A Study of Communication Networks through the Lens of Reduction

Thesis by
Ming Fai Wong

In Partial Fulfillment of the Requirements for the
degree of
Doctor of Philosophy

CALIFORNIA INSTITUTE OF TECHNOLOGY
Pasadena, California

2017
Defended May 24, 2017

To my family

and

Mengyao

ACKNOWLEDGMENTS

The material in this thesis is based upon work supported by the National Science Foundation under Grant Numbers 1018741, 1038578, 1321129, 1527524, and 1526771, and by the Israel Science Foundation under Grant Number 480/08.

I would like to thank Prof. Effros for being such a wonderful research advisor throughout the course of my studies. Her wisdom and experience have helped me tremendously in my development towards an independent researcher. Her patience and kindness have enabled me to truly enjoy the learning experience at Caltech.

I would like to thank Prof. Langberg for being such a great mentor. It is a privilege and my pleasure to be able to work with him. Prof. Langberg has provided me with very valuable insights and feedback, which enabled me to make steady progress during my doctoral research.

I would like to thank Prof. Bruck, Prof. Ho, Prof. Low, and Prof. Umans for serving on my committee and for the useful suggestions.

I would also like to acknowledge Howard and Jan Oringer for their continuous generous support of the lab.

Throughout the years at Caltech, I have met many exceptional people and friends. I have had the fortune to meet Ruizhe, Donglei, Ding, and Yingrui during my first year at Caltech. I thank them for being such wonderful travel companions and for the unforgettable hiking adventures. To Parham, thank you for being such a great friend and a great lab-mate; I have thoroughly enjoyed the discussions with you, both research related and otherwise. Special thanks to Boyu, Lucy, Qiuyu, Wentao and Yang for making the everyday life at Caltech fun and exciting, and for introducing me to my fiancée, Mengyao. I would also like to thank Matthew for being such an understanding roommate and a good friend. To Carlos and Kishore, thanks for making going to class fun. I have also learned a lot through my interactions with Chris, Derek, Mayank, Hongyi, Howard, Sahin, Shirin, Ted, and Wael; I thank them for making the lab so enjoyable.

Finally, I must express my heartfelt gratitude to my parents and to my fiancée, Mengyao, for providing me with unyielding support and encouragements throughout. Mengyao, this thesis would not have been possible without you. Thank you.

ABSTRACT

A central goal of information theory is to characterize the capacity regions of communication networks. Due to the difficulty of the general problem, research is primarily focused on families of problems defined by various classifiers. These classifiers include the channel transition function (i.e., noisy, deterministic, network coding), demand type (i.e., single-source, 2-unicast), network topology (i.e. acyclic network coding, index coding). To date, the families of networks that are fully solved remain limited. Moreover, results derived for one specific family often do not extend easily to other families of problems.

Our work shifts from the traditional focus on solving example networks to one that builds connections between problem solutions so that we can say where and when solving a problem in one domain would also solve a corresponding problem in another domain. Central to our approach is a technique called "reduction", in which we connect the solutions and results of communication problems. We say that problem A reduces to problem B when A can be solved by first transforming it to B and then applying a solution for B. We focus on two notions of reduction: reduction in code design and reduction in capacity region.

Our central results demonstrate reductions with respect to a variety of classifiers. We show that finding multiple multicast network capacity regions reduces to finding multiple unicast network capacity regions both when capacity is defined as the maximal rate over all possible codes and when capacity is defined as the optimal rate over linear codes. As a corollary to this result, we show that the same capacity reduction holds for when network types are limited to either network coding networks or index coding networks. In several instances, we show that a reduction in code design extends to a reduction in capacity region if and only if the edge removal conjecture holds. Here, the edge removal conjecture states that removing an edge of negligible capacity from a network does not change its capacity region.

One of the key challenges in network coding research is how to handle networks containing cycles. As a result, many papers on network coding restrict attention to acyclic networks and some results derived for acyclic networks do not extend to networks containing cycles. We consider a streaming model for network communication where information is streamed to its destination under a constraint on maximal delay at the decoder. Restricting our attention to this scenario enables us to prove a code reduction from network coding to index coding in both acyclic and cyclic networks. Since index coding networks are acyclic, a consequence of this reduction is that under the streaming model, there is no fundamental difference between acyclic and cyclic networks.

