related properties, we identify the minimal set of coupling relations, the presence of which will affect

the achievable rate of PBNA. We present characterizations of these coupling relations in terms of network topology. Based on these characterizations, we present polynomial-time algorithms to check the existence of these coupling relations.

For the linear network model, since finding the optimal linear network coding scheme is NP-hard, we consider a constructive approach to constructing linear network coding schemes, *i.e.,* partitioning the multiple unicast sessions into disjoint subsets of unicast sessions, and mapping each subset of unicast session to a multicast scenario. We refer to this approach as a multicast-packing coding scheme (or MPC for short). We show that the rate region achieved by MPC for a given partition of the multiple unicast sessions can be described by a set of linear constraints. These linear constraints can be combined with various objectives and additional constraints to form linear programs to calculate the performance achieved by MPC. We further present a practical simulated annealing algorithm to find good partitions such that the resulting MPC approximates the best performance among all MPCs. Simulation results demonstrate the performance of MPC and the efficiency of the partitioning algorithm.

For the nonlinear network model, we focus on characterizing the topological features of routing-optimal networks, *i.e.,* for these networks, linear/nonlinear network coding cannot provide any benefit over routing. We identify a class networks, namely information-distributive networks, which are defined by three topological features. We prove that the information-distributive networks are routing-optimal, and show that information-distributive networks don't subsume all possible routing-optimal networks. We further present examples of information-distributive networks related to the index coding problem, and single-unicast with hard deadline constraint.

Bibliography

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

[2] D. S. Lun, N. Ratnakar, M. Mdard, R. Koetter, D. R. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 6, pp. 2608–2623, June 2006.

[3] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1906 –1918, Nov. 2005.

[4] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "A network coding approach to energy efficient broadcasting: from theory to practice," in *the Proceedings of 25th IEEE International Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, April 2006, pp. 1–11.

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

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

[7] Z. Li and B. Li, "Network coding in undirected networks," in *the Proceedings of the 38th Annual Conference on Information Science and Systems (CISS)*, Princeton, NJ, U.S.A., March 2004, pp. 257–262.

[8] Z. Li, B. Li, and L. C. Lau, "On achieving maximum multicast throughput in undirected networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. SI, pp. 2467–2485, June 2006.

[9] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. M. G. M. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, June 2005.

[10] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[11] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 829–848, March 2006.

[12] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," in *the Proceedings of the 42nd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, U.S.A., Sept. 2004, pp. 11–19.

[13] 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.

[14] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Mdard, "Insufficiency of linear coding in network information flow," *IEEE Transactions on Information Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[15] N. J. A. Harvey, R. Kleinberg, and A. R. Lehman, "On the capacity of information networks," *IEEE/ACM Transactions on Networking (TON) - Special issue on networking and information theory*, vol. 52, no. SI, pp. 2345–2364, June 2006.

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

[17] X. Yan, R. W. Yeung, and Z. Zhang, "The capacity region for multi-source multi-sink network coding," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Nice, France, June 2007, pp. 116–120.

[18] T. Chan and A. Grant, "On capacity regions of non-multicast networks," in *the Proceedings of IEEE International Symposium onInformation Theory Proceedings (ISIT)*, Austin, TX, U.S.A., June 2010, pp. 2378–2382.

[19] A. R. Lehman and E. Lehman, "Complexity classification of network information flow problems," in *the Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, Philadelphia, PA, U.S.A., 2004, pp. 142–150.

[20] M. Langberg and A. Sprintson, "On the hardness of approximating the network coding capacity," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Toronto, ON, Canada, July 2008, pp. 315–319.

[21] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Médard, "Network coding for multiple unicasts: An approach based on linear optimization," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Seattle, WA, U.S.A., July 2006, pp. 1758–1762.

[22] N. Ratnakar, R. Koetter, and T. Ho, "Linear flow equations for network coding in the multiple unicast case," in *the Proceedings of DIMACS Working Group on Network Coding*, Piscataway, NJ, U.S.A., Jan. 2005.

[23] A. Eryilmaz and D. S. Lun, "Control for inter-session network coding," in *the Proceedings of the Workshop on Network Coding, Theory and Applications (NetCod)*, San Diego, CA, U.S.A., Jan. 2007, pp. 1–9.

[24] M. Effros, T. Ho, and S. Kim, "A tiling approach to network code design for wireless networks," in *the Proceedings of IEEE Information Theory Workshop (ITW)*, Punta del Este, Uruguay, March 2006, pp. 62–66.

[25] J. Price and T. Javidi, "Network coding games with unicast flows," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 7, pp. 1302–1316, Sept. 2008.

[26] J. R. Marden and M. Effros, "A game theoretic approach to network coding," in *the Proceedings of IEEE Information Theory Workshop on Networking and Information Theory*, Volos, Greek, June 2009, pp. 147–151.

[27] M. Kim, M. Médard, U.-M. O'Reilly, and D. Traskov, "An evolutionary approach to inter-session network coding," in *the Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, Apr. 2009, pp. 450–458.

[28] Y. Wu, P. A. Chou, S.-Y. Kung *et al.*, "Information exchange in wireless networks with network coding and physical-layer broadcast," Tech. Rep., 2005.

[29] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, "Xors in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 3, pp. 497–510, June 2008.

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

[31] Z. Bar-yossef, Y. Birk, T. S. Jayram, and T. Kol, "Index coding with side information," in *the Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, Berkeley, CA, U.S.A., Oct. 2006, pp. 197–206.

[32] A. Das, S. Vishwanath, S. Jafar, and A. Markopoulou, "Network coding for multiple unicasts: An interference alignment approach," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Austin, TX, USA, June 2010, pp. 1878–1882.

[33] C. Meng, A. K. Das, A. Ramakrishnan, S. A. Jafar, A. Markopoulou, and S. Vishwanath, "Precoding-based network alignment for three unicast sessions," Tech. Rep., May 2013. [Online]. Available: http://arxiv.org/abs/1305.0868

[34] A. Ramakrishnan, A. Das, H. Maleki, A. Markopoulou, S. Jafar, and S. Vishwanath, "Network coding for three unicast sessions: Interference alignment approaches," in *the Proceedings of the 48th Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, U.S.A., Sept. 2010, pp. 1054–1061.

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

[36] ——, "Intersession network coding for two simple multicast sessions," in *the Proceedings of Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, U.S.A., Sept. 2007, pp. 682–689.

[37] ——, "Pairwise intersession network coding on directed networks," *IEEE Transactions on Information Theory*, vol. 56, no. 8, pp. 3879–3900, Aug. 2010.

[38] I.-H. Wang, S. U. Kamath, and D. N. Tse, "Two unicast information flows over linear deterministic networks," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Saint-Petersburg, Russia, July 2011, pp. 2462–2466.

[39] S. U. Kamath, D. N. Tse, and V. Anantharam, "Generalized network sharing outer bound and the two-unicast problem," in *the Proceedings of International Symposium on Network Coding (NetCod)*, Boston, MA, U.S.A., June 2011, pp. 1–6.

[40] T. Ho, Y. Chang, and K. J. Han, "On constructive network coding for multiple unicasts," in *the Proceeding of the 44th Allerton Conference on Communication, Control and Computing (Allerton)*, Monticello, IL, U.S.A., Sept. 2006, pp. 779–788.

[41] J. B. Ebrahimi and C. Fragouli, "Properties of network polynomials," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Cambridge, MA, U.S.A., July 2012, pp. 1306–1310.

[42] W.Zeng, C. Viveck., and M. Médard, "An edge reduction lemma and application to linear network coding for two-unicast networks," in *the Proceedings of 50th Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, U.S.A., Oct. 2012, pp. 509–516.

[43] B. Nazer, S. Jafar, M. Gastpar, and S. Vishwanath, "Ergodic interference alignment," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Seoul, South Korea, June 2009, pp. 1769–1773.

[44] G. Bresler, A. Parekh, and D. N. C. Tse, "The approximate capacity of the many-to-one and one-to-many gaussian interference channels," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4566–4592, Sept. 2010.

[45] S. A. Jafar, "Exploiting channel correlations-simple interference alignment schemes with no csit," in *the Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, Miami, FL, U.S.A., Dec. 2010, pp. 1–5.

[46] M. Maddah-Ali and D. Tse, "On the degrees of freedom of miso broadcast channels with delayed feedback," Tech. Rep., 2010.

[47] H. Weingarten, S. Shamai, and G. Kramer, "On the compound mimo broadcast channel," in *the Proceedings of Annual Information Theory and Applications Workshop (ITA)*, San Diego, CA, U.S.A., Jan. 2007.

[48] C. Suh and D. Tse, "Interference alignment for cellular networks," in *the Proceedings of the 46th Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, U.S.A., Sept. 2008, pp. 1037–1044.

[49] N. Lee and J. Lim, "A novel signaling for communication on mimo y channel: Signal space alignment for network coding," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Seoul, South Korea, June 2009, pp. 2892–2896.

[50] S. Gollakota, S. Perli, and D. Katabi, "Interference alignment and cancellation," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 159–170, Oct. 2009.

[51] C. Suh and K. Ramchandran, "Exact-repair mds code construction using interference alignment," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1425–1442, March 2011.

[52] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh, "Asymptotic interference alignment for optimal repair of mds codes in distributed storage," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 2974–2987, May 2013.

[53] J. Han, C.-C. Wang, and N. B. Shroff, "Analysis of precoding-based intersession network coding and the corresponding 3-unicast interference alignment scheme," in *the Proceedings of 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, U.S.A., Sept. 2011, pp. 1033–1040.

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

[55] J. Han, C.-C. Wang, and N. B. Shroff, "Graph-theoretic characterization of the feasibility of the precoding-based 3-unicast interference alignment scheme," Tech. Rep., May 2013.

[56] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge Univ. Press, 1995.

[57] V. R. Cadambe and S. A. Jafar, "Parallel gaussian interference channels are not always separable," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 3983–3990, Sept. 2009.

[58] S. Kirkpatrick, C. D. G. Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[59] Y. Bir and T. ol, "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, June 2006.

[60] X. Yin, X. Wang, J. Zhao, X. Xue, and Z. Li, "On benefits of network coding in bidirected networks and hyper-networks," in *the Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Orlando, FL, U.S.A., March 2012, pp. 325–333.

[61] Z. Li and B. Li, "Network coding in undirected networks," in *the Proceedings of the 38th Annual Conference on Information Sciences and Systems (CISS)*, Princeton, NJ, U.S.A., March 2004.

[62] K. Jain, V. V. Vazirani, and G. Yuval, "On the capacity of multiple unicast sessions in undirected graphs," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2805–2809, June 2006.

[63] T. Xiahou, C. Wu, J. Huang, and Z. Li, "A geometric framework for investigating the multiple unicast network coding conjecture," in *the Proceedings of International Symposium on Network Coding (NetCod)*, Cambridge, MA, U.S.A., June 2012, pp. 37–42.

[64] M. Langberg and M. Médard, "On the multiple unicast network coding conjecture," in *the Proceedings of 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, U.S.A., Sept. 2009, pp. 222–227.

[65] S. Sengupta, M. Chen, P. A. Chou, and J. Li, "On optimality of routing for multi-source multicast communication scenarios with node uplink constraints," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Toronto, ON, Canada, July 2008, pp. 330–334.

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

[67] M. Kodialam and T. Lakshman, "On allocating capacity in networks with path length constrained routing," in *the Proceedings of Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, IL, U.S.A., Sept. 2002.

[68] M. Chen, personal communication, Mar. 2012.

[69] C. Wang and M. Chen, "Sending perishable information: Coding improves delay-constrained throughput even for single unicast," in *the Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Hawaii, U.S.A., June 2014.

Appendices

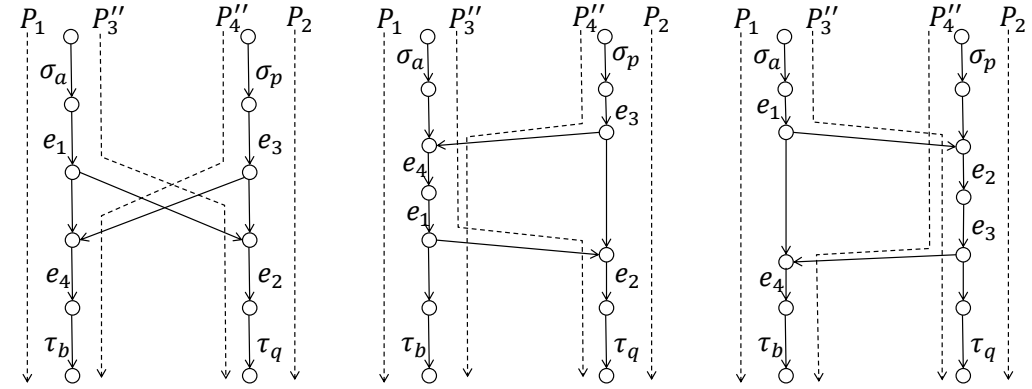
A Proof of Graph-Related Properties of Transfer Functions

A.1 Linearization Property

We first present the following lemma, which plays an important role in the proof of Linearization Property and the interpretation of the coupled relations, $p_i(\mathbf{x}) = 1$ and $p_i(\mathbf{x}) = \eta(\mathbf{x})$. The basic idea of this lemma is that we can multicast two symbols from two senders to two receivers via network coding if and only if the minimum cut separating the senders from the receivers is greater than one.

Lemma A.1. $m_{ab}(\mathbf{x})m_{pq}(\mathbf{x}) \neq m_{aq}(\mathbf{x})m_{pb}(\mathbf{x})$ if and only if there is disjoint path pair $(P_1, P_2) \in \mathcal{P}_{ab} \times \mathcal{P}_{pq}$ or $(P_3, P_4) \in \mathcal{P}_{aq} \times \mathcal{P}_{pb}$.

Proof. We add a super sender s and connect it to s'_a and s'_p via two edges of unit capacity, and a super receiver d, to which we connect d'_b and d'_q via two edges of unit capacity. Thus,



(a) $o(e_2) > o(e_3)$ and $o(e_1) < o(e_4)$ (b) $o(e_2) > o(e_3)$ and $o(e_1) > o(e_4)$ (c) $o(e_2) < o(e_3)$

Figure A.1: The construction of H (in the proof of the Linearization Property) enabled by Lemma A.1 (P_1 is disjoint with P_2)

the transfer matrix at d is

$$\mathbf{M} = \begin{pmatrix} m_{ab}(\mathbf{x}) & m_{aq}(\mathbf{x}) \\ m_{pb}(\mathbf{x}) & m_{pq}(\mathbf{x}) \end{pmatrix}$$

It is easy to see $\det(\mathbf{M}) = m_{ab}(\mathbf{x})m_{pq}(\mathbf{x}) - m_{aq}(\mathbf{x})m_{pb}(\mathbf{x})$. Hence, we can multicast two symbols from s to d, i.e., $\det(\mathbf{M}) \neq 0$, if and only if the minimum cut separating s from d is at least two, or equivalently there is a disjoint path pair $(P_1, P_2) \in \mathcal{P}_{ab} \times \mathcal{P}_{pq}$ or $(P_3, P_4) \in \mathcal{P}_{aq} \times \mathcal{P}_{pb}$. ■

The key to the proof of Lemma 2.6.1 is to find a subgraph H and consider $h(\mathbf{x})$ restricted to H, i.e., $h(\mathbf{x}_H) = \frac{m_{ab}(\mathbf{x}_H)m_{pq}(\mathbf{x}_H)}{m_{aq}(\mathbf{x_H})m_{pb}(\mathbf{x}_H)}$, where $\mathbf{x}_H$ consists of the coding variables in H. In fact, due to graph structure, we can always find H such that some variable $x_{ee'}$ appears exclusively in the numerator or the denominator of $h(\mathbf{x}_H)$. Thus, by assigning values to $\mathbf{x}_H$ other than $x_{ee'}$, we can transform $h(\mathbf{x}_H)$ into a linear function or the inverse of a linear function in terms of $x_{ee'}$. Since $h(\mathbf{x}_H)$ can be acquired through a partial assignment to $\mathbf{x}$, this transformation also holds for the whole graph G. The detailed proof is presented below.

Proof of Lemma 2.6.1. In this proof, given a path P, let $P[e : e']$ denote the path segment of P between two edges e and e', including e, e'. We arrange the edges of G' in topological order, and for $e \in E'$, let $o(e)$ denote e's position in this ordering. Moreover, denote $h_1(\mathbf{x}) = m_{ab}(\mathbf{x})m_{pq}(\mathbf{x})$, $h_2(\mathbf{x}) = m_{aq}(\mathbf{x})m_{pb}(\mathbf{x})$ and $d(\mathbf{x}) = \gcd(h_1(\mathbf{x}), h_2(\mathbf{x}))$. Let $s_1(\mathbf{x}) = \frac{h_1(\mathbf{x})}{d(\mathbf{x})}$ and $s_2(\mathbf{x}) = \frac{h_2(\mathbf{x})}{d(\mathbf{x})}$. Hence $\gcd(s_1(\mathbf{x}), s_2(\mathbf{x})) = 1$. It follows $u(\mathbf{x}) = cs_1(\mathbf{x}), v(\mathbf{x}) = cs_2(\mathbf{x})$, where c is a non-zero constant in $\mathbb{F}_{2^m}$. By Lemma A.1, there exists disjoint path pair $(P_1, P_2) \in \mathcal{P}_{ab} \times \mathcal{P}_{pq}$ or $(P_3, P_4) \in \mathcal{P}_{aq} \times \mathcal{P}_{pb}$. Now we consider the first case.

We arbitrarily select another path pair $(P_3', P_4') \in \mathcal{P}_{aq} \times \mathcal{P}_{pb}$. Since P_1, P_3' both originate at σ_a, and P_2, P_3' both terminate at τ_q, there exist $e_1 \in P_1 \cap P_3'$ and $e_2 \in P_2 \cap P_3'$ such that the path segment along P_3' between e_1 and e_2 is disjoint with $P_1 \cup P_2$. Similarly, there exist $e_3 \in P_2 \cap P_4'$ and $e_4 \in P_1 \cap P_4'$ such that the path segment between e_3 and e_4 along P_4' is disjoint with $P_1 \cup P_2$. Construct the following two paths: $P_3'' = P_1[\sigma_a : e_1] \cup P_3'[e_1 : e_2] \cup P_2[e_2 : \tau_q]$ and $P_4'' = P_2[\sigma_p : e_3] \cup P_4'[e_3 : e_4] \cup P_1[e_4 : \tau_b]$ (see Fig. A.1). Let H denote the subgraph of G' induced by $P_1 \cup P_2 \cup P_3'' \cup P_4''$.

We then prove that the theorem holds for H. If $o(e_2) > o(e_3)$ (Fig. A.1a and A.1b), the variables along $P_2[e_3 : e_2]$ are absent in $h_2(\mathbf{x}_H)$. We then arbitrarily select a variable $x_{ee'}$ from $P_2[e_3 : e_2]$, and write $h_1(\mathbf{x}_H)$ as $f(\mathbf{x}'_H)x_{ee'} + g(\mathbf{x}'_H)$, where $\mathbf{x}'_H$ includes all the variables in $\mathbf{x}_H$ other than $x_{ee'}$. Meanwhile, $h_2(\mathbf{x}_H)$ can be written as $h_2(\mathbf{x}'_H)$. Clearly, $x_{ee'}$ will not show up in $d(\mathbf{x}_H)$ and thus it can also be written as $d(\mathbf{x}'_H)$. We then find values for $\mathbf{x}'_H$, denoted by $\mathbf{r}$, such that $f(\mathbf{r})h_2(\mathbf{r})d(\mathbf{r}) \neq 0$. Finally, denote $c_0 = cg(\mathbf{r})d^{-1}(\mathbf{r})$, $c_1 = cf(\mathbf{r})d^{-1}(\mathbf{r})$ and $c_2 = ch_2(\mathbf{r})d^{-1}(\mathbf{r})$ and the theorem holds. On the other hand, if $o(e_2) < o(e_3)$ (see Fig. A.1c), the variables along $P_1[e_1 : e_4]$ are absent in $h_2(\mathbf{x}_H)$. We then select a variable $x_{ee'}$ from $P_1[e_1 : e_4]$. Similar to above, it's easy to see that $u(\mathbf{x})$ and $v(\mathbf{x})$ can be transformed into $c_1x_{ee'} + c_0$ and c_2 respectively.

For the case where $(P_3, P_4) \in \mathcal{P}_{aq} \times \mathcal{P}_{pb}$ is a disjoint path pair, we can show that $u(\mathbf{x})$ and $v(\mathbf{x})$ can be transformed into c_2 and $c_1x_{ee'} + c_0$ respectively. ■

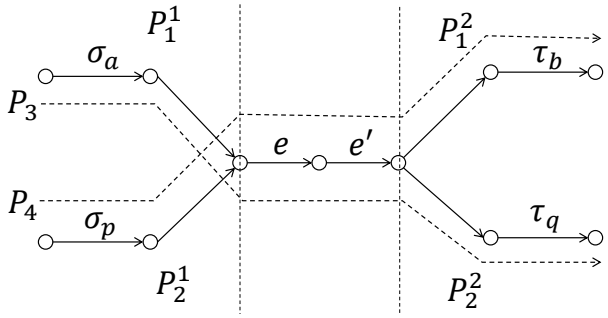


Figure A.2: Illustration of Square-Term Property. A term with $x_{ee'}^2$ introduced by (P_1, P_2) in the numerator of $h(\mathbf{x})$ equals another term introduced by (P_3, P_4) in the denominator of $h(\mathbf{x})$.

A.2 Square-Term Property

The key to the proof of Lemma 2.6.2 is that due to graph structure, each path pair which contributes an $x_{ee'}^2$ term in the numerator of $h(\mathbf{x})$ corresponds to another path pair which contributes an equivalent $x_{ee'}^2$ term in the denominator of $h(\mathbf{x})$. This correspondence relation automatically yields a one-to-one mapping from the $x_{ee'}^2$ terms in the numerator of $h(\mathbf{x})$ to those in the denominator of $h(\mathbf{x})$. Thus, the summation of the $x_{ee'}^2$ terms in the numerator of $h(\mathbf{x})$ equals the summation of the $x_{ee'}^2$ terms in the denominator of $h(\mathbf{x})$, and hence $f_1(\mathbf{x}) = f_2(\mathbf{x})$. The detailed proof is presented below.

Proof of Lemma 2.6.2. First, we define two sets $\mathcal{Q}_1 = \{(P_1, P_2) \in \mathcal{P}_{ab} \times \mathcal{P}_{pq} : x_{ee'}^2 \mid t_{P_1}(\mathbf{x}) t_{P_2}(\mathbf{x})\}$ and $\mathcal{Q}_2 = \{(P_3, P_4) \in \mathcal{P}_{aq} \times \mathcal{P}_{pb} : x_{ee'}^2 \mid t_{P_3}(\mathbf{x}) t_{P_4}(\mathbf{x})\}$. Consider a path pair $(P_1, P_2) \in \mathcal{Q}_1$. Since the degree of $x_{ee'}$ in $t_{P_1}(\mathbf{x})$ and $t_{P_2}(\mathbf{x})$ is at most one, we must have $x_{ee'} \mid t_{P_1}(\mathbf{x})$ and $x_{ee'} \mid t_{P_2}(\mathbf{x})$. Thus $e, e' \in P_1 \cap P_2$. Let P_1^1, P_1^2 be the parts of P_1 before e and after e' respectively. Similarly, define P_2^1 and P_2^2. Then construct two new paths: $P_3 = P_1^1 \cup \{e, e'\} \cup P_2^2$ and $P_4 = P_2^1 \cup \{e, e'\} \cup P_1^2$ (see Fig. A.2). Clearly, $t_{P_1}(\mathbf{x}) t_{P_2}(\mathbf{x}) = t_{P_3}(\mathbf{x}) t_{P_4}(\mathbf{x})$, and thus $(P_3, P_4) \in \mathcal{Q}_2$. The above method establishes a one-to-one mapping $\phi : \mathcal{Q}_1 \to \mathcal{Q}_2$, such that for $\phi((P_1, P_2)) = (P_3, P_4)$, $t_{P_1}(\mathbf{x}) t_{P_2}(\mathbf{x}) = t_{P_3}(\mathbf{x}) t_{P_4}(\mathbf{x})$. Hence, $f_1(\mathbf{x}) = \frac{1}{x_{ee'}^2} \sum_{(P_1,P_2)\in\mathcal{Q}_1} t_{P_1}(\mathbf{x}) t_{P_2}(\mathbf{x}) = \frac{1}{x_{ee'}^2} \sum_{(P_3,P_4)\in\mathcal{Q}_2} t_{P_3}(\mathbf{x}) t_{P_4}(\mathbf{x}) = f_2(\mathbf{x})$. ■

A.3 Other Graph-Related Properties

In this section, we present other graph-related properties, which reveal more microscopic structures of transfer functions, and are to be used in the proofs of Theorems 2.5.3 and 2.5.5. Before proceeding, we first extend the concept of transfer function to any two edges $e, e' \in E'$, i.e., $m_{ee'}(\mathbf{x}) = \sum_{P \in \mathcal{P}_{ee'}} t_P(\mathbf{x})$, where $\mathcal{P}_{ee'}$ is the set of paths from e to e'.

The following lemma states that any transfer function $m_{ee'}(\mathbf{x})$ is fully determined by the two edges e, e'.

Lemma A.2. Consider two transfer functions $m_{e_1e_2}(\mathbf{x})$ and $m_{e_3e_4}(\mathbf{x})$. Then $m_{e_1e_2}(\mathbf{x}) = m_{e_3e_4}(\mathbf{x})$ if and only if $e_1 = e_3$ and $e_2 = e_4$.

Proof. Apparently, the "if" part holds trivially. Now assume $e_1 \neq e_3$ or $e_2 \neq e_4$. Then, there must be some edge which appears exclusively in $\mathcal{P}_{e_1e_2}$ or $\mathcal{P}_{e_3e_4}$, implying $m_{e_1e_2}(\mathbf{x}) \neq m_{e_3e_4}(\mathbf{x})$. Thus, the lemma holds. ■

The following result was first proved by Han et al. [53]. It states that each transfer function $m_{ee'}(\mathbf{x})$ can be uniquely factorized into a product of irreducible polynomials according to the bottlenecks between e and e'.

Lemma A.3. We arrange the bottlenecks in $\mathcal{C}_{ee'}$ in topological order: $e_1, e_2, \cdots, e_k$, such that $e = e_1$, $e' = e_k$. Then, $m_{ee'}(\mathbf{x})$ can be factorized as $m_{ee'}(\mathbf{x}) = \prod_{i=1}^{k-1} m_{e_ie_{i+1}}(\mathbf{x})$, where $m_{e_ie_{i+1}}(\mathbf{x})$ is an irreducible polynomial.

As shown below, any transfer function $m_{ee'}(\mathbf{x})$ can be partitioned into a summation of products of transfer functions according to a cut between e and e'.

Lemma A.4. Assume $\mathcal{U} = \{e_1, e_2, \cdots, e_k\}$ is a cut which separates e from e'. If $e_i || e_j$ for $e_i \neq e_j \in \mathcal{U}$, we have $m_{ee'}(\mathbf{x}) = \sum_{i=1}^{k} m_{ee_i}(\mathbf{x}) m_{e_ie'}(\mathbf{x})$. Otherwise, the above equality doesn't hold.

Proof. For $e_i \in \mathcal{U}$, let $\mathcal{P}^i_{ee'}$ denote the set of paths in $\mathcal{P}_{ee'}$ which pass through e_i. Because $e_i||e_j$ for $e_i \neq e_j \in \mathcal{U}$, $\mathcal{P}^i_{ee'}$ is disjoint with $\mathcal{P}^j_{ee'}$. Hence, $m_{ee'}(\mathbf{x}) = \sum_{i=1}^{k} \sum_{P \in \mathcal{P}^i_{ee'}} t_P(\mathbf{x})$. Note that $m_{ee_i}(\mathbf{x}) m_{e_i e'}(\mathbf{x}) = \sum_{(P_1,P_2) \in \mathcal{P}_{ee_i} \times \mathcal{P}_{e_i e'}} t_{P_1}(\mathbf{x}) t_{P_2}(\mathbf{x})$. Moreover, each monomial $t_P(\mathbf{x})$ in $m_{ee'}(\mathbf{x})$ corresponds to a monomial $t_{P_1}(\mathbf{x}) t_{P_2}(\mathbf{x})$ in $m_{ee_i}(\mathbf{x}) m_{e_i e'}(\mathbf{x})$. Hence, $m_{ee_i}(\mathbf{x}) m_{e_i e'}(\mathbf{x}) = \sum_{P \in \mathcal{P}^i_{ee'}} t_P(\mathbf{x})$, and the lemma holds. On the other hand, if some e_i is upstream of e_j, $P^i_{ee'} \cap P^j_{ee'} \neq \emptyset$, and thus $m_{ee'}(\mathbf{x}) \neq \sum_{i=1}^{k} \sum_{P \in \mathcal{P}^i_{ee'}} t_P(\mathbf{x})$, indicating that the lemma doesn't hold. ■

B Proofs of Feasibility Conditions of PBNA

B.1 Reducing $\mathcal{S}'$ to $\mathcal{S}'_i$

In order to utilize the degree-counting technique, we use the following lemma. Basically, it allows us to reformulate each $\frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}'$ to its unique form $\frac{\alpha(\mathbf{x})}{\beta(\mathbf{x})}$, such that we can compare the degrees of a coding variable in $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ with its degrees in the numerator and denominator of $p_i(\mathbf{x})$ respectively.

Lemma B.1. Let $\mathbb{F}$ be a field. z is a variable and $\mathbf{y} = (y_1, y_2, \cdots, y_k)$ is a vector of variables. Consider four non-zero polynomials $f(z), g(z) \in \mathbb{F}[z]$ and $s(\mathbf{y}), t(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$, such that $\gcd(f(z), g(z)) = 1$ and $\gcd(s(\mathbf{y}), t(\mathbf{y})) = 1$. Denote $d = \max\{d_f, d_g\}$. Define two polynomials in $\mathbb{F}[\mathbf{y}]$: $\alpha(\mathbf{y}) = f\left(\frac{s(\mathbf{y})}{t(\mathbf{y})}\right) t^d(\mathbf{y})$ and $\beta(\mathbf{y}) = g\left(\frac{s(\mathbf{y})}{t(\mathbf{y})}\right) t^d(\mathbf{y})$. Then $\gcd(\alpha(\mathbf{y}), \beta(\mathbf{y})) = 1$.

Proof. See Appendix D. ■

We use the following three steps to reduce $\mathcal{S}'$ to $\mathcal{S}'_i$.

Step 1: $\mathcal{S}' \Rightarrow \mathcal{S}''_1 = \{\frac{a_0 + a_1 \eta(\mathbf{x})}{b_0 + b_1 \eta(\mathbf{x})} : a_0, a_1, b_0, b_1 \in \mathbb{F}_{2^m}\}$. Assume $p_i(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}'$. We will prove that $d = \max\{d_f, d_g\} = 1$. Let $p_i(\mathbf{x}) = \frac{u(\mathbf{x})}{v(\mathbf{x})}$, $\eta(\mathbf{x}) = \frac{s(\mathbf{x})}{t(\mathbf{x})}$ denote the unique

forms of $p_i(\mathbf{x})$ and $\eta(\mathbf{x})$ respectively. Without loss of generality, let $f(z) = \sum_{j=0}^{k} a_j z^j$, $g(z) = \sum_{j=0}^{l} b_j z^j$ where $a_k b_l \neq 0$. We first consider the case where $l \leq k$ and thus $d = k$. Define the following two polynomials:

$$\alpha(\mathbf{x}) = f(\eta(\mathbf{x}))t^k(\mathbf{x}) = \sum_{j=0}^{k} a_j t^{k-j}(\mathbf{x}) s^j(\mathbf{x})$$
$$\beta(\mathbf{x}) = g(\eta(\mathbf{x}))t^k(\mathbf{x}) = \sum_{j=0}^{l} b_j t^{k-j}(\mathbf{x}) s^j(\mathbf{x})$$

Due to Lemma B.1, we have $\alpha(\mathbf{x}) = cu(\mathbf{x}), \beta(\mathbf{x}) = cv(\mathbf{x})$, where c in a non-zero constant in $\mathbb{F}_q$. Moreover, according to Lemma 2.6.1, we assign values to $\mathbf{x}$ other than a coding variable $x_{ee'}$ such that $u(\mathbf{x})$ and $v(\mathbf{x})$ are transformed into:

$$u(x_{ee'}) = c_1 x_{ee'} + c_0 \quad v(x_{ee'}) = c_2$$
$$\text{or } u(x_{ee'}) = c_2 \quad v(x_{ee'}) = c_1 x_{ee'} + c_0$$

where $c_0, c_1, c_2 \in \mathbb{F}_q$ and $c_1 c_2 \neq 0$. We only consider the first case. The proof for the other case is similar. In this case, $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ are transformed into $\alpha(x_{ee'}) = cc_1 x_{ee'} + cc_0$ and $\beta(x_{ee'}) = cc_2$ respectively.

By contradiction, assume $d \geq 2$. We first consider the case where $l \leq k$ and thus $d = k$. In this case, we have

$$\alpha(x_{ee'}) = \sum_{j=0}^{k} a_j t^{k-j}(x_{ee'}) s^j(x_{ee'}) = cc_1 x_{ee'} + cc_0$$
$$\beta(x_{ee'}) = \sum_{j=0}^{l} b_j t^{k-j}(x_{ee'}) s^j(x_{ee'}) = cc_2$$

Assume $s(x_{ee'}) = \sum_{j=0}^{r} s_j x_{ee'}^j$ and $t(x_{ee'}) = \sum_{j=0}^{r'} t_j x_{ee'}^j$, where $s_r t_{r'} \neq 0$. Thus $\max\{r, r'\} \geq 1$. Note that the degree of $x_{ee'}$ in $t^{k-j}(x_{ee'}) s^j(x_{ee'})$ is $kr' + j(r - r')$. We consider the following two cases:

Case I: $r \neq r'$. If $r > r'$, $d_\alpha = kr \geq 2$, contradicting that $d_\alpha = 1$. Now assume $r < r'$.

Let l_1 and l_2 be the minimum exponents of z in $f(z)$ and $g(z)$ respectively. It follows that $d_\alpha = kr' - l_1(r'-r) = 1$ and $d_\beta = kr' - l_2(r'-r) = 0$. Clearly, $l_2 > 0$ due to $d_\beta = 0$. If $r > 0$, $kr' - l_2(r'-r) > kr' - l_2 r' \geq 0$, contradicting $d_\beta = 0$. Hence, $r = 0$, and $l_2 = k$ due to $d_\beta = 0$. Meanwhile, $d_\alpha = (k - l_1)r' = 1$, which implies that $l_1 = k-1$ and $r' = 1$. Thus, z^{k-1} is a common divisor of $f(z)$ and $g(z)$, contradicting $\gcd(f(z), g(z)) = 1$.

Case II: $r = r'$. Since $d_\alpha = 1$ and $d_\beta = 0$, all the terms in $\alpha(x_{ee'})$ and $\beta(x_{ee'})$ containing $x_{ee'}^{kr}$ must be cancelled out, implying that

$$\sum_{j=0}^{k} a_j t_r^{k-j} s_r^j = t_r^k \sum_{j=0}^{k} a_j \left(\frac{s_r}{t_r}\right)^j = t_r^k f\left(\frac{s_r}{t_r}\right) = 0$$
$$\sum_{j=0}^{l} b_j t_r^{k-j} s_r^j = t_r^k \sum_{j=0}^{l} b_j \left(\frac{s_r}{t_r}\right)^j = t_r^k g\left(\frac{s_r}{t_r}\right) = 0$$

Hence $z - \frac{s_r}{t_r}$ is a common divisor of $f(z)$ and $g(z)$, contradicting $\gcd(f(z), g(z)) = 1$.