PUBLISHED CONTENT AND CONTRIBUTIONS

[1] M. F. Wong, M. Effros, and M. Langberg, “A code equivalence between streaming network coding and streaming index coding,” in *To appear in Information Theory, 2017 IEEE International Symposium on*, IEEE, 2017,
M. F. Wong participated in the conception of the project and prepared the manuscript.

[2] M. F. Wong, M. Effros, and M. Langberg, “On tightness of an entropic region outer bound for network coding and the edge removal property,” in *Information Theory, 2016 IEEE International Symposium on*, IEEE, 2016, pp. 1769–1773. DOI: 10.1109/ISIT.2016.7541603,
M. F. Wong participated in the conception of the project and prepared the manuscript.

[3] M. F. Wong, M. Effros, and M. Langberg, “On an equivalence of the reduction of k-unicast to 2-unicast capacity and the edge removal property,” in *Information Theory, 2015 IEEE International Symposium on*, IEEE, 2015, pp. 371–375. DOI: 10.1109/ISIT.2015.7282479,
M. F. Wong participated in the conception of the project and prepared the manuscript.

[4] M. F. Wong, M. Langberg, and M. Effros, “Linear capacity equivalence between multiple multicast and multiple unicast,” in *Information Theory, 2014 IEEE International Symposium on*, IEEE, 2014, pp. 2152–2156. DOI: 10.1109/ISIT.2014.6875214,
M. F. Wong participated in the conception of the project and prepared the manuscript.

[5] M. F. Wong, M. Langberg, and M. Effros, “On a capacity equivalence between multiple multicast and multiple unicast,” in *Communication, Control, and Computing, 2013 51st Annual Allerton Conference on*, IEEE, 2013, pp. 1537–1544. DOI: 10.1109/Allerton.2013.6736710,
M. F. Wong participated in the conception of the project and prepared the manuscript.

[6] M. F. Wong, M. Langberg, and M. Effros, “On a capacity equivalence between network and index coding and the edge removal problem,” in *Information Theory Proceedings, 2013 IEEE International Symposium on*, IEEE, 2013, pp. 972–976. DOI: 10.1109/ISIT.2013.6620371,
M. F. Wong participated in the conception of the project and prepared the manuscript.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The question of how to derive capacity region characterizations for communication networks remain a central open problem in information theory. So far, the capacity region is known only for a limited collection of networks, most of which are quite small. Examples include the point-to-point channel [1] and the multiple access channel [2], [3]. Knowledge of how to design good codes is even more limited. The difficulty of both capacity characterization and code design grows quickly with the number of terminals and demands. For example, the problem of determining the capacity regions of both the relay channel (which differs from the point-to-point channel in the addition of a relay node) and the broadcast channel (which differs from the point-to-point channel in the addition of a receiver) remains open. Due to the complexity of the general problem, research is primarily focused on subclasses of problems. These subclasses are defined by classifiers restricting network properties such as the channel transition function, demand type, network topology and so on. Results derived for one subclass do not extend easily to other subclasses in general.

In this work, instead of solving example networks, we study the taxonomy of communication problems by building connections between problem solutions using the reduction technique. We say that problem A reduces to problem B if A can be solved by first transforming it to B and then solving B. Reductions can be proven even when solutions to both problems are unavailable; when a solution for B is known, a reduction form A to B enables the propagation of results from B to A. Thus, understanding these connections expands tools and results.

We employ two notions of reduction: reduction in code design ("code reduction") and reduction in capacity region ("capacity reduction"). Roughly speaking, a reduction in code design refers to a mechanism for designing codes for a network in class A using a code design algorithm for networks in class B. Similarly, a reduction in capacity region refers to a mechanism for obtaining the capacity region of a network in class A using an algorithm for deriving the capacity region of a network in class B. We derive code reduction and capacity reduction results with respect to various classifiers. We also study known code reduction results and show that these code reductions can be extended to similar reductions in the capacity region if and only if the asymptotic edge removal statement (AERS) holds. The edge removal statement studies how incremental changes in network topology affect the network coding capacity region. It is directly connected to the vanishment conjecture [4] which states that the network coding capacity region remains unchanged when an edge of negligible capacity is added to the network.