Therefore, we have proved $d = 1$ when $l \leq k$. Using similar technique, we can prove that $d = 1$ when $l \geq k$. This implies that $\frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))}$ can only be of the form $\frac{a_0 + a_1\eta(\mathbf{x})}{b_0 + b_1\eta(\mathbf{x})}$. Hence, we have reduced $\mathcal{S}'$ to $\mathcal{S}_1''$.

Step 2: $\mathcal{S}_1'' \Rightarrow \mathcal{S}_2'' = \{1, \eta(\mathbf{x}), 1 + \eta(\mathbf{x}), \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}\}$. We consider the coupling relation $p_1(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))}$. The coupling relations $p_2(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))}$ and $p_3(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))}$ can be dealt with similarly. Define $q_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{p_1(\mathbf{x})} = \frac{m_{11}(\mathbf{x})m_{23}(\mathbf{x})}{m_{13}(\mathbf{x})m_{21}(\mathbf{x})}$. Assume the characteristic of $\mathbb{F}_q$ is p. Given an integer m, let m_p denote the remainder of m divided by p. Since $\mathcal{S}_1''$ only consists of a finite number of rational functions, we iterate all possible configurations of a_0, a_1, b_0, b_1 as follows:

Case I: $\frac{f(z)}{g(z)} = \frac{a_0 + a_1 z}{b_0 + b_1 z}$, where $a_1 a_0 b_1 b_0 \neq 0$, and $a_0 b_1 \neq a_1 b_0$. For this case, we have $p_1(x_{ee'}) = \frac{a_0 + a_1 p_1(x_{ee'}) q_1(x_{ee'})}{b_0 + b_1 p_1(x_{ee'}) q_1(x_{ee'})}$. It immediately follows

$$q_1(x_{ee'}) = \frac{a_0 c_2^2 - b_0 c_0 c_2 - b_0 c_1 c_2 x_{ee'}}{b_1 c_1^2 x_{ee'}^2 + (2_p b_1 c_0 c_1 - a_1 c_1 c_2) x_{ee'} + b_1 c_0^2 - a_1 c_0 c_2}$$

Let $u_1(x_{ee'}), v_1(x_{ee'})$ denote the numerator and denominator of the above equation respectively. Assume $u_1(x_{ee'}) \mid v_1(x_{ee'})$ and thus $x_{ee'} = \frac{a_0c_2-b_0c_0}{b_0c_1}$ is a solution to $v_1(x_{ee'}) = 0$. However, $v_1(\frac{a_0c_2-b_0c_0}{b_0c_1}) = \frac{a_0c_2^2}{b_0^2}(a_0b_1 - a_1b_0) \neq 0$. Hence, $u_1(x_{ee'}) \nmid v_1(x_{ee'})$. Thus, by the definition of $q_1(\mathbf{x})$ and Lemma 2.6.2, $x_{ee'}^2$ must appear in $u_1(x_{ee'})$, which contradicts the formulation of $u_1(x_{ee'})$.

Case II: $\frac{f(z)}{g(z)} = \frac{a_0+a_1z}{b_1z}$, where $a_0a_1b_0 \neq 0$. Similar to Case I, we can derive

$$q_1(x_{ee'}) = \frac{a_0c_2^2}{b_1c_1^2x_{ee'}^2 + (2_pb_1c_0c_1 - a_1c_1c_2)x_{ee'} + b_1c_0^2 - a_1c_0c_2}$$

which contradicts Lemma 2.6.2.

Case III: $\frac{f(z)}{g(z)} = \frac{a_1z}{b_0+b_1z}$, where $a_1b_0b_1 \neq 0$. Thus $\frac{1}{p_1(\mathbf{x})} = \frac{b_0}{a_1}\frac{1}{\eta(\mathbf{x})} + \frac{b_1}{a_1}$. Since the coefficient of each monomial in the denominators and numerators of $p_1(\mathbf{x})$ and $\eta(\mathbf{x})$ equals one, it follows $\frac{a_0}{b_1} = \frac{b_1}{a_1} = 1$. This indicates that $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{\eta(\mathbf{x})+1}$.

Case IV: $\frac{f(z)}{g(z)} = \frac{a_0}{b_0+b_1z}$, where $a_0b_0b_1 \neq 0$. It follows that

$$q_1(x_{ee'}) = \frac{a_0c_2^2 - b_0c_0c_2 - b_0c_1c_2x_{ee'}}{b_1c_0^2 + 2_pb_1c_0c_1x_{ee'} + b_1c_1^2x_{ee'}^2}$$

Similar to Case I, this also contradicts Lemma 2.6.2.

Case V: $\frac{f(z)}{g(z)} = \frac{a_0}{z}$, where $a_0 \neq 0$. Hence, $q_1(x_{ee'}) = \frac{a_0c_2^2}{c_1^2x_{ee'}^2+2_pc_0c_1x_{ee'}+c_0^2}$, contradicting Lemma 2.6.2.

Case VI: $\frac{f(z)}{g(z)} = a_0 + a_1z$, where $a_0a_1 \neq 0$. Thus, it follows $p_1(\mathbf{x}) = a_0 + a_1\eta(\mathbf{x})$. Similar to Case III, $a_1 = a_0 = 1$, implying that $p_1(\mathbf{x}) = 1 + \eta(\mathbf{x})$.

Case VII: $\frac{f(z)}{g(z)} = a_1z$, where $a_1 \neq 0$. Similar to Case III, $a_1 = 1$ and hence $p_1(\mathbf{x}) = \eta(\mathbf{x})$.

Therefore, we have proved that $\frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))}$ can only take the form of the four rational functions

in $\mathcal{S}_2''$. Thus, we have reduced $\mathcal{S}_1''$ to $\mathcal{S}_2''$.

Step 3: $\mathcal{S}_2'' \Rightarrow \mathcal{S}_i'$. We note that in Proposition 3 of [53], it was proved that $p_1(\mathbf{x}) \neq 1+\eta(\mathbf{x})$, $p_2(\mathbf{x}) \neq \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$ and $p_3(\mathbf{x}) \neq \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$. Combined with the above results, we have reduced $\mathcal{S}_2''$ to $\mathcal{S}_i'$.

In summary, according to Theorem 2.4.1, if the conditions of Theorem 2.5.1 are satisfied, the three unicast sessions can asymptotically achieve the rate tuple $(\frac{1}{2}, \frac{1}{2}, \frac{1}{2})$ through PBNA.

B.2 Necessity of the Conditions in Theorem 2.5.1

As shown in Subsection 2.6.2, each row of $\mathbf{V}_1$ satisfying the alignment conditions corresponds to a non-zero solution to Eq. (2.36).

Lemma B.2. $\text{rank}(z\mathbf{C} - \mathbf{BA}) = n$.

Proof. Denote $\mathbf{D} = \mathbf{BA}$. Let $\mathbf{c}_i$ and $\mathbf{d}_i$ denote the ith column of $\mathbf{C}$ and $\mathbf{D}$ respectively. Hence, $\mathbf{c}_1, \cdots, \mathbf{c}_n$ are linearly independent and so are $\mathbf{d}_1, \cdots, \mathbf{d}_n$. Assume there exist $f_1(z), \cdots, f_n(z) \in \mathbb{F}_{2^m}(\xi)(z)$ such that $\sum_{i=1}^n f_i(z)(z\mathbf{c}_i - \mathbf{d}_i) = 0$. Without loss of generality, assume $f_i(z) = \frac{g_i(z)}{h(z)}$ for $i \in \{1, 2, \cdots, n\}$, where $g_i(z), h(z) \in \mathbb{F}_{2^m}(\xi)[z]$. Thus, $\sum_{i=1}^n g_i(z)(z\mathbf{c}_i - \mathbf{d}_i) = 0$. Let $k = \max_{i\in\{1,2,\cdots,n\}}\{d_{g_i}\}$ and assume $g_i(z) = \sum_{l=0}^k a_{l,i}(\xi)z^l$, where $a_{l,i}(\xi) \in \mathbb{F}_{2^m}(\xi)$. Then, it follows

$$\begin{aligned}
&\sum_{i=1}^n g_i(z)(z\mathbf{c}_i - \mathbf{d}_i) = \sum_{l=0}^k \sum_{i=1}^n (a_{l,i}(\xi)z^{l+1}\mathbf{c}_i - a_{l,i}(\xi)z^l\mathbf{d}_i) \\
=&z^{k+1}\sum_{i=1}^n a_{k,i}(\xi)\mathbf{c}_i + \sum_{l=0}^{k-1} z^{l+1}\sum_{i=1}^n (a_{l,i}(\xi)\mathbf{c}_i - a_{l+1,i}(\xi)\mathbf{d}_i) \\
&-\sum_{i=1}^n a_{0,i}(\xi)\mathbf{d}_i = \mathbf{0}
\end{aligned}$$

Therefore, the following equations must hold:

$$\sum_{i=1}^{n} a_{k,i}(\xi)\mathbf{c}_i = 0 \quad \sum_{i=1}^{n} a_{0,i}(\xi)\mathbf{d}_i = 0$$
$$\sum_{i=1}^{n}(a_{l,i}(\xi)\mathbf{c}_i - a_{l+1,i}(\xi)\mathbf{d}_i) = 0 \quad \forall l \in \{0, \cdots, k-1\}$$

Thus $a_{l,i}(\xi) = 0$ for any $i \in \{1, \cdots, n\}, l \in \{0, \cdots, k\}$, implying $f_i(z) = 0$. Hence, $\text{rank}(z\mathbf{C} - \mathbf{D}) = n$. ■

The following lemma reveals that any non-zero solution to Eq. (2.36) is linearly dependent on the particular vector $(1, z, z^2, \cdots, z^n)$, which forms each row of the precoding matrix $\mathbf{V}_1^*$.

Corollary B.1. Eq. (2.36) has a non-zero solution if and only if $s = 1$. Moreover, when $s = 1$, Eq. (2.36) has a non-zero solution in the form of $\mathbf{r}(z) = (1, z, z^2, \cdots, z^n)\mathbf{F}$, where $\mathbf{F}$ is an $(n+1) \times (n+1)$ matrix over $\mathbb{F}_{2^m}(\xi)$. Moreover, any solution to Eq. (2.36) is linearly dependent on $(1, z, \cdots, z^n)\mathbf{F}$.

Proof. We first prove the "only if" part. If $s = 0$, $z\mathbf{C} - \mathbf{BA}$ is an invertible square matrix. Thus, Eq. (2.36) has only zero solution. Hence, if Eq. (2.36) has only non-zero solution, it must be that $s = 1$.

We then prove the "if" part. Assume $s = 1$. We will construct a non-zero solution to Eq. (2.36) as follows. There must be an $n \times n$ invertible submatrix in $z\mathbf{C} - \mathbf{D}$. Without loss of generality, assume this submatrix consists of the top n rows of $z\mathbf{C} - \mathbf{D}$ and denote this submatrix by $\mathbf{E}_{n+1}$. Let $\mathbf{b}$ denote the $(n+1)$th row of $z\mathbf{C} - \mathbf{D}$. In order to get a non-zero solution to equation (2.36), we first fix $r_{n+1}(z) = -1$. Therefore, equation (2.36) is transformed into $(r_1(z), \cdots, r_n(z))\mathbf{E}_{n+1} = \mathbf{b}$. Let $\mathbf{E}_i$ denote the submatrix acquired by replacing the ith row of $\mathbf{E}_{n+1}$ with $\mathbf{b}$. Hence, we get a non-zero solution to (2.36), $\mathbf{r}(z) = (\frac{\det \mathbf{E_1}}{\det \mathbf{E}_{n+1}}, \cdots, \frac{\det \mathbf{E}_n}{\det \mathbf{E}_{n+1}}, -1)$. Moreover, $\bar{\mathbf{r}}(z) = (\det \mathbf{E}_1, \cdots, \det \mathbf{E}_n, -\det \mathbf{E}_{n+1})$ is also

a solution. Note that the degree of z in each $\det \mathbf{E}_i$ is at most n. Thus, $\bar{\mathbf{r}}(z)$ can be formulated as $(1, z, \cdots, z^n)\mathbf{F}$, where $\mathbf{F}$ is an $(n+1) \times (n+1)$ matrix. Since $\text{rank}(z\mathbf{C} - \mathbf{D}) = n$, all the solutions to equation (2.36) form a one-dimensional linear space. Thus, all solutions must be linearly dependent on $\bar{\mathbf{r}}(z)$. ■

Based on Corollary B.1, we can easily derive that each $\mathbf{V}_1$ satisfying Eq. (2.10) is related to $\mathbf{V}_1^*$ through a transform equation, as defined in Lemma 2.6.3.

Proof of Lemma 2.6.3. Let $\mathbf{r}_i$ be the ith row of $\mathbf{V}_1$, which satisfies Eq. (2.10). According to Corollary B.1, $\mathbf{r}_i$ must have the form $f_i(\eta(\mathbf{x}^i))(1, \eta(\mathbf{x}^i), \cdots, \eta^n(\mathbf{x}^i))\mathbf{F}$, where $f_i(z)$ is a non-zero rational function in $\mathbb{F}_{2^m}(\xi)(z)$. Hence, $\mathbf{V}_1$ can be written as $\mathbf{G}\mathbf{V}_1^*\mathbf{F}$. Moreover, Eq. (2.36) can be rewritten as follows:

$$(z, z^2, \cdots, z^{n+1})\mathbf{FC} = (1, z, \cdots, z^n)\mathbf{FBA}$$

The right side of the above equation contains no z^{n+1}, and thus the $(n+1)$th row of $\mathbf{FC}$ must be zero. Similarly, there is no constant term on the left side of the above equation, implying that the 1st row of $\mathbf{FBA}$ is zero. ■

In the followings, we will prove the necessity of the conditions in Theorem 2.5.1. Assume a coupling relation $p_i(\mathbf{x}) = \frac{f(\eta(\mathbf{x}))}{g(\eta(\mathbf{x}))} \in \mathcal{S}_i'$ is present in the network. Without loss of generality, assume $f(z) = \sum_{k=0}^{p} a_k z^k$ and $g(z) = \sum_{k=0}^{q} b_k z^k$, where $a_p \neq 0$ and $b_q \neq 0$. We'll prove that it is impossible for ω_i to asymptotically achieve one half rate by using any PBNA. We only consider the case $i = 1$. The other cases $i = 2, 3$ can be proved similarly, and are omitted.

Consider a PBNA $\lambda = (\xi, \mathbf{V}_i : 1 \leq i \leq 3)$ with $2n + s$ symbol extensions, where $n > \max\{p, q\} + 1$. According to Corollary B.1, s must equal 1, and thus $\mathbf{V}_1$ is a $(2n+1) \times (n+1)$ matrix. By Lemma 2.6.3, $\mathbf{V}_1 = \mathbf{G}\mathbf{V}_1^*\mathbf{F}$, where $\mathbf{F}$ is an $(n+1) \times (n+1)$ invertible matrix.

The jth row of $\mathbf{V}_1$ is $\mathbf{r}_j = f_j(\eta(\mathbf{x}^{(j)}))(1\ \eta(\mathbf{x}^{(j)})\ \cdots\ \eta^n(\mathbf{x}^{(j)}))\mathbf{F}$. Since the $(n+1)$th row of $\mathbf{FC}$ is zero, we have

$$\mathbf{r}_j\mathbf{C} = f_j(\eta(\mathbf{x}^{(j)}))(1\ \eta(\mathbf{x}^{(j)})\ \cdots\ \eta^{n-1}(\mathbf{x}^{(j)}))\mathbf{H} \tag{B.1}$$

where $\mathbf{H}$ consists of the top n rows of $\mathbf{FC}$ and $\text{rank}(\mathbf{H}) = n$. For $0 \le l \le n-p-1$, define the following vector:

$$\mathbf{a}_l = (\overbrace{0\ \cdots\ 0}^{l}\ a_0\ \cdots\ a_p\ \overbrace{0\ \cdots\ 0}^{n-p-l})^T$$
$$\mathbf{b}_l = (\overbrace{0\ \cdots\ 0}^{l}\ b_0\ \cdots\ b_q\ \overbrace{0\ \cdots\ 0}^{n-p-l-1})^T$$

It follows that

$$f(\eta(\mathbf{x}^{(j)}))\eta^l(\mathbf{x}^{(j)}) = (1\ \eta(\mathbf{x}^{(j)})\ \cdots\ \eta^n(\mathbf{x}^{(j)}))\mathbf{a}_l \tag{B.2}$$
$$g(\eta(\mathbf{x}^{(j)}))\eta^l(\mathbf{x}^{(j)}) = (1\ \eta(\mathbf{x}^{(j)})\ \cdots\ \eta^{n-1}(\mathbf{x}^{(j)}))\mathbf{b}_l \tag{B.3}$$

Define $\mathbf{a}'_l = \mathbf{F}^{-1}\mathbf{a}_l$ and $\mathbf{b}'_l = \mathbf{H}^{-1}\mathbf{b}_l$. We can derive:

$$\begin{aligned}
\mathbf{r}_j\mathbf{a}'_l &= f_j(\eta(\mathbf{x}^{(j)}))(1\ \eta(\mathbf{x}^{(j)})\ \cdots\ \eta^n(\mathbf{x}^{(j)}))\mathbf{F}\mathbf{a}'_l \\
&= f_j(\eta(\mathbf{x}^{(j)}))(1\ \eta(\mathbf{x}^{(j)})\ \cdots\ \eta^n(\mathbf{x}^{(j)}))\mathbf{a}_l \\
&\overset{(a)}{=} f_j(\eta(\mathbf{x}^{(j)}))f(\eta(\mathbf{x}^{(j)}))\eta^l(\mathbf{x}^{(j)}) \\
&\overset{(b)}{=} f_j(\eta(\mathbf{x}^{(j)}))p_1(\mathbf{x}^{(j)})g(\eta(\mathbf{x}^{(j)}))\eta^l(\mathbf{x}^{(j)}) \\
&\overset{(c)}{=} p_i(\mathbf{x}^{(j)})f_j(\eta(\mathbf{x}^{(j)}))(1\ \eta(\mathbf{x}^{(j)})\ \cdots\ \eta^{n-1}(\mathbf{x}^{(j)}))\mathbf{b}_l \\
&= p_i(\mathbf{x}^{(j)})f_j(\mathbf{x}^{(j)}))(1\ \eta(\mathbf{x}^{(j)})\ \cdots\ \eta^{n-1}(\mathbf{x}^{(j)}))\mathbf{H}\mathbf{b}'_l \\
&\overset{(d)}{=} p_i(\mathbf{x}^{(j)})\mathbf{r}_j\mathbf{C}\mathbf{b}'_l
\end{aligned}$$

where (a) follows from Eq. (B.2); (b) follows because of the following equation:

$$p_1(\mathbf{x}^{(j)}) = \frac{f(\eta(\mathbf{x}^{(j)}))}{g(\eta(\mathbf{x}^{(j)}))} = \frac{f(\eta(\mathbf{x}^{(j)}))\eta^l(\mathbf{x}^{(j)})}{g(\eta(\mathbf{x}^{(j)}))\eta^l(\mathbf{x}^{(j)})}$$

(c) is due to Eq. (B.3); (d) follows from Eq. (B.1). Let $\mathbf{H}_1 = (\mathbf{V}_1 \quad \mathbf{P}_1\mathbf{V}_1\mathbf{C})$ denote the matrix in the reformulated rank condition $\mathscr{B}_1'$. Since $\mathbf{a}_0, \cdots, \mathbf{a}_{n-p-1}$ are linearly independent, the above equation means that there are at most $n+1-(n-p) = p+1$ columns in $\mathbf{V}_1$ that are linearly independent of the columns in $\mathbf{P}_1\mathbf{V}_1\mathbf{C}$. Therefore, d_1 can decode at most $p+1$ source symbols. This means that it is impossible for ω_1 to achieve one half rate by using any PBNA. ■

C Proofs of Interpretation of Coupling Relations

C.1 $\eta(\mathbf{x}) = 1$

First, note that $\eta(\mathbf{x})$ can be rewritten as a ratio of two rational functions $\eta(\mathbf{x}) = \frac{f_{213}(\mathbf{x})}{f_{312}(\mathbf{x})}$, where $f_{ijk}(\mathbf{x}) \triangleq \frac{m_{ij}(\mathbf{x})m_{jk}(\mathbf{x})}{m_{ik}(\mathbf{x})}$. Hence, in order to interpret $\eta(\mathbf{x}) = 1$, we first study the properties of $f_{ijk}(\mathbf{x})$.

The following lemma is to be used to derive the general structure of $f_{ijk}(\mathbf{x})$. Basically, it provides an easy method to calculate the greatest common divisor of two transfer functions with one common starting edge or ending edge.

Lemma C.1. The following statements hold:

1. For $e_1, e_2, e_3 \in E'$ such that e_2, e_3 are both downstream of e_1. Let e be the last edge of the topological ordering of the edges in $\mathcal{C}_{e_1e_2} \cap \mathcal{C}_{e_1e_3}$. Then $m_{e_1e}(\mathbf{x}) = \gcd(m_{e_1e_2}(\mathbf{x}), m_{e_1e_3}(\mathbf{x}))$.

2. For $e_1, e_2, e_3 \in E'$ such that e_1, e_2 are both upstream of e_3. Let e be the first edge of the

topological ordering of the edges in $\mathcal{C}_{e_1e_3} \cap \mathcal{C}_{e_2e_3}$. Then $m_{ee_3}(\mathbf{x}) = \gcd(m_{e_1e_3}(\mathbf{x}), m_{e_2e_3}(\mathbf{x}))$.

Proof. First, consider the first statement. By Lemma A.3, the following equations hold: $m_{e_1e_2}(\mathbf{x}) = m_{e_1e}(\mathbf{x})m_{ee_2}(\mathbf{x})$ and $m_{e_1e_3}(\mathbf{x}) = m_{e_1e}(\mathbf{x})m_{ee_3}(\mathbf{x})$. Thus $m_{e_1e}(\mathbf{x}) \mid \gcd(m_{e_1e_2}(\mathbf{x}), m_{e_1e_3}(\mathbf{x}))$. Assume $\gcd(m_{ee_2}(\mathbf{x}), m_{ee_3}(\mathbf{x})) \neq 1$. By Lemma A.3, there exists bottlenecks e_4, e_5 such that $m_{e_4e_5}(\mathbf{x}) \mid \gcd(m_{ee_2}(\mathbf{x}), m_{ee_3}(\mathbf{x}))$. Clearly, $e_5 \in \mathcal{C}_{e_1e_2} \cap \mathcal{C}_{e_1e_3}$ and e_5 is downstream of e, which contradicts that e is the last edge of the topological ordering of $\mathcal{C}_{e_1e_2} \cap \mathcal{C}_{e_1e_3}$. Hence, we have proved that $\gcd(m_{ee_2}(\mathbf{x}), m_{ee_3}(\mathbf{x})) = 1$, which in turn implies that $m_{e_1e}(\mathbf{x}) = \gcd(m_{e_1e_2}(\mathbf{x}), m_{e_1e_3}(\mathbf{x}))$. Similarly, we can prove the other statement. ■

Using the above lemma, $f_{ijk}(\mathbf{x})$ can be reformulated as a fraction of two coprime polynomials, as shown below.

Corollary C.1. $f_{ijk}(\mathbf{x})$ can be formulated as

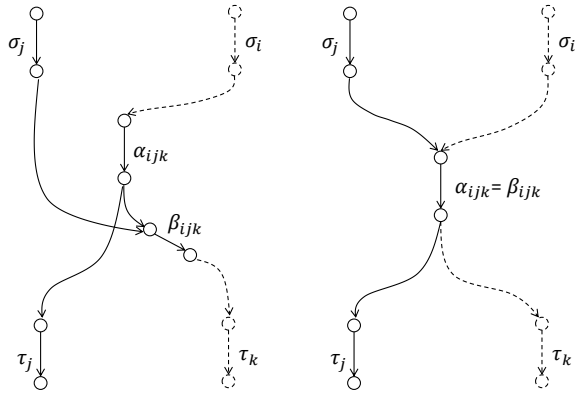
$$f_{ijk}(\mathbf{x}) = \frac{m_{\sigma_j,\beta_{ijk}}(\mathbf{x})m_{\alpha_{ijk},\tau_j}(\mathbf{x})}{m_{\alpha_{ijk},\beta_{ijk}}(\mathbf{x})} \tag{C.4}$$

where $\gcd(m_{\sigma_j,\beta_{ijk}}(\mathbf{x})m_{\alpha_{ijk},\tau_j}(\mathbf{x}), m_{\alpha_{ijk},\beta_{ijk}}(\mathbf{x})) = 1$.

Proof. $f_{ijk}(\mathbf{x})$ can be calculated as

$$\begin{aligned} f_{ijk}(\mathbf{x}) &= \frac{m_{\sigma_i,\alpha_{ijk}}(\mathbf{x})m_{\alpha_{ijk},\tau_j}(\mathbf{x})m_{jk}(\mathbf{x})}{m_{\sigma_i,\alpha_{ijk}}(\mathbf{x})m_{\alpha_{ijk},\tau_k}(\mathbf{x})} \\ &= \frac{m_{\alpha_{ijk},\tau_j}(\mathbf{x})m_{jk}(\mathbf{x})}{m_{\alpha_{ijk},\tau_k}(\mathbf{x})} \\ &= \frac{m_{\alpha_{ijk},\tau_j}(\mathbf{x})m_{\sigma_j,\beta_{ijk}}(\mathbf{x})m_{\beta_{ijk},\tau_k}(\mathbf{x})}{m_{\alpha_{ijk},\beta_{ijk}}(\mathbf{x})m_{\beta_{ijk},\tau_k}(\mathbf{x})} \\ &= \frac{m_{\sigma_j,\beta_{ijk}}(\mathbf{x})m_{\alpha_{ijk},\tau_j}(\mathbf{x})}{m_{\alpha_{ijk},\beta_{ijk}}(\mathbf{x})} \end{aligned}$$

By Lemma , $\gcd(m_{\alpha_{ijk},\tau_k}(\mathbf{x}), m_{\alpha_{ijk},\tau_j}(\mathbf{x})) = 1$ and thus $\gcd(m_{\alpha_{ijk},\beta_{ijk}}(\mathbf{x}), m_{\alpha_{ijk},\tau_j}(\mathbf{x})) = 1$. Meanwhile, $\gcd(m_{\alpha_{ijk},\beta_{ijk}}(\mathbf{x}), m_{\sigma_j,\beta_{ijk}}(\mathbf{x})) = 1$. Hence, we must have $\gcd(m_{\sigma_j,\beta_{ijk}}(\mathbf{x})\, m_{\alpha_{ijk},\tau_j}(\mathbf{x}), m_{\alpha_{ijk},\beta_{ijk}}(\mathbf{x})$



(a) $\alpha_{ijk} \neq \beta_{ijk}$ (b) $\alpha_{ijk} = \beta_{ijk}$

Figure C.3: The structure of $f_{ijk}(\mathbf{x})$ can be classified into two types: 1) $\alpha_{ijk} \neq \beta_{ijk}$ such that $f_{ijk}(\mathbf{x})$ is a rational function with non-constant denominator; 2) $\alpha_{ijk} = \beta_{ijk}$ such that $f_{ijk}(\mathbf{x})$ is a polynomial.

1. ■

According to Corollary C.1, the structure of $f_{ijk}(\mathbf{x})$ must fall into one of the two types, as shown in Fig. C.3. In Fig. C.3a, $\alpha_{ijk} \neq \beta_{ijk}$ and $f_{ijk}(\mathbf{x})$ is a rational function, the denominator of which is a non-constant polynomial $m_{\alpha_{ijk},\beta_{ijk}}(\mathbf{x})$. On the other hand, when $\alpha_{ijk} \in \mathcal{C}_{jk}$ and thus $\alpha_{ijk} = \beta_{ijk}$, as shown in Fig. C.3b, $f_{ijk}(\mathbf{x})$ becomes a polynomial $m_{\sigma_j,\alpha_{ijk}}(\mathbf{x})m_{\alpha_{ijk},\tau_j}(\mathbf{x})$.

Moreover, using Corollary C.1, we can easily check whether two $f_{ijk}(\mathbf{x})$'s are equivalent, as shown in the next corollary. It is easy to see that Theorem 2.5.3 is just a special case of this corollary.

Corollary C.2. Assume $i, j, k, i', k' \in \{1, 2, 3\}$ such that $i \neq j, j \neq k$ and $i' \neq j, j \neq k'$. $f_{ijk}(\mathbf{x}) = f_{i'jk'}(\mathbf{x})$ if and only if $\alpha_{ijk} = \alpha_{i'jk'}$ and $\beta_{ijk} = \beta_{i'jk'}$.

Proof. By Corollary C.1, if $\alpha_{ijk} = \alpha_{i'jk'}$ and $\beta_{ijk} = \beta_{i'jk'}$, we must have $f_{ijk}(\mathbf{x}) = f_{i'jk'}(\mathbf{x})$. Conversely, if $f_{ijk}(\mathbf{x}) = f_{i'jk'}(\mathbf{x})$, $m_{\alpha_{ijk},\beta_{ijk}}(\mathbf{x}) = m_{\alpha_{i'jk'},\beta_{i'jk'}}(\mathbf{x})$. Thus $\alpha_{ijk} = \alpha_{i'jk'}$ and $\beta_{ijk} = \beta_{i'jk'}$ by Lemma A.2. ■

C.2 $p_i(\mathbf{x}) = 1$ and $p_i(\mathbf{x}) = \eta(\mathbf{x})$

Using Lemma A.1, we can easily prove Theorem 2.5.4, as shown below.

Proof of Theorem 2.5.4. Apparently, by Lemma A.1 and the definition of $p_1(\mathbf{x})$, $p_1(\mathbf{x}) = 1$ if and only if the minimum cut separating σ_1, σ_2 from τ_1 and τ_3 is one, i.e., $C_{12,13} = 1$. In order to interpret $p_1(\mathbf{x}) = \eta(\mathbf{x})$, we consider $q_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{p_1(\mathbf{x})} = \frac{m_{11}(\mathbf{x})m_{32}(\mathbf{x})}{m_{12}(\mathbf{x})m_{31}(\mathbf{x})}$. Hence $p_1(\mathbf{x}) = \eta(\mathbf{x})$ is equivalent to $q_1(\mathbf{x}) = 1$. Similarly, using Lemma A.1, it is easy to see that $p_1(\mathbf{x}) = \eta(\mathbf{x})$ if and only if the minimum cut separating σ_1, σ_3 from τ_1, τ_2 is one, i.e., $C_{13,12} = 1$. ■

C.3 $p_1(\mathbf{x}) = \frac{\eta(\mathbf{x})}{1+\eta(\mathbf{x})}$ and $p_2(\mathbf{x}), p_3(\mathbf{x}) = 1 + \eta(\mathbf{x})$

Note that the three coupling relations can be respectively reformulated in terms of $f_{ijk}(\mathbf{x})$ as follows:

$$m_{11}(\mathbf{x}) = f_{312}(\mathbf{x}) + f_{213}(\mathbf{x})$$
$$m_{22}(\mathbf{x}) = f_{123}(\mathbf{x}) + f_{321}(\mathbf{x})$$
$$m_{33}(\mathbf{x}) = f_{231}(\mathbf{x}) + f_{132}(\mathbf{x})$$

Thus, as shown below, the three coupling relations can also be interpreted by using the properties of $f_{ijk}(\mathbf{x})$.

Proof of Theorem 2.5.5. We only prove statement 1). The other statements can be proved similarly. First, we prove the "if" part. Due to $\alpha_{312} \in \mathcal{C}_{12}$ and $\alpha_{213} \in \mathcal{C}_{13}$, $f_{312}(\mathbf{x}) = m_{\sigma_1,\alpha_{312}}(\mathbf{x})m_{\alpha_{312},\tau_1}(\mathbf{x})$ and $f_{213}(\mathbf{x}) = m_{\sigma_1,\alpha_{213}}(\mathbf{x})m_{\alpha_{213},\tau_1}(\mathbf{x})$. Hence, $f_{312}(\mathbf{x}) + f_{213}(\mathbf{x}) = m_{\sigma_1,\alpha_{312}}(\mathbf{x})m_{\alpha_{312},\tau_1}(\mathbf{x}) + m_{\sigma_1,\alpha_{213}}(\mathbf{x})m_{\alpha_{213},\tau_1}(\mathbf{x})$. On the other hand, because $\alpha_{312}||\alpha_{213}$ and $\{\alpha_{312}, \alpha_{213}\}$ forms a cut which separates σ_1 from τ_1, $m_{11}(\mathbf{x}) = m_{\sigma_1,\alpha_{312}}(\mathbf{x})m_{\alpha_{312},\tau_1}(\mathbf{x}) + m_{\sigma_1,\alpha_{213}}(\mathbf{x})m_{\alpha_{213},\tau_1}(\mathbf{x})$ by Lemma A.4. Therefore, $m_{11}(\mathbf{x}) = f_{312}(\mathbf{x}) + f_{213}(\mathbf{x})$.

Next we prove the "only if" part. Assume $m_{11}(\mathbf{x}) = f_{312}(\mathbf{x}) + f_{213}(\mathbf{x})$. If $\alpha_{312} \notin \mathcal{C}_{12}$ but $\alpha_{213} \in \mathcal{C}_{13}$, $f_{312}(\mathbf{x})$ is a rational function whose denominator is a non-constant polynomial, while $f_{213}(\mathbf{x})$ is a polynomial. Hence $f_{312}(\mathbf{x}) + f_{213}(\mathbf{x})$ must be a rational function with non-constant denominator, and thus $m_{11}(\mathbf{x}) \neq f_{312}(\mathbf{x}) + f_{213}(\mathbf{x})$. Similarly, if $\alpha_{312} \in \mathcal{C}_{12}$ but $\alpha_{213} \notin \mathcal{C}_{13}$, we can also prove that $m_{11}(\mathbf{x}) \neq f_{312}(\mathbf{x}) + f_{213}(\mathbf{x})$.

Now assume $\alpha_{312} \notin \mathcal{C}_{12}$ and $\alpha_{213} \notin \mathcal{C}_{13}$. It follows that $f_{312}(\mathbf{x}) = \frac{m_{\sigma_1,\beta_{312}}(\mathbf{x}) m_{\alpha_{312},\tau_1}(\mathbf{x})}{m_{\alpha_{312},\beta_{312}}(\mathbf{x})}$ and $f_{213}(\mathbf{x}) = \frac{m_{\sigma_1,\beta_{213}}(\mathbf{x}) m_{\alpha_{213},\tau_1}(\mathbf{x})}{m_{\alpha_{213},\beta_{213}}(\mathbf{x})}$. Because $\eta(\mathbf{x}) \neq 1$, we have $f_{312}(\mathbf{x}) \neq f_{213}(\mathbf{x})$, which indicates that $\alpha_{312} \neq \alpha_{213}$ or $\beta_{312} \neq \beta_{213}$ by Corollary C.2, and $m_{\alpha_{312},\beta_{312}}(\mathbf{x}) \neq m_{\alpha_{213},\beta_{213}}(\mathbf{x})$. Therefore, by Lemma A.3, one of the following cases must hold: 1) There exists an irreducible polynomial $m_{ee'}(\mathbf{x})$ such that $m_{ee'}(\mathbf{x}) \mid m_{\alpha_{312},\beta_{312}}(\mathbf{x})$ but $m_{ee'}(\mathbf{x}) \nmid m_{\alpha_{213},\beta_{213}}(\mathbf{x})$; 2) there exists an irreducible polynomial $m_{ee'}(\mathbf{x})$ such that $m_{ee'}(\mathbf{x}) \nmid m_{\alpha_{312},\beta_{312}}(\mathbf{x})$ but $m_{ee'}(\mathbf{x}) \mid m_{\alpha_{213},\beta_{213}}(\mathbf{x})$.