We next give a summary of the contributions of this thesis.

1.1 Contributions

We study code design and the characterization of the capacity region for general memoryless networks. The general class of memoryless networks contains traditional memoryless channel models for point-to-point and multi-terminal channels as well as network coding and index coding networks. Full details of our communication model are given in **Chapter 2**.

Central to our approach is reduction and the edge removal statement, which enable us to connect the results of communication problems. **Chapter 3** introduces these two concepts. We derive code reductions and capacity reductions between subclasses of problems with respect to network demands and topologies. In some instances, we also connect code reduction to capacity reduction, demonstrating that the extension from code reduction to capacity reduction hinges on the edge removal statement. The tools and techniques that are used to derive these connections are documented in **Chapter 5**. In what follows, we describe our contributions in the regime of various taxonomies.

Network Reductions Related to Network Topology

Network coding networks are memoryless networks comprised of independent, noiseless, directed point-to-point channels. A network coding problem is specified by a directed graph comprised of capacitated edges and a set of network demands. Both the demand type and the network topology play an important role in the code design and the characterization of the capacity region. We here differentiate between general network coding networks and acyclic network coding networks, which differ only in that the former allow for directed cycles in the underlying graph while the latter do not. The class of index coding networks introduced by [5] is a subset of the class of acyclic network problems which restricts the network topology to a simple setting in which a single node with access to all the sources broadcasts a common message to all receivers, each of which has access to a subset of the sources as side information.

The authors of [6], [7] prove a code reduction from acyclic network coding to index coding. This reduction is proved under the assumptions of both general codes and linear codes. This result is intriguing because it shows that the complexity of acyclic networks is completely captured by the simple topology of index coding networks in which encoding is required at only one node in the network. Our work extends this idea by showing that under the same reduction mapping, capacity region for acyclic network coding reduces to index coding if and only if the edge removal statement holds (**Chapter 7**). We show this connection for both general and linear capacity regions. Since the edge removal statement holds for linear capacity regions [8], we obtain as a corollary a linear capacity reduction from acyclic network coding to index coding.

While many of the results derived in the field of network coding are derived only for acyclic networks, it is unknown whether the same results hold for networks with cycles or whether, instead, acyclic and cyclic networks are fundamentally different [9]. In acyclic networks, a valid network code can be characterized by assigning a *single* function of the sources to each edge such that each of these functions can be computed locally by internal nodes. However, such a simple characterization does not exist for networks containing cycles. As such, proofs that rely on this single-function characterization do not immediately extend to networks containing cycles. Our work considers a streaming network coding model [10], [11] in which each demand has to satisfy both rate and reconstruction delay constraints. Motivated by the goal of extending tools and results derived for acyclic networks to the more practically relevant domain of cyclic networks, we derive a code reduction from general network coding to index coding under the streaming model (**Chapter 10**), which enables us to overcome the key challenges for the cyclic case. A consequence of this reduction is that under the streaming model, there is no fundamental difference between acyclic and cyclic networks.

Network Reductions Related to Demand Type

In the realm of network communication, one may also distinguish between networks based on different demand types. Examples include single unicast, single multicast, multiple unicast and multiple multicast networks. A *single unicast* network is a network in which there is exactly one source and exactly one terminal demanding that source. A *multiple unicast* network is a network containing one or more sources and exactly one terminal demanding each source. A *single multicast* network is a network in which there is exactly one source and there are one or more terminals demanding that source. A *multiple multicast* network is a network in which there are one or more sources and one or more terminals demanding each source. The most general of these demand types is the multiple multicast case.

The work of [12] shows that code design for multiple multicast network coding reduces to code design for multiple unicast network coding; that is, solving a code design problem for any multiple multicast network can be achieved by solving a related code design problem for a related multiple unicast network. Similarly, the work of [13] shows that solving the linear code capacity region for any multiple multicast index coding network reduces to solving the linear code capacity region for a related multiple unicast index coding network. Thus, there is no loss of generality in restricting attention to multiple unicast demands in the above scenarios.

In this work we tackle capacity derivation in a more general setting. We show deriving the capacity region for multiple multicast networks reduces to deriving the capacity region for multiple unicast networks. We derive this reduction for both general codes and linear codes (**Chapter 4**). As a corollary, the reduction results for both the general and linear capacity regions apply to network coding as well as index coding. Thus, our work unifies some of the existing results.