Consider case 1). We use $\mathrm{lcm}(\alpha(\mathbf{x}), \beta(\mathbf{x}))$ to denote the least common multiple of two polynomials $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$. Define the following polynomials:

$$f(\mathbf{x}) = \mathrm{lcm}(m_{\alpha_{312},\beta_{312}}(\mathbf{x}), m_{\alpha_{213},\beta_{213}}(\mathbf{x}))$$
$$f_1(\mathbf{x}) = f(\mathbf{x})/m_{\alpha_{312},\beta_{312}}(\mathbf{x}) \quad f_2(\mathbf{x}) = f(\mathbf{x})/m_{\alpha_{213},\beta_{213}}(\mathbf{x})$$

Hence, we have $m_{ee'}(\mathbf{x}) \nmid f_1(\mathbf{x})$, $m_{ee'}(\mathbf{x}) \mid f_2(\mathbf{x})$, and the following equation holds:

$$\begin{aligned} & f_{312}(\mathbf{x}) + f_{213}(\mathbf{x}) \\ =& \frac{m_{\sigma_1,\beta_{312}}(\mathbf{x}) m_{\alpha_{312},\tau_1}(\mathbf{x}) f_1(\mathbf{x}) + m_{\sigma_1,\beta_{213}}(\mathbf{x}) m_{\alpha_{213},\tau_1}(\mathbf{x}) f_2(\mathbf{x})}{f(\mathbf{x})} \end{aligned}$$

Moreover, due to $\gcd(m_{\alpha_{312},\beta_{312}}(\mathbf{x}),\ m_{\sigma_1,\beta_{312}}(\mathbf{x}) m_{\alpha_{312},\tau_1}(\mathbf{x})) = 1$, it follows that $m_{ee'}(\mathbf{x}) \nmid m_{\sigma_1,\beta_{312}}(\mathbf{x}) m_{\alpha_{312},\tau_1}(\mathbf{x})$. This implies that:

$$m_{ee'}(\mathbf{x}) \nmid m_{\sigma_1,\beta_{312}}(\mathbf{x}) m_{\alpha_{312},\tau_1}(\mathbf{x}) f_1(\mathbf{x}) + m_{\sigma_1,\beta_{213}}(\mathbf{x}) m_{\alpha_{213},\tau_1}(\mathbf{x}) f_2(\mathbf{x})$$

However, $m_{ee'}(\mathbf{x}) \mid f(\mathbf{x})$. This indicates that $f_{312}(\mathbf{x}) + f_{213}(\mathbf{x})$ is a rational function with non-constant denominator. Thus $m_{11}(\mathbf{x}) \neq f_{312}(\mathbf{x}) + f_{213}(\mathbf{x})$. Similarly, for case 2), we can also prove that $m_{11}(\mathbf{x}) \neq f_{312}(\mathbf{x}) + f_{213}(\mathbf{x})$.

Thus, we have proved that $\alpha_{312} \in \mathcal{C}_{12}$ and $\alpha_{213} \in \mathcal{C}_{13}$. It immediately follows that $m_{11}(\mathbf{x}) = m_{\sigma_1,\alpha_{312}}(\mathbf{x})m_{\alpha_{312},\tau_1}(\mathbf{x}) + m_{\sigma_1,\alpha_{213}}(\mathbf{x})m_{\alpha_{213},\tau_1}(\mathbf{x})$. Hence each path P in $\mathcal{P}_{\sigma_1\tau_1}$ either pass through α_{312} or α_{213}, implying that $\{\alpha_{312}, \alpha_{213}\}$ forms a cut separating σ_1 from τ_1. Moreover, according to Lemma A.4, $\alpha_{312}||\alpha_{213}$. ■

D Proofs of Lemmas on Multivariate Polynomials

In this section, we present the proof of Lemma B.1. We first prove that Lemma B.1 holds for the case where $s(\mathbf{x})$ and $t(\mathbf{x})$ are both univariate polynomials. In order to extend this result to multivariate polynomials, we employ a simple idea that each multivariate polynomial can be viewed as an equivalent univariate polynomial on a field of rational functions. Specifically, we prove that the problem of checking if two multivariate polynomials are co-prime is equivalent to checking if their equivalent univariate polynomials are co-prime. Finally, based on this result, we prove that Lemma B.1 also holds for the multivariate case.

D.1 The Univariate Case

In the following lemma, we show that Lemma B.1 holds for the univariate case.

Lemma D.1. Let $\mathbb{F}$ be a field, and z, y are two variables. Consider four non-zero polynomials $f(z), g(z) \in \mathbb{F}[z]$ and $s(y), t(y) \in \mathbb{F}[y]$, such that $\gcd(f(z), g(z)) = 1$ and $\gcd(s(y), t(y)) = 1$. Denote $d = \max\{d_f, d_g\}$. Define two polynomials $\alpha(y) = f(\frac{s(y)}{t(y)})t^d(y)$ and $\beta(y) = g(\frac{s(y)}{t(y)})t^d(y)$. Then $\gcd(\alpha(y), \beta(y)) = 1$.

Proof. Assume $w(x) = \gcd(\alpha(x), \beta(x))$ is non-trivial. Thus we can find an extension field $\bar{\mathbb{F}}$ of $\mathbb{F}$ such that there exists $x_0 \in \bar{\mathbb{F}}$ which satisfies $w(x_0) = 0$ and hence $\alpha(x_0) = \beta(x_0) = 0$. In the rest of this proof, we restrict our discussion in $\bar{\mathbb{F}}$. Note that $\gcd(f(z), g(z)) = 1$ and $\gcd(s(x), t(x)) = 1$ also hold for $\bar{\mathbb{F}}$. Assume $t(x_0) = 0$ and thus $x - x_0 \mid t(x)$. Since $\gcd(s(x), t(x)) = 1$, it follows that $x - x_0 \nmid s(x)$ and thus $s(x_0) \neq 0$. Hence, either $\alpha(x_0) \neq 0$ or $\beta(x_0) \neq 0$, contradicting that $\alpha(x_0), \beta(x_0)$ are both zeros. Hence, we have proved that $t(x_0) \neq 0$. Then we have $f\left(\frac{s(x_0)}{t(x_0)}\right) = \frac{\alpha(x_0)}{t^d(x_0)} = 0$ and $g\left(\frac{s(x_0)}{t(x_0)}\right) = \frac{\beta(x_0)}{t^d(x_0)} = 0$, which implies that $z - \frac{s(x_0)}{t(x_0)}$ is a common divisor of $f(z)$ and $g(z)$, contradicting $\gcd(f(z), g(z)) = 1$. Thus, we have proved that $\gcd(\alpha(y), \beta(y)) = 1$. ■

D.2 Viewing Multivariate as Univariate

In order to extend Lemma D.1 to the multivariate case, we first show that each multivariate polynomial can be viewed as an equivalent univariate polynomial on a field of rational functions. Let $\mathbf{y} = (y_1, y_2, \cdots, y_k)$ be a vector of variables. For any $i \in \{1, 2, \cdots, k\}$, define $\mathbf{y}_i = (y_1, \cdots, y_{i-1}, y_{i+1}, \cdots, y_k)$, i.e., the vector consisting of all variables in $\mathbf{y}$ other than y_i. Note that any polynomial $f(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$ can be formulated as $f(\mathbf{y}) = f_0(\mathbf{y}_i) + f_1(\mathbf{y}_i)y_i + \cdots + f_p(\mathbf{y}_i)y_i^p$, where each $f_j(\mathbf{y}_i)$ is a polynomial in $\mathbb{F}[\mathbf{y}_i]$. Because $\mathbb{F}[\mathbf{y}_i]$ is a subset of $\mathbb{F}(\mathbf{y}_i)$, $f(\mathbf{y})$ can also be viewed as a univariate polynomial in $\mathbb{F}(\mathbf{y}_i)[y_i]$. We use $f(y_i)$ to denote $f(\mathbf{y})$'s equivalent counterpart in $\mathbb{F}(\mathbf{y}_i)[y_i]$. To differentiate these two concepts, we reserve the notations, such as "|", "gcd" and "lcm" for field $\mathbb{F}$, and append "1" as a subscript to these notations to suggest they are specific to field $\mathbb{F}(\mathbf{y}_i)$. For example, for $f(\mathbf{y}), g(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$ and $u(y_i), v(y_i) \in \mathbb{F}(\mathbf{y}_i)[y_i]$, $g(\mathbf{y}) \mid f(\mathbf{y})$ means that there exists $h(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$ such that $f(\mathbf{y}) = h(\mathbf{y})g(\mathbf{y})$, and $u(y_i) \mid_1 v(y_i)$ means that there exists $w(y_i) \in \mathbb{F}[\mathbf{y}_i](y_i)$ such that $v(y_i) = w(y_i)u(y_i)$.

Lemma D.2. Assume $g(\mathbf{y}_i) \in \mathbb{F}[\mathbf{y}_i]$ and $f(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$ is of the form $f(\mathbf{y}) = \sum_{j=0}^{p} f_j(\mathbf{y}_i)y_i^j$, where $f_j(\mathbf{y}_i) \in \mathbb{F}[\mathbf{y}_i]$. Then $g(\mathbf{y}_i) \mid f(\mathbf{y})$ if and only if $g(\mathbf{y}_i) \mid f_j(\mathbf{y}_i)$ for each $j \in \{0, 1, \cdots, p\}$.

Proof. Apparently, if $g(\mathbf{y}_i) \mid f_j(\mathbf{y}_j)$ for any $j \in \{0, 1, \cdots, p\}$, $g(\mathbf{y}_i) \mid f(\mathbf{y})$. Now assume $g(\mathbf{y}_i) \mid f(\mathbf{y})$. Thus there exists $h(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$ such that $f(\mathbf{y}) = g(\mathbf{y}_i)h(\mathbf{y})$. Let $h(\mathbf{y}) = \sum_{j=0}^{p} h_j(\mathbf{y}_i)y_i^j$. Hence, it follows that $f_j(\mathbf{y}_i) = h_j(\mathbf{y}_i)g(\mathbf{y}_i)$ and thus $g(\mathbf{y}_i) \mid f_j(\mathbf{y}_i)$. ■

The following result follows immediately from Lemma D.2.

Corollary D.1. Let $g(\mathbf{y}_i)$ and $f(\mathbf{y})$ be defined as Lemma D.2. Then $\gcd(g(\mathbf{y}_i), f(\mathbf{y})) = \gcd(g(\mathbf{y}_i), f_0(\mathbf{y}_i), \cdots, f_p(\mathbf{y}_i))$.

Proof. Note that any divisor of $g(\mathbf{y}_i)$ must be a polynomial in $\mathbb{F}[\mathbf{y}_i]$. Let $d(\mathbf{y}_i) = \gcd(g(\mathbf{y}_i), f(\mathbf{y}))$ and $d'(\mathbf{y}_i) = \gcd(g(\mathbf{y}_i), f_0(\mathbf{y}_i), \cdots, f_p(\mathbf{y}_i))$. By Lemma D.2, $d(\mathbf{y}_i) \mid f_j(\mathbf{y}_i)$ for any $j \in \{0, 1, \cdots, p\}$, implying that $d(\mathbf{y}_i) \mid d'(\mathbf{y}_i)$. On the other hand, $d'(\mathbf{y}_i) \mid f(\mathbf{y})$, and thus $d'(\mathbf{y}_i) \mid d(\mathbf{y}_i)$. Hence, $d(\mathbf{y}_i) = d'(\mathbf{y}_i)$. ■

Corollary D.2. For $t \in \{1, 2, \cdots, s\}$, let $f_t(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$ be defined as $f_t(\mathbf{y}) = \sum_{j=0}^{p_t} f_{tj}(\mathbf{y}_i)y_i^j$, where $f_{tj}(\mathbf{y}_i) \in \mathbb{F}[\mathbf{y}_i]$. Let $g(\mathbf{y}_i) \in \mathbb{F}[\mathbf{y}_i]$. It follows

$$\begin{aligned}
&\gcd(g(\mathbf{y}_i), f_1(\mathbf{y}), \cdots, f_t(\mathbf{y})) \\
=&\gcd(g(\mathbf{y}_i), f_{10}(\mathbf{y}_i), \cdots, f_{1p_1}(\mathbf{y}_i), \cdots, f_{s0}(\mathbf{y}_i), \cdots, f_{sp_s}(\mathbf{y}_i))
\end{aligned}$$

Proof. We have the following equations

$$\begin{aligned}
&\gcd(g(\mathbf{y}_i), f_1(\mathbf{y}), \cdots, f_t(\mathbf{y})) = \gcd(g(\mathbf{y}_i), f_1(\mathbf{y}), \cdots, g(\mathbf{y}_i), f_t(\mathbf{y})) \\
=&\gcd(\gcd(g(\mathbf{y}_i), f_1(\mathbf{y})), \cdots, \gcd(g(\mathbf{y}_i), f_s(\mathbf{y}))) \\
=&\gcd(g(\mathbf{y}_i), f_{10}(\mathbf{y}_i), \cdots, f_{1p_1}(\mathbf{y}_i), \cdots, g(\mathbf{y}_i), f_{s0}(\mathbf{y}_i), \cdots, f_{sp_s}(\mathbf{y}_i)) \\
=&\gcd(g(\mathbf{y}_i), f_{10}(\mathbf{y}_i), \cdots, f_{1p_1}(\mathbf{y}_i), \cdots, f_{s0}(\mathbf{y}_i), \cdots, f_{sp_s}(\mathbf{y}_i))
\end{aligned}$$

■

Lemma D.3. For $t \in \{1, 2, \cdots, s\}$, let $a_t(\mathbf{y}), b_t(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$ such that $b_t(\mathbf{y}) \neq 0$ and $\gcd(a_t(\mathbf{y}), b_t(\mathbf{y})) = 1$. For $t \in \{1, 2, \cdots, s\}$, let $v_t(\mathbf{y}) = \mathrm{lcm}(b_1(\mathbf{y}), \cdots, b_t(\mathbf{y}))$. Then we have

$$\gcd\left(a_1(\mathbf{y})\frac{v_s(\mathbf{y})}{b_1(\mathbf{y})}, \cdots, a_s(\mathbf{y})\frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}, v_s(\mathbf{y})\right) = 1$$

Proof. We use induction on s to prove this lemma. Apparently, the lemma holds for $s = 1$ due to $\gcd(a_1(\mathbf{y}), b_1(\mathbf{y})) = 1$. Assume it holds for $s - 1$. Thus it follows

$$\begin{aligned}
&\gcd\left(a_1(\mathbf{y})\frac{v_s(\mathbf{y})}{b_1(\mathbf{y})}, \cdots, a_s(\mathbf{y})\frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}, v_s(\mathbf{y})\right)\\
=&\gcd\left(a_1(\mathbf{y})\frac{v_s(\mathbf{y})}{b_1(\mathbf{y})}, \cdots, a_s(\mathbf{y})\frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}, b_s(\mathbf{y})\frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}\right)\\
=&\gcd\left(a_1(\mathbf{y})\frac{v_s(\mathbf{y})}{b_1(\mathbf{y})}, \cdots, \gcd(a_s(\mathbf{y}), b_s(\mathbf{y}))\frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}\right)\\
\overset{(a)}{=}&\gcd\left(a_1(\mathbf{y})\frac{v_s(\mathbf{y})}{b_1(\mathbf{y})}, \cdots, a_{s-1}(\mathbf{y})\frac{v_s(\mathbf{y})}{b_{s-1}(\mathbf{y})}, \frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}\right)\\
\overset{(b)}{=}&\gcd\left(a_1(\mathbf{y})\frac{v_s(\mathbf{y})}{b_1(\mathbf{y})}, \cdots, a_{s-1}(\mathbf{y})\frac{v_s(\mathbf{y})}{b_{s-1}(\mathbf{y})}, \gcd\left(v_{s-1}(\mathbf{y}), \frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}\right)\right)\\
=&\gcd\left(a_1(\mathbf{y})\frac{v_s(\mathbf{y})}{b_1(\mathbf{y})}, \cdots, a_{s-1}(\mathbf{y})\frac{v_s(\mathbf{y})}{b_{s-1}(\mathbf{y})}, v_{s-1}(\mathbf{y}), \frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}\right)\\
=&\gcd\left(\frac{v_s(\mathbf{y})}{v_{s-1}(\mathbf{y})}\gcd\left(a_1(\mathbf{y})\frac{v_{s-1}(\mathbf{y})}{b_1(\mathbf{y})}, \cdots, a_{s-1}(\mathbf{y})\frac{v_{s-1}(\mathbf{y})}{b_{s-1}(\mathbf{y})}\right), v_{s-1}(\mathbf{y}), \frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}\right)\\
\overset{(c)}{=}&\gcd\left(\frac{v_s(\mathbf{y})}{v_{s-1}(\mathbf{y})}, v_{s-1}(\mathbf{y}), \frac{v_s(\mathbf{y})}{b_s(\mathbf{y})}\right)\\
\overset{(d)}{=}&\gcd\left(\frac{b_s(\mathbf{y})}{\gcd(v_{s-1}(\mathbf{y}), b_s(\mathbf{y}))}, v_{s-1}(\mathbf{y}), \frac{v_{s-1}(\mathbf{y})}{\gcd(v_{s-1}(\mathbf{y}), b_s(\mathbf{y}))}\right) = \gcd(1, v_{s-1}(\mathbf{y})) = 1
\end{aligned}$$

In the above equations, (a) is due to $\gcd(a_s(\mathbf{y}), b_s(\mathbf{y})) = 1$; (b) follows from the fact that $\frac{v_s(\mathbf{y})}{b_s(\mathbf{y})} \mid v_{s-1}(\mathbf{y})$ and thus $\frac{v_s(\mathbf{y})}{b_s(\mathbf{y})} = \gcd(v_{s-1}(\mathbf{y}), \frac{v_s(\mathbf{y})}{b_s(\mathbf{y})})$; (c) follows from the inductive assumption; (d) is due to $v_s(\mathbf{y}) = \mathrm{lcm}(v_{s-1}(\mathbf{y}), b_s(\mathbf{y})) = \frac{v_{s-1}(\mathbf{y})b_s(\mathbf{y})}{\gcd(v_{s-1}(\mathbf{y}), b_s(\mathbf{y}))}$. ■

In general, each polynomial $h(y_i) \in \mathbb{F}(\mathbf{y}_i)[y_i]$ is of the form $h(y_i) = \frac{a_0(\mathbf{y}_i)}{b_0(\mathbf{y}_i)} + \frac{a_1(\mathbf{y}_i)}{b_1(\mathbf{y}_i)}y_i + \cdots + \frac{a_p(\mathbf{y}_i)}{b_p(\mathbf{y}_i)}y_i^p$, where for each $j \in \{0, 1, \cdots, p\}$, $a_j(\mathbf{y}_i), b_j(\mathbf{y}_i) \in \mathbb{F}[\mathbf{y}_i]$, $b_j(\mathbf{y}_i) \neq 0$, $\gcd(a_j(\mathbf{y}_i), b_j(\mathbf{y}_i)) = 1$, and $a_p(\mathbf{y}_i) \neq 0$. Note that for each y_i^j which is absent in $h(y_i)$, we let $a_j(\mathbf{y}_i) = 0$ and

$b_j(\mathbf{y}_i) = 1$. Moreover, define the following polynomial $\mu_h(\mathbf{y}_i) = \text{lcm}(b_0(\mathbf{y}_i), b_1(\mathbf{y}_i), \cdots, b_p(\mathbf{y}_i))$.

Corollary D.3. For $j \in \{1, 2, \cdots, s\}$, let $f_j(y_i) \in \mathbb{F}(\mathbf{y}_i)[y_i]$. Define $v(\mathbf{y}_i) = \text{lcm}(\mu_{f_1}(\mathbf{y}_i), \cdots, \mu_{f_s}(\mathbf{y}_i))$ and $\bar{f}_j(\mathbf{y}) = v(\mathbf{y}_i) f_j(y_i)$. Thus $\gcd(v(\mathbf{y}_i), \bar{f}_1(\mathbf{y}), \cdots, \bar{f}_s(\mathbf{y})) = 1$

Proof. Assume $f_j(y_i)$ has the following form:

$$f_j(y_i) = \frac{a_{j0}(\mathbf{y}_i)}{b_{j0}(\mathbf{y}_i)} + \frac{a_{j1}(\mathbf{y}_i)}{b_{j1}(\mathbf{y}_i)} y_i + \cdots + \frac{a_{jp_j}(\mathbf{y}_i)}{b_{jp_j}(\mathbf{y}_i)} y_i^{p_j}$$

where for any $j \in \{1, 2, \cdots, s\}$ and $t \in \{0, 1, \cdots, p_j\}$, $a_{jt}(\mathbf{y}_i), b_{jt}(\mathbf{y}_i) \in \mathbb{F}[\mathbf{y}_i]$, $b_{jt}(\mathbf{y}_i) \neq 0$ and $\gcd(a_{jt}(\mathbf{y}_i), b_{jt}(\mathbf{y}_i)) = 1$. Apparently, $v(\mathbf{y}_i)$ is the least common multiple of all $b_{jt}(\mathbf{y}_i)$'s. Define $u_{jt}(\mathbf{y}_i) = \frac{v(\mathbf{y}_i)}{b_{jt}(\mathbf{y}_i)} \in \mathbb{F}[\mathbf{y}_i]$. Hence, we have $\bar{f}_j(\mathbf{y}) = \sum_{t=0}^{p_j} a_{jt}(\mathbf{y}_i) u_{jt}(\mathbf{y}_i) y_i^t$. Then it follows

$$\begin{aligned}
&\gcd(v(\mathbf{y}_i), \bar{f}_1(\mathbf{y})), \cdots, \bar{f}_s(\mathbf{y})) \\
\overset{(a)}{=}&\gcd(v(\mathbf{y}_i), a_{10}(\mathbf{y}_i)u_{10}(\mathbf{y}_i), \cdots, a_{1p_1}(\mathbf{y}_i)u_{1p_1}(\mathbf{y}_i), \cdots, \\
&\quad a_{s0}(\mathbf{y}_i)u_{s0}(\mathbf{y}_i), \cdots, a_{sp_s}(\mathbf{y}_i)u_{sp_s}(\mathbf{y}_i)) \\
\overset{(b)}{=}&1
\end{aligned}$$

where (a) is due to Corollary D.2 and (b) follows from Lemma D.3. ■

Generally, the definitions of division in $\mathbb{F}[\mathbf{y}]$ and $\mathbb{F}(\mathbf{y}_i)[y_i]$ are different. However, the following theorem reveals the two definitions are closely related.

Theorem D.1. Consider two polynomials $f(\mathbf{y}), g(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$, where $g(\mathbf{y}) \neq 0$. Then $g(\mathbf{y}) \mid f(\mathbf{y})$ if and only if $g(y_i) \mid_1 f(y_i)$ for every $i \in \{1, 2, \cdots, k\}$.

Proof. The division equation between $f(y_i)$ and $g(y_i)$ is as follows

$$f(y_i) = h_i(y_i) g(y_i) + r_i(y_i) \tag{D.5}$$

where $h_i(y_i), r_i(y_i) \in \mathbb{F}(\mathbf{y}_i)[y_i]$, and either $r_i(y_i) = 0$ or $d_{r_i} < d_g$. Due to the uniqueness of Equation (D.5), $f(\mathbf{y}) \mid g(\mathbf{y})$ immediately implies that for any $i \in \{1, 2, \cdots, k\}$, $r_i(y_i) = 0$ and thus $g(y_i) \mid_1 f(y_i)$.

Conversely, assume for every $i \in \{1, \cdots, k\}$, $g(y_i) \mid_1 f(y_i)$ and hence $r_i(y_i) = 0$. Denote $\bar{h}_i(\mathbf{y}) = \mu_{h_i}(\mathbf{y}_i)h_i(y_i)$. Clearly, $\bar{h}_i(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$. Then, the following equation holds

$$\mu_{h_i}(\mathbf{y}_i)f(\mathbf{y}) = \bar{h}_i(\mathbf{y})g(\mathbf{y})$$

By Corollary D.3, $\gcd(\mu_{h_i}(\mathbf{y}_i), \bar{h}_i(\mathbf{y})) = 1$. Thus, $\mu_{h_i}(\mathbf{y}_i) \mid g(\mathbf{y})$. Define $\bar{g}(\mathbf{y}) = \frac{g(\mathbf{y})}{\mu_{h_i}(\mathbf{y}_i)}$. By Lemma D.2, $\bar{g}(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$. Define $u(\mathbf{y}) = \frac{g(\mathbf{y})}{\gcd(f(\mathbf{y}), g(\mathbf{y}))} \in \mathbb{F}[\mathbf{y}]$. It follows that

$$\begin{aligned} u(\mathbf{y}) &= \frac{g(\mathbf{y})}{\gcd(f(\mathbf{y}), g(\mathbf{y}))} = \frac{\mu_{h_i}(\mathbf{y}_i)\bar{g}(\mathbf{y})}{\gcd(\bar{h}_i(\mathbf{y})\bar{g}(\mathbf{y}), \mu_{h_i}(\mathbf{y}_i)\bar{g}(\mathbf{y}))} \\ &= \frac{\mu_{h_i}(\mathbf{y}_i)\bar{g}(\mathbf{y})}{\bar{g}(\mathbf{y})\gcd(\bar{h}_i(\mathbf{y}), \mu_{h_i}(\mathbf{y}_i))} = \frac{\mu_{h_i}(\mathbf{y}_i)\bar{g}(\mathbf{y})}{\bar{g}(\mathbf{y})} = \mu_{h_i}(\mathbf{y}_i) \end{aligned}$$

Note that variable y_i is absent in $u(\mathbf{y})$. Because y_i can be any arbitrary variable in $\mathbf{y}$, it immediately follows that all the variables in $\mathbf{y}$ must be absent in $u(\mathbf{y})$, implying that $u(\mathbf{y})$ is a constant in $\mathbb{F}$. Hence $g(\mathbf{y}) \mid f(\mathbf{y})$. ■

Moreover, in the next theorem, we will prove that checking if two multivariate polynomials are co-prime is equivalent to checking if their equivalent univariate polynomials are co-prime.

Theorem D.2. Let $f(\mathbf{y}), g(\mathbf{y})$ be two non-zero polynomials in $\mathbb{F}[\mathbf{y}]$. Then $\gcd(f(\mathbf{y}), g(\mathbf{y})) = 1$ if and only if $\gcd_1(f(y_i), g(y_i)) = 1$ for any $i \in \{1, 2, \cdots, k\}$.

Proof. First, assume for any $i \in \{1, 2, \cdots, k\}$, $\gcd_1(f(y_i), g(y_i)) = 1$. We use contradiction to prove that $\gcd(f(\mathbf{y}), g(\mathbf{y})) = 1$. Assume $u(\mathbf{y}) = \gcd(f(\mathbf{y}), g(\mathbf{y}))$ is not constant. Let y_i be a variable which is present in $u(\mathbf{y})$. By Theorem D.1, $u(y_i) \mid_1 f(y_i)$ and $u(y_i) \mid_1 g(y_i)$, which contradicts that $\gcd_1(f(y_i), g(y_i)) = 1$.

Then, assume $\gcd(f(\mathbf{y}), g(\mathbf{y})) = 1$. We also use contradiction to prove that for any $i \in \{1, 2, \cdots, k\}$, $\gcd_1(f(y_i), g(y_i)) = 1$. Assume there exists $i \in \{1, \cdots, k\}$ such that $v(y_i) = \gcd_1(f(y_i), g(y_i))$ is non-trivial. Define $w(\mathbf{y}) = \mu_v(\mathbf{y}_i)v(y_i) \in \mathbb{F}[\mathbf{y}]$. Clearly, $w(y_i) \mid_1 f(y_i)$ and $w(y_i) \mid_1 g(y_i)$. Thus, there exists $p(y_i), q(y_i) \in \mathbb{F}(\mathbf{y}_i)[y_i]$ such that

$$f(y_i) = w(y_i)p(y_i) \qquad g(y_i) = w(y_i)q(y_i)$$

Let $s(\mathbf{y}_i) = \mathrm{lcm}(\mu_p(\mathbf{y}_i), \mu_q(\mathbf{y}_i))$. Define $\bar{p}(\mathbf{y}) = s(\mathbf{y}_i)p(y_i)$ and $\bar{q}(\mathbf{y}) = s(\mathbf{y}_i)q(y_i)$. Apparently, $\bar{p}(\mathbf{y}), \bar{q}(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$. It follows that

$$s(\mathbf{y}_i)f(\mathbf{y}) = w(\mathbf{y})\bar{p}(\mathbf{y}) \qquad s(\mathbf{y}_i)g(\mathbf{y}) = w(\mathbf{y})\bar{q}(\mathbf{y})$$

Then the following equation holds

$$s(\mathbf{y}_i)\gcd(f(\mathbf{y}), g(\mathbf{y})) = w(\mathbf{y})\gcd(\bar{p}(\mathbf{y}), \bar{q}(\mathbf{y}))$$

Due to Corollary D.3, $\gcd(s(\mathbf{y}_i), \gcd(\bar{p}(\mathbf{y}), \bar{q}(\mathbf{y}))) = \gcd(s(\mathbf{y}_i), \bar{p}(\mathbf{y}), \bar{q}(\mathbf{y})) = 1$. Hence $s(\mathbf{y}_i) \mid w(\mathbf{y})$. Let $\bar{w}(\mathbf{y}) = \frac{w(\mathbf{y})}{s(\mathbf{y}_i)}$. According to Lemma D.2, $\bar{w}(\mathbf{y})$ is a non-trivial polynomial in $\mathbb{F}[\mathbf{y}]$. Thus, $\bar{w}(\mathbf{y}) \mid \gcd(f(\mathbf{y}), g(\mathbf{y}))$, contradicting $\gcd(f(\mathbf{y}), g(\mathbf{y})) = 1$. ■

D.3 The Multivariate Case

Now, we are in the place of extending Lemma D.1 to the multivariate case.

Proof of Lemma B.1. Note that if we substitute $\mathbb{F}$ with $\mathbb{F}(\mathbf{y}_i)$ and gcd with $\gcd_1$ in Lemma D.1, the lemma also holds. Apparently, $f(z), g(z) \in \mathbb{F}(\mathbf{y}_i)[z]$. We will prove that $\gcd_1(f(z), g(z)) = 1$. By contradiction, assume $r(z) = \gcd_1(f(z), g(z)) \in \mathbb{F}(\mathbf{y}_i)[z]$ is non-trivial. Let $\bar{f}(z) = \frac{f(z)}{r(z)}$ and $\bar{g}(z) = \frac{g(z)}{r(z)}$. Clearly, $\bar{f}(z)$ and $\bar{g}(z)$ are both non-zero polynomials in $\mathbb{F}(\mathbf{y}_i)[z]$. Then we

can find an assignment to $\mathbf{y}_i$, denoted by $\mathbf{y}_i^*$, such that the coefficients of the maximum powers of z in $r(z), \bar{f}(z)$ and $\bar{g}(z)$ are all non-zeros. Let $\bar{r}(z)$ denote the univariate polynomial acquired by assigning $\mathbf{y}_i = \mathbf{y}_i^*$ to $r(z)$. Clearly, $\bar{r}(z)$ is a common divisor of $f(z)$ and $g(z)$ in $\mathbb{F}[z]$, contradicting $\gcd(f(z), g(z)) = 1$. Moreover, due to $\gcd(s(\mathbf{y}), t(\mathbf{y})) = 1$ and Theorem D.2, $\gcd_1(s(y_i), t(y_i)) = 1$. Thus, by Lemma D.1, $\gcd_1(\alpha(y_i), \beta(y_i)) = 1$. Since i can be any integer in $\{1, 2, \cdots, k\}$, it follows that $\gcd(\alpha(\mathbf{y}), \beta(\mathbf{y})) = 1$ by Theorem D.2. ■

E Proofs for Multicast-Packing Coding Scheme

E.1 Proof of Proposition 3.3.1

To prove Proposition 3.3.1, we use the general framework of network coding scheme as defined in [17]. Under a network coding scheme defined in [17], each sender s_i transmits a random variable X_i. $X_1, \cdots, X_{|\Omega|}$ are mutually independent random variables. Each edge $e \in E$ transmits a random variable U_e, which is a function of the random variables injected at tail(e), *i.e.,* , the following equations hold:

$$\begin{cases} H(U_e|X_i) = 0 & \text{if tail}(e) = s_i; \\ H(U_e|U_{e'}, e' \in E, \text{head}(e') = \text{tail}(e)) = 0 & \text{otherwise.} \end{cases}$$

Given a subset of edges $E' \subseteq E$, denote $U_{E'} = (U_e : e \in E')$. Let δ_i denote the decoding error for ω_i.

Assume a rate vector $\mathbf{R} = (R_i : 1 \leq i \leq |\Omega|)$ is achievable by network coding schemes. Then, for any $\epsilon > 0$, there exists a network coding scheme of length n such that the following

conditions are satisfied:

$$\frac{1}{n}H(U_e) \leq h(e) + \epsilon \qquad \forall e \in E \tag{E.6}$$

$$\delta_i < \epsilon \qquad \forall 1 \leq i \leq |\Omega| \tag{E.7}$$

$$\frac{1}{n}H(X_i) \geq R_i - \epsilon \qquad \forall 1 \leq i \leq |\Omega| \tag{E.8}$$

By Fano's inequality, we have:

$$H(X_i|U_{\mathrm{In}(d_i)}) \leq 1 + \delta_i H(X_i) \leq 1 + \epsilon H(X_i) \tag{E.9}$$

Proof of Proposition 3.3.1. In the following proof, for simplicity, we use $X_{i:j}$ to denote the vector $(X_i, X_{i+1}, \cdots, X_j)$. For these two examples, we assume that each unicast session can achieve a symmetrical rate R. Thus, there is a network coding scheme of length n such that (E.6)-(E.9) are satisfied. In the rest of this proof, all the random variables are defined in this network coding scheme.

We first consider the example shown in Fig. 3.1a. Clearly, $U_{(s_2,d_1)}$ is a function of X_2, $U_{(s_3,d_1)}$ is a function of X_3, and $U_{(v,d_1)}$ is a function of U_{e_1,e_2}. Thus, $U_{\mathrm{In}(d_1)}$ is a function of $U_{e_1,e_2}, X_{2:3}$. Due to (E.9), we have:

$$H(X_1|U_{e_1,e_2}, X_{2:3}) \leq H(X_1|U_{\mathrm{In}(d_1)}) \leq 1 + \epsilon H(X_1) \tag{E.10}$$

Similarly, we can derive

$$H(X_5|U_{e_1,e_2}, X_{1:4}) \leq 1 + \epsilon H(X_5) \tag{E.11}$$

Thus, the following equations hold:

$$\begin{aligned}
&H(X_{1:5}, U_{e_1,e_2}) = H(U_{e_1,e_2}) + H(X_{2:3}|U_{e_1,e_2}) + H(X_1|U_{e_1,e_2}, X_{2:3}) \\
&\quad + H(X_4|U_{e_1,e_2}, X_{1:3}) + H(X_5|U_{e_1,e_2}, X_{1:4}) \\
\overset{(a)}{\leq} &H(U_{e_1,e_2}) + H(X_{2:3}|U_{e_1,e_2}) + 1 + \epsilon H(X_1) + H(X_4|U_{e_1,e_2}, X_{1:3}) + 1 + \epsilon H(X_5) \\
= &H(U_{e_1,e_2}) + H(X_{2:3}|U_{e_1,e_2}) + H(X_4|U_{e_1,e_2}, X_{1:3}) + 2 + \epsilon(H(X_1) + H(X_5))
\end{aligned} \tag{E.12}$$

where (a) is due to (E.10) and (E.11). Meanwhile, due to $H(U_{e_1,e_2}|X_{1:5}) = 0$, we have $H(X_{1:5}, U_{e_1,e_2}) = H(X_{1:5})$. Hence, we can derive:

$$\begin{aligned}
n(2+\epsilon) \overset{(b)}{\geq} &H(U_{e_1,e_2}) \\
\overset{(c)}{\geq} &H(X_{1:5}) - H(X_{2:3}|U_{e_1,e_2}) - H(X_4|U_{e_1,e_2}, X_{1:3}) - 2 - \epsilon(H(X_1) + H(X_5)) \\
\geq &H(X_{1:5}) - H(X_{2:3}) - H(X_4) - 2 - \epsilon(H(X_1) + H(X_5)) \\
\overset{(d)}{=} &H(X_1) + H(X_5) - 2 - \epsilon(H(X_1) + H(X_5)) \\
= &(1-\epsilon)(H(X_1) + H(X_5)) - 2 \\
\overset{(e)}{\geq} &2n(1-\epsilon)(R-\epsilon) - 2
\end{aligned}$$

where (b) holds because of (E.6); (c) is due to (E.12); (d) is due to the assumption that $X_1, X_2, \cdots, X_5$ are mutually independent random variables; (e) follows from (E.8). Thus, we immediately get: $1 + \frac{\epsilon}{2} \geq (1-\epsilon)(R-\epsilon) - \frac{1}{n}$. Let $\epsilon \to 0$ and $n \to \infty$, we then get $R \leq 1$. Since the MPC shown in Fig. 3.1b achieves a symmetrical rate of 1, this establishes the optimality of MPC for this example.

Next, we consider the example shown in Fig. 3.2a. Similar to above, we can derive $H(X_1|U_{(u,v)}, X_2) \leq 1 + \epsilon H(X_1)$, $H(X_4|U_{(u,v)}, X_3) \leq 1 + \epsilon H(X_4)$, and $H(X_{1:4}, U_{(u,v)}) =$

$H(X_{1:4})$. Meanwhile, the following equations hold:

$$\begin{aligned}
&H(X_{1:4}, U_{(u,v)}) \\
=&H(U_{(u,v)}) + H(X_2|U_{(u,v)}) + H(X_1|U_{(u,v)}, X_2) \\
&+ H(X_3|U_{(u,v)}, X_{1:2}) + H(X_4|U_{(u,v)}, X_{1:3}) \\
\leq&H(U_{(u,v)}) + H(X_2|U_{(u,v)}) + H(X_3|U_{(u,v)}, X_{1:2}) + 2 + \epsilon(H(X_1) + H(X_4))
\end{aligned}$$

Hence, it follows that

$$\begin{aligned}
n(1+\epsilon) \geq&H(U_{(u,v)}) \\
\geq&H(X_{1:4}) - H(X_2|U_{(u,v)}) \\
&- H(X_3|U_{(u,v)}, X_{1:2}) - 2 - \epsilon(H(X_1) + H(X_4)) \\
\geq&H(X_{1:4}) - H(X_2) - H(X_3) - 2 - \epsilon(H(X_1) + H(X_4)) \\
=&H(X_1) + H(X_4) - 2 - \epsilon(H(X_1) + H(X_4)) \\
=&(1-\epsilon)(H(X_1) + H(X_4)) - 2 \\
\geq&2n(1-\epsilon)(R-\epsilon) - 2
\end{aligned}$$

Hence, we get: $\frac{1}{2} + \frac{\epsilon}{2} \geq (1-\epsilon)(R-\epsilon) - \frac{2}{n}$. Let $\epsilon \to 0$ and $n \to \infty$. We then get $R \leq \frac{1}{2}$. Since the MPC in Fig. 3.2b achieves a symmetrical rate $\frac{1}{2}$, this establishes the optimality of MPC for this example. ■

E.2 Proofs of Results on Flow Theory

Given a node $u \in V$, let $N^+(u)$ denote the set of downstream neighbours of u. The set of edges from u to $U \subseteq N^+(u)$ is denoted by $E^+(u, U)$. The following theorem provides an iterative approach to calculate $\text{mincut}(u, v, \mathcal{N})$. Given a set of edges $E_1 \subseteq E$, define

$c(E_1) = \sum_{e \in E_1} h(e)$. If $U = \emptyset$, we define $\text{mincut}(U, v, \mathcal{N}) = 0$.

Lemma E.1. Given two distinct nodes $u, v \in V$, let $N_1 = N^+(u) - \{v\}$. The following equation holds:

$$\text{mincut}(u, v, \mathcal{N}) = \sum_{e \in E^+(u,v)} h(e) + \min_{U \subseteq N_1} \left\{ \sum_{e \in E^+(u,U)} h(e) + \text{mincut}(N_1 - U, v, \mathcal{N}) \right\}$$

Proof. Define the following subset of N_1:

$$U_1 = \underset{U \subseteq N_1}{\text{argmin}} \left\{ \sum_{e \in E^+(u,U)} h(e) + \text{mincut}(N_1 - U, v, \mathcal{N}) \right\}$$

Denote $U_2 = N_1 - U_1$. Because the outgoing edges of u are not traversed by any path from U_2 to v, there exists a $U_2 - v$ cut-set E_1 such that $E_1 \cap N^+(u) = \emptyset$ and $c(E_1) = \text{mincut}(U_2, v, \mathcal{N})$. Clearly, $E_2 = E^+(u, v) \cup E^+(u, U_1) \cup E_1$ forms a $u - v$ cut-set, the capacity of which is

$$\begin{aligned} c(E_2) &= c(E^+(u, v)) + c(E^+(u, U_1)) + c(E_1) \\ &= \sum_{e \in E^+(u,v)} h(e) + \sum_{e \in E^+(u,U_1)} h(e) + \text{mincut}(U_2, v, \mathcal{N}) \end{aligned}$$

Consider an arbitrary $U - v$ cut-set E_3. Apparently, $E^+(u, v) \subseteq E_3$. Denote $U_3 = (E_3 - E^+(u, v)) \cap \text{Out}(u)$ and $W_3 = E_3 - (E^+(u, v) \cup U_3)$. Since the removal of W_3 will break up every path from $N_1 - U_3$ to v, W_3 is also a $(N_1 - U_3) - v$ cut-set. Hence, it follows that

$c(W_3) \geq \text{mincut}(N_1 - U_3, v, \mathcal{N})$. This indicates that:

$$\begin{aligned}
c(E_3) &= c(E^+(u,v)) + c(U_3) + c(W_3) \\
&= \sum_{e \in E^+(u,v)} h(e) + \sum_{e \in U_3} h(e) + c(W_3) \\
&\geq \sum_{e \in E^+(u,v)} h(e) + \sum_{e \in U_3} h(e) + \text{mincut}(N_1 - U_3, d) \\
&\geq \sum_{e \in E^+(u,v)} h(e) + \sum_{e \in U_1} h(e) + \text{mincut}(N_1 - U_1, d) \\
&= c(E_2)
\end{aligned}$$

Thus, we have proved that $c(E_2) = \text{mincut}(u, v, \mathcal{N})$. This completes the proof. ■

The following theorem can be seen as a generalized version of the Max-Flow and Min-cut Theorem.

Theorem E.1. Given a function $g : S \to \mathbb{R}_{\geq 0}$, there exists a $S - d$ flow f over $\mathcal{N}$ such that $\text{val}(f, v) = g(v)$ for $v \in S$ if and only if the following conditions are satisfied:

$$\sum_{v \in U} g(v) \leq \text{mincut}(U, d, \mathcal{N}) \quad \forall U \subseteq S, U \neq \emptyset$$

Proof. First, we prove the "if" part. Assume g satisfies the following condition:

$$\sum_{v \in U} g(v) \leq \text{mincut}(U, d, \mathcal{N}) \quad \forall U \subseteq S, U \neq \emptyset$$

We add a node s' and connect it to each node v in U via an directed edge (s', v) with capacity $g(v)$. Let $\bar{\mathcal{N}}$ denote this network. Clearly, $\text{Out}(s')$ forms a $s' - d$ cut-set and

$c(\text{Out}(s')) = \sum_{v\in S} g(v)$. According to Lemma E.1, the following equation holds:

$$\begin{aligned}\text{mincut}(s', d, \bar{\mathcal{N}}) &= \min_{U\subseteq S}\left\{\sum_{v\in U} g(v) + \text{mincut}(S-U, d, \bar{\mathcal{N}})\right\} \\ &= \min_{U\subseteq S}\left\{\sum_{v\in U} g(v) + \text{mincut}(S-U, d, \mathcal{N})\right\}\end{aligned}$$

For $U \subseteq S$, we have

$$\begin{aligned}&\sum_{v\in U} g(v) + \text{mincut}(S-U, d, \mathcal{N}) \\ \geq &\sum_{v\in U} g(v) + \sum_{v\in S-U} g(v) = \sum_{v\in S} g(v) = c(\text{Out}(s'))\end{aligned}$$

This indicates that $c(\text{Out}(s')) = \text{mincut}(s', d, \bar{\mathcal{N}})$. According to the Max-flow and Min-cut Theorem, there is a $s'-d$ flow f_1 over $\bar{\mathcal{N}}$ such that $\text{val}(f_1, s') = c(\text{Out}(s')) = \sum_{v\in U} g(v)$. Clearly, $f_1(s', v) = g(v)$ for each $v \in S$. We then construct a $S-d$ flow f over $\mathcal{N}$ simply by setting $f(e) = f_1(e)$ for each $e \in E$. Apparently, $\text{val}(f, v) = g(v)$ for each $v \in S$.

We next prove the "only if" part. Assume there is a $S-d$ flow f such that $\text{val}(f, v) = g(v)$ for $v \in S$. Thus there exists a $U-d$ flow f_1 such that $\text{val}(f_1, v) = \text{val}(f, v) = g(v)$ for $v \in U$ and $\text{val}(f_1, v) = 0$ for $v \in S-U$. We then introduce a new node s'' and connect it to each node of U via a directed edge with infinite capacity. Let $\mathcal{N}' = (V', E', h')$ denote this new network. We then define a $s''-d$ flow f_2 over $\mathcal{N}'$ as follows:

$$f_2(e) = \begin{cases}\text{val}(f_1, \text{head}(e)) & \text{if head}(e) \in U; \\ f_1(e) & \text{otherwise.}\end{cases}$$

Due to the Max-flow and Min-cut Theorem, this suggests that

$$\begin{aligned}&\sum_{v\in U} g(v) = \sum_{v\in U} \mathrm{val}(f_1, v) = \mathrm{val}(f_2, s'')\\ \leq &\mathrm{mincut}(s'', d, \mathcal{N}') = \mathrm{mincut}(U, d, \mathcal{N})\end{aligned}$$

This completes the proof. ■

E.3 Proof of Theorem 3.3.1

Proof of Theorem 3.3.1. $1 \Rightarrow 2$: Assume $\mathbf{R} = (R_l, \cdots, R_{|\Omega|})$ can be achieved by an MPC with respect to $(\mathcal{G}, \mathcal{H})$. Hence, for any $\epsilon > 0$, there exists an MPC of length t with respect to $(\mathcal{G}, \mathcal{H})$ such that $\frac{|\mathbf{X}_l|}{t} \geq R_l - \epsilon$ for $1 \leq l \leq |\Omega|$. Consider a subset of unicast sessions $\Omega_i \in \mathcal{G}$ over the sub-capacitated network $\mathcal{N}_i$. Let U be an arbitrary non-empty subset of $S(\Omega_i)$. We add a super-sender s and link it to each $s_j \in U$ via an edge e_j of infinite capacity, which transmits a vector $\mathbf{Y}_{e_j} = \mathbf{X}_j$. Let $\bar{\mathcal{N}}_i$ denote this newly constructed network. Clearly, the encoding matrices and decoding matrices in the MPC for Ω_i also serve as a linear network coding scheme for the single unicast session (s, d_j) $(s_j \in U)$ over the network $\bar{\mathcal{N}}_i$. This unicast session achieves a rate $\sum_{s_l\in U} \frac{|\mathbf{X}_l|}{t}$. Thus, we have:

$$\sum_{s_l\in U}(R_l - \epsilon) \leq \sum_{s_l\in U} \frac{|\mathbf{X}_l|}{t} \leq \mathrm{mincut}(s, d_j, \bar{\mathcal{N}}_i) = \mathrm{mincut}(U, d_j, \mathcal{N}_i)$$

Let $\epsilon \to 0$, we then get $\sum_{s_j\in U} R_j \leq \mathrm{mincut}(U, d_j, \mathcal{N}_i)$. According to Theorem E.1, this implies that there exists a $S(\Omega_i) - d_j$ flow f_{ij} over $\mathcal{N}_i$ such that $\mathrm{val}(f_{ij}, s_l) = R_l$ for $s_l \in S(\Omega_i)$.

$2 \Rightarrow 3$: This directly follows from Theorem E.1.

$3 \Rightarrow 1$: Consider a subset of unicast sessions $\Omega_i \in \mathcal{G}$ over the sub-capacitated network $\mathcal{N}_i$. Assume that for any non-empty subset $U \in S(\Omega_i)$, $\sum_{s_l\in U} R_l \leq \mathrm{mincut}(U, d_j, \mathcal{N}_i)$. According

to Theorem 8 of [6], there is a linear network coding scheme for the multicast-scenario $\Gamma_i = \{(s_j, D(\Omega_i)) : s_j \in S(\Omega_i)\}$. Combining these linear network coding schemes, we then get an MPC for Ω with respect to $(\mathcal{G}, \mathcal{H})$. Hence $\mathbf{R}$ is achievable by the MPC. ■

F Proofs for Routing-Optimal Networks

F.1 Useful Tools

In this section, we present some useful tools to be used in the sequel.

Proposition F.1. The following equations hold:

1. $H(X|Y) = H(X|Y, f(Y))$.
2. $I(X;Y|Z) = I(X;Y|Z, f(Z))$.
3. $H(X|f(Y)) \geq H(X|Y)$.
4. $I(X;Y|Z,W) \geq I(X; f(Y,Z)|Z,W)$.

Proof. 1) The following equation holds:

$$H(X,Y,f(Y)) = H(Y) + H(X|Y) + H(f(Y)|X,Y) = H(Y) + H(X|Y) \tag{F.13}$$

Meanwhile, we have:

$$\begin{aligned} H(X,Y,f(Y)) &= H(Y) + H(f(Y)|Y) + H(X|Y,f(Y)) \\ &= H(Y) + H(X|Y,f(Y)) \end{aligned} \tag{F.14}$$

Combining Eq. (F.13) and Eq. (F.14), we have $H(X|Y) = H(X|Y, f(Y))$.

2) Due to 1), we can derive:

$$\begin{aligned} I(X;Y|Z,f(Z)) =& H(X|Z,f(Z)) - H(X|Y,Z,f(Z)) \\ =& H(X|Z) - H(X|Y,Z) = I(X;Y|Z) \end{aligned}$$

3) First, the following equalities hold:

$$H(X,Y,f(Y)) = H(f(Y)) + H(X|f(Y)) + H(Y|X,f(Y)) \tag{F.15}$$

Combining Eq. (F.13) and Eq. (F.15), we then have:

$$\begin{aligned} H(X|f(Y)) =& H(X|Y) + H(Y) - H(f(Y)) - H(Y|X,f(Y)) \\ \overset{(a)}{=}& H(X|Y) + H(Y|f(Y)) - H(Y|X,f(Y)) \\ =& H(X|Y) + I(X;Y|f(Y)) \geq H(X|Y) \end{aligned}$$

where (a) follows from the equation: $H(Y) = H(Y,f(Y)) = H(f(Y)) + H(Y|f(Y))$.

4) We have the following equations:

$$\begin{aligned} & I(X;Y|Z,W) - I(X;f(Y,Z)|Z,W) \\ =& H(X|Z,W) - H(X|Y,Z,W) - [H(X|Z,W) - H(X|f(Y,Z),Z,W)] \\ =& H(X|f(Y,Z),Z,W) - H(X|Y,Z,W) \geq 0 \end{aligned}$$

where the last inequality is due to 3) and the fact that $(f(Y,Z),Z,W)$ is a function of (Y,Z,W). ■

Proposition F.2. If $Y \to (X,W) \to Z$, then $I(X;Y|W) \geq I(X;Y|W,Z)$ and $I(X;Y|W) \geq I(Z;Y|W)$. As a special case, we have $I(X;Y|W) \geq I(X;Y|W,f(X,W))$ and $I(X;Y|W) \geq I(f(X,W);Y|W)$.

Proof. We have the following equations:

$$\begin{aligned} &I(X,Z;Y|W) = I(Z;Y|W) + I(X;Y|W,Z) \\ =&I(X;Y|W) + I(Z;Y|X,W) = I(X;Y|W) \end{aligned}$$

Thus, it must be that $I(X;Y|W) \geq I(X;Y|W,Z)$ and $I(X;Y|W) \geq I(Z;Y|W)$. Since the following chain: $Y \to (X,W) \to f(X,W)$ holds, we must have $I(X;Y|W) \geq I(X;Y|W,f(X,W))$ and $I(X;Y|W) \geq I(f(X,W);Y|W)$. ■

F.2 Proofs for Information-Distributive Networks

Proof of Lemma 4.4.1. Let S'_i denote the set consisting of the outgoing edges of $s_1, \cdots, s_{i-1}$. Since each path from s_j $(i \leq j < K)$ to d_i must pass through an edge in C_i, $S'_i \cup C_i$ forms a cut-set between $\{s_1, \cdots, s_K\}$ and d_i. Thus $U_{\mathrm{In}(d_i)}$ is a function of $U_{S'_i}, U_{C_i}$. Meanwhile, $Y_{S'_i}$ is a function of $Y_{1:i-1}$. Thus, $U_{\mathrm{In}(d_i)}$ is a function of $Y_{1:i-1}, U_{C_i}$. According to Proposition F.2, (4.24) holds. ■

Proof of Lemma 4.4.2. Let $\mathcal{T}$ be the permutation sequence as defined in Definition 4.4.2. Consider an arbitrary edge $e \in \bigcup_{i=1}^{K} C_i$. Without loss of generality, let $\mathcal{W}(e) = \{C_{n_1}, \cdots, C_{n_k}\}$, where $1 \leq n_1 < \cdots < n_k \leq K$. Then we have:

$$\sum_{1 \leq i \leq K, e \in C_i} I(Y_i; U_e | Y_{1:i-1}, U_{T_i(e)}) = \sum_{i=1}^{k} I(Y_i; U_e | Y_{1:n_i-1}, U_{T_{n_i}(e)})$$

For $k = 1$, the following equation holds $\sum_{i=1}^{k} I(Y_i; U_e | Y_{1:n_i-1}, U_{T_{n_i}(e)}) = I(Y_i : U_e |\, Y_{1:n_1-1}, U_{T_{n_1}(e)}) \leq H(U_e)$. Hence, (4.28) holds for $k = 1$. We now consider the case $k > 1$. We will prove the

following inequality holds for $1 \leq p \leq k$:

$$\sum_{i=p}^{k} I(Y_i; U_e | Y_{1:n_i-1}, U_{T_{n_i}(e)}) \leq I(Y_{n_p:n_k}; U_e | Y_{1:n_p-1}, U_{T_{n_p}(e)}) \tag{F.16}$$

Clearly, (F.16) holds trivially for $p = k$. Assume it holds for $p > 1$. We will prove it also holds for $p - 1$.

$$\begin{aligned}
&\sum_{i=p-1}^{k} I(Y_i; U_e | Y_{1:n_i-1}, U_{T_{n_i}(e)}) \\
\overset{(a)}{\leq} &I(Y_{n_p:n_k}; U_e | Y_{1:n_p-1}, U_{T_{n_p}(e)}) + I(Y_{n_{p-1}}; U_e | Y_{1:n_{p-1}-1}, U_{T_{n_{p-1}}(e)}) \\
\overset{(b)}{=} &I(Y_{n_p:n_k}; U_e | Y_{1:n_p-1}, U_{T_{n_p}(e)\cap T_{n_{p-1}}(e)}, U_{T_{n_p}(e)-T_{n_{p-1}}(e)}) \\
&+ I(Y_{n_{p-1}}; U_e | Y_{1:n_{p-1}-1}, U_{T_{n_p}(e)\cap T_{n_{p-1}}(e)}, U_{T_{n_{p-1}}(e)-T_{n_p}(e)}) \\
\overset{(c)}{\leq} &I(Y_{n_p:n_k}; U_e | Y_{1:n_p-1}, U_{T_{n_p}(e)\cap T_{n_{p-1}}(e)}) + \\
&I(Y_{n_{p-1}}; U_e | Y_{1:n_{p-1}-1}, U_{T_{n_p}(e)\cap T_{n_{p-1}}(e)}, U_{T_{n_{p-1}}(e)-T_{n_p}(e)}) \\
\overset{(d)}{=} &I(Y_{n_p:n_k}; U_e | Y_{1:n_p-1}, U_{T_{n_p}(e)\cap T_{n_{p-1}}(e)}, U_{T_{n_{p-1}}(e)-T_{n_p}(e)}) \\
&+ I(Y_{n_{p-1}}; U_e | Y_{1:n_{p-1}-1}, U_{T_{n_p}(e)\cap T_{n_{p-1}}(e)}, U_{T_{n_{p-1}}(e)-T_{n_p}(e)}) \\
\overset{(e)}{=} &I(Y_{n_p:n_k}; U_e | Y_{1:n_p-1}, U_{T_{n_{p-1}}(e)}) + I(Y_{n_{p-1}}; U_e | Y_{1:n_{p-1}-1}, U_{T_{n_{p-1}}(e)}) \\
\leq &I(Y_{n_p:n_k}; U_e | Y_{1:n_p-1}, U_{T_{n_{p-1}}(e)}) + I(Y_{n_{p-1}:n_p-1}; U_e | Y_{1:n_{p-1}-1}, U_{T_{n_{p-1}}(e)}) \\
\overset{(f)}{=} &I(Y_{n_{p-1}:n_k}; U_e | Y_{1:n_{p-1}-1}, U_{T_{n_{p-1}}(e)})
\end{aligned}$$

where (a) is due to our assumption that (F.16) holds for p; (b) is due to the equalities, $T_{n_p}(e) = (T_{n_p}(e) \cap T_{n_{p-1}}(e)) \cup (T_{n_p}(e) - T_{n_{p-1}}(e))$ and $T_{n_{p-1}}(e) = (T_{n_p}(e) \cap T_{n_{p-1}}(e)) \cup (T_{n_{p-1}}(e) - T_{n_p}(e))$; (c) is due to our premise that $\mathcal{W}$ is distributive: for each $e' \in T_{n_p}(e) - T_{n_{p-1}}(e)$, $\alpha(e') \leq n_k$, and thus $U_{e'}$ is a function of $Y_{1:n_k}$; therefore, according to Proposition F.2, we

have:

$$I(Y_{n_p:n_k}; U_e | Y_{1:n_p-1}, U_{T_{n_p}(e)\cap T_{n_{p-1}}(e)}, U_{T_{n_p}(e)-T_{n_{p-1}}(e)})$$
$$\leq I(Y_{n_p:n_k}; U_e | Y_{1:n_p-1}, U_{T_{n_p}(e)-T_{n_{p-1}}(e)})$$

(d) is also due to our premise that $\mathcal{W}$ is distributive: for each $e' \in T_{n_{p-1}}(e) - T_{n_p}(e)$, $\alpha(e') \leq n_p - 1$, and thus $U_{e'}$ is a function of $Y_{1:n_p-1}$; therefore, the following equality holds according to Proposition F.1:

$$I(Y_{n_{p-1}}; U_e | Y_{1:n_{p-1}-1}, U_{T_{n_p}(e)\cap T_{n_{p-1}}(e)}, U_{T_{n_{p-1}}(e)-T_{n_p}(e)})$$
$$= I(Y_{n_{p-1}}; U_e | Y_{1:n_{p-1}-1}, U_{T_{n_p}(e)\cap T_{n_{p-1}}(e)})$$

(e) is again due to $T_{n_{p-1}}(e) = (T_{n_p}(e) \cap T_{n_{p-1}}(e)) \cup (T_{n_{p-1}}(e) - T_{n_p}(e))$; (f) is due to chain rule of mutual information. Thus, (F.16) holds for $p-1$. This means that (F.16) must hold for all $1 \leq p \leq k$. Letting $p = 1$ in (F.16), we have:

$$\sum_{i=1}^{k} I(Y_i; U_e | Y_{1:n_i-1}, U_{T_{n_i}(e)}) \leq I(Y_{n_1:n_k}; U_e | Y_{1:n_1-1}, U_{T_{n_1}(e)}) \leq H(U_e)$$

Thus, the lemma holds. ■

Let e be an edge that is passed through by at least one path in an extendable path-set sequence $\mathcal{K}$. According to the above definition, all the paths in $\mathcal{K}$ that pass through e must pass through a single edge in $\mathcal{W}$. We use μ_e to denote this edge, and refer to it as the *representative* of e in $\mathcal{W}$.

Proof of Theorem 4.4.1. Let $\mathcal{W} = \{C_i : 1 \leq i \leq K\}$ be a cumulative and distributive cut-set sequence, $\mathcal{T}$ a permutation sequence for $\mathcal{W}$ that satisfies the conditions of Definition 4.4.2, and $\mathcal{K} = \{\mathcal{P}_i : 1 \leq i \leq K\}$ an extendable path-set sequence for $\mathcal{W}$. Let $\mathbf{R} = (R'_i : 1 \leq i \leq K)$

be an arbitrary rate vector in $\mathcal{R}_{nc}$. Therefore, for $\epsilon = \frac{1}{k} > 0$ ($k \in \mathbb{Z}_{>0}$), there exists a network code which satisfies (4.3)-(4.5). In the rest of this proof, all the random variables are defined in this network code.

We then define the following routing scheme: for $1 \leq i \leq K$,

$$f_i^{n,k}(P) = \begin{cases} \frac{1}{n} I(Y_i; U_e | Y_{1:i-1}, U_{T_i(e)}) & \text{if } P \in \mathcal{P}_i, e \in P \cap C_i; \\ 0 & \text{otherwise.} \end{cases}$$

Since $\mathcal{W}$ is cumulative, the following equation holds:

$$\begin{aligned} &\sum_{P \in \mathcal{P}_{s_i d_i}} f_i^{n,k}(P) = \sum_{P \in \mathcal{P}_i} f_i^{n,k}(P) \\ =&\frac{1}{n} \sum_{e \in C_i} I(Y_i; U_e | Y_{1:i-1}, U_{T_i(e)}) \\ \overset{(a)}{=}&\frac{1}{n} I(Y_i; U_{C_i} | Y_{1:i-1}) \overset{(b)}{\geq} \frac{1}{n} I(Y_i; U_{\mathrm{In}(d_i)} | Y_{1:i-1}) \end{aligned} \tag{F.17}$$

where (a) is due to (4.25), and (b) is due to (4.24). Define $\delta_i' = Pr(Y_i$ cannot be decoded from $U_{\mathrm{In}(d_i)}, Y_{1:i-1})$. Clearly, $\delta_i' \leq \delta_i \leq \frac{1}{k}$. Then, we can derive the following equation:

$$\begin{aligned} &\frac{1}{n} I(Y_i; U_{\mathrm{In}(d_i)} | Y_{1:i-1}) = \frac{1}{n} (H(Y_i | Y_{1:i-1}) - H(Y_i | U_{\mathrm{In}(d_i)}, Y_{1:i-1})) \\ \overset{(c)}{=}&\frac{1}{n} (H(Y_i) - H(Y_i | U_{\mathrm{In}(d_i)}, Y_{1:i-1})) \overset{(d)}{\geq} \frac{1}{n} (H(Y_i) - 1 - \delta_i' \log |\mathcal{Y}_i|) \\ =&(1 - \delta_i') \frac{1}{n} H(Y_i) - \frac{1}{n} \overset{(e)}{\geq} \left(1 - \frac{1}{k}\right) \left(R_i' - \frac{1}{k}\right) - \frac{1}{n} \end{aligned}$$

where (c) is due to the fact that Y_i is independent from $Y_{1:i-1}$; (d) is due to Fano Inequality; (e) is due to (4.7). Combining the above equation with (F.17), the following inequality holds:

$$\sum_{P \in \mathcal{P}_{s_i d_i}} f_i^{n,k}(P) \geq \left(1 - \frac{1}{k}\right) \left(R_i' - \frac{1}{k}\right) - \frac{1}{n} \tag{F.18}$$

Let e be an edge that is passed through by at least one path in $\mathcal{K}$. Since $\mathcal{K}$ is extendable, the paths in $\mathcal{K}$ that pass through e must pass through e's representative μ_e in $\mathcal{W}$. Hence, the following equation holds:

$$\begin{aligned}
&\sum_{i=1}^{K} \sum_{P\in\mathcal{P}_{s_i d_i}, e\in P} f_i^{n,k}(P) = \sum_{i=1}^{K} \sum_{P\in\mathcal{P}_i, e\in P} f_i^{n,k}(P) \\
\leq &\sum_{i=1}^{K} \sum_{P\in\mathcal{P}_i, \mu_e\in P} f_i^{n,k}(P) = \frac{1}{n} \sum_{1\leq i\leq K, \mu_e\in C_i} I(Y_i; U_{\mu_e}|Y_{1:i-1}, U_{T_i(\mu_e)}) \\
\overset{(f)}{\leq} &\frac{1}{n} H(U_{\mu_e}) \overset{(g)}{\leq} 1 + \frac{1}{k}
\end{aligned} \tag{F.19}$$

where (f) is due to (4.28); (g) is due to (4.6).

Since each $f_i^{n,k}(P)$ has an upper bound, there exists a sequence $(n_l, k_l)_{l=1}^{\infty}$ such that for $1 \leq i \leq K$, the sequence $(f_i^{n_l,k_l}(P))_{l=1}^{\infty}$ approaches a finite limit. Define the following routing scheme:

$$f_i(P) = \begin{cases} \lim_{l\to\infty} f_i^{n_l,k_l}(P) & \text{if } P \in \mathcal{P}_i \\ 0 & \text{otherwise.} \end{cases}$$

Due to (F.18) and (F.19), $f_i(P)$ satisfies (4.1) and (4.2). Hence, $\mathbf{R} \in \mathcal{R}_r$. This implies that $\mathcal{R}_{nc} \subseteq \mathcal{R}_r$, and the network is routing-optimal. ■

F.3 Proofs for Examples

Proof of Theorem 4.5.1. Assume $\mathcal{W}$ is cumulative. Hence, G_1 is information-distributive According to Theorem 4.4.1, G_1 is routing-optimal. Since routing can achieve a common rate of at most $\frac{1}{K}$, $l_{min} = mK$.

Now assume $l_{min} = mK$. We consider a side-information graph $G' = (V', E')$ [66], where

$V' = \{1, \cdots, K\}$, and $E' = \{(j,i) : X_i \in \mathcal{H}_j, 1 \leq i,j \leq K\}$. It has been shown that if $l_{min} = mK$, then G' is acyclic [66]. We will show that $\mathcal{W}$ is information-distributive. Since G' is acyclic, we can re-index the nodes in G', such that if $(j,i) \in E'$, $j < i$. Let $1 \leq i < j \leq K$. Consider a path P from s_j to d_i. Since $(j,i) \notin E'$, $X_j \notin \mathcal{H}_i$. Thus, there is no directed edge from s_j to d_i in G_1, and P must pass through $(u,v) \in C_i$. Hence, $\mathcal{W}$ is cumulative, and G_1 is information-distributive. ■

Proof of Lemma 4.5.1. Let $0 \leq i < j \leq K$. Assume there is a directed path P from s_j to d_i. Let P_1 be the part of P after $s[j]$. Clearly, $P' = \{(s_i, s[i]), (s[i], s[i+1]), \cdots, (s[j-1], s[j])\} \cup P_1$ is a directed path from s_i to d_i. Since $C[i]$ is a cut-set between $s[i]$ and $d[i]$, P' must pass through an edge $e[k] \in C[i]$. Thus, $e[k] \in P$. This means that $\mathcal{W}$ is cumulative. ■

Since the duration between $e[t]$ and $s[\alpha(e[t])]$ is $\delta(e)$, we have:

$$\alpha(e[t]) = t - \delta(e) \tag{F.20}$$

Lemma F.1. If $C[0]$ is distributive, $\mathcal{W}$ is distributive.

Proof. Let $T[t] = (e_i[t_i + t])_{i=1}^k$, and define a permutation sequence $\mathcal{T} = (T[t])_{t=0}^K$ for $\mathcal{W}$. We will prove that if $C[0]$ is distributive, $\mathcal{T}$ satisfies (4.26) and (4.27).

Consider an edge $e_p[t_p] \in C[0]$. Let$(e_p[t_{n_i}])_{i=1}^k$ be the recurrent sequence in $C[0]$, in which all the edges are time-shifted versions of e_p. Without loss of generality, let $n_j = p$. Next, consider $e_p[t_p + k] \in C[k]$. Let $\mathcal{W}(e_p[t_p + k]) = \{C[t] : e_p[t_p + k] \in C[t], 0 \leq t \leq K\}$ denote the subset of cut-sets which contain $e_p[t_p + k]$. Clearly, $C[k - t_{n_{j+1}} + t_{n_j}]$ and $C[k + t_{n_j} - t_{n_{j-1}}]$ are the cut-sets in $\mathcal{W}(e_p[t_p + k])$ that lies immediately before and after $C[k]$ respectively, and $C[k + t_{n_j} - t_{n_1}]$ is the last cut-set in $\mathcal{W}(e_p[t_p + k])$.

Consider an edge $e_q[t_q + k] \in C[k]$ be an edge that lies before $e_p[t_p + k]$ in $T[k]$, but doesn't

appear before $e_p[t_p + k]$ in $T[k - t_{n_{j+1}} + t_{n_j}]$. This means that $e_q[t_q + t_{n_{j+1}} - t_{n_j}] \notin C[0]$. Thus, the following equation holds:

$$\alpha(e_q[t_q + k]) = k + t_q - \delta(e_q) \overset{(a)}{\leq} k + t_{n_j} - t_{n_1}.$$

where (a) is due to the premise that $C[0]$ is distributive. Hence, (4.26) is satisfied.

Now assume that $e_q[t_q + k] \in C[k]$ lies before $e_p[t_p + k]$ in $T[k]$, but doesn't appear before $e_p[t_p + k]$ in $T[k - t_{n_j} + t_{n_{j-1}}]$. This implies that $e_q[t_q - t_{n_j} + t_{n_{j-1}}] \notin C[0]$. Thus, the following equation holds:

$$\alpha(e_q[t_q + k]) = k + t_q - \delta(e_q) \overset{(b)}{\leq} k + t_{n_j} - t_{n_{j-1}} - 1$$

where (b) is again due to the premise that $C[0]$ is distributive. Hence, (4.27) is satisfied. $\mathcal{W}$ is distributive. ■

Lemma F.2. If $\mathcal{P}$ is extendable, $\mathcal{K}$ is extendable.

Proof. Consider two paths $P_i, P_j \in \mathcal{P}$. Assume $P_i[k_1]$ overlaps with $P_j[k_2]$ at $e[t]$. Thus, $e[t - k_1] \in P_i$ and $e[t - k_2] \in P_j$. Since $\mathcal{P}$ is extendable, this means that $e_i = e_j$ and

$$t_i - t_j = t - k_1 - (t - k_2) = k_2 - k_1$$

Note that $e_i[t_i + k_1]$ is the edge in $\mathcal{W}$ that is passed through by $P_i[k_1]$. We have:

$$e_i[t_i + k_1] = e_j[t_j + k_2] \in P_j[k_2] \cap C[k_2].$$

Thus, $P_i[k_1]$ and $P_j[k_2]$ pass through the same edge $e_i[t_i + k_1]$ in $\mathcal{W}$. Hence, $\mathcal{K}$ is extendable.

■

Proof of Theorem 4.5.2. Due to Lemmas F.2, 4.5.1 and F.1, the theorem holds. ■