In network coding networks, the capacity region of a single unicast network are fully characterized by the max flow of the underlying directed graph, and code design is well understood. The code design problem is open for networks with two or more sources. A surprising result of [14] shows that under the assumption of zero-error codes, the code design problem for multiple unicast network coding reduces to the code design problem for 2-unicast network coding. This implies that 2-unicast network coding problems are representative of network coding problems with general demands. Our work connects the code reduction in [14] to a capacity reduction by showing that the capacity region for multiple unicast network coding reduces to the capacity region for 2-unicast network coding if and only if the edge removal statement holds (**Chapter 6**). We prove this connection for both general and linear capacity regions. Again, we obtain a linear capacity reduction from multiple unicast network coding to 2-unicast network coding as a corollary since the edge removal statement holds for the linear capacity region [8].

Reduction Related to Entropic Vectors Characterizations of Network Capacity Region

The Yeung outer bound on the network coding capacity region is derived in [15, Theorem 15.9] under the notion of entropic vectors. Although a characterization of the capacity region is already known [16], due to the relative simplicity of the Yeung outer bound, it is well-studied in the literature [17], [18].

While the tightness of the Yeung outer bound remains open, the authors of [17] show that the Yeung outer bound is tight if the edge removal statement holds. We extend the result of [17] to an if-and-only-if relationship. That is, we show that the Yeung outer bound is tight if and only if the edge removal statement holds (**Chapter 8**). If the Yeung outer bound is tight, it provides us with another characterization of the network coding capacity region.

Zero-Error Versus Epsilon-Error Capacity Region

We study the zero-error network coding capacity region. The zero-error network coding capacity region equals the epsilon-error capacity region for super-source networks and networks with co-located sources [17], [19] but remains open in general. The authors of [19] show that whether or not the zero-error capacity region equals the epsilon-error capacity region is closely related to the edge removal question. Bounds for the zero-error capacity region are derived by relaxing the edge capacity constraint from a strict (worst case) requirement to an average requirement [20].

In **Chapter 9**, we derive a full characterization of the zero-error network coding capacity region using a dense subset of the entropic region. Our approach is inspired by [16]. Further, we show that the zero-error capacity region is equal to the epsilon-error network coding capacity region when restricted to linear codes.

Chapter 2

NETWORK MODELS

In this chapter, we give the formal definition of our main network model. We consider communication over general multi-terminal memoryless channels which we refer to as *communication networks.* The full model of a canonical communication network is given in Section 2.1, which also defines code and capacity region.

General network coding networks and index coding networks are sub-classes of general communication networks; these are defined in Sections 2.2 and 2.3, respectively. The acyclic network coding model is introduced in Chapter 5. The streaming network coding and index coding model is introduced in Chapter 10.

2.1 Canonical Communication Network

Here, we describe a canonical model for communication networks. We assume all networks are canonical unless stated otherwise.

For a positive i, let $[i]$ denote $\{1, \cdots, \lceil i \rceil\}$. Following the canonical model in [13], we specify k by l communication network instance $\mathcal{I}$ by a vector of network parameters

$$\mathcal{I} = (S, T, U, H, p(\mathbf{Y}|\mathbf{X})).$$

We define each of the terms in that vector below. The sets S and T represent the k *source* nodes $S = \bigcup_{i \in [k]} \{s_i\}$ where source messages originate and the kl *terminal* nodes $T = \bigcup_{i \in [k]} \bigcup_{j \in [l]} \{t_{i,j}\}$ that demand those messages. We use s_i to represent the source node of the ith message and $t_{i,j}$ to represent the jth terminal node of message i. Thus, terminal node $t_{i,j}$ demands message from source node s_i. We require source nodes and terminal nodes to be unique (i.e., $|S \cup T| = k + kl$). When $l = 1$, we refer to the instance as a k-unicast communication network. For ease of notation, when $l = 1$, the second subscript of the terminal node is dropped (i.e., $t_i = t_{i,1}$).

The set U represents the *relay* nodes of the network, these nodes do not have any demands and do not generate any source messages (i.e., $U \cap (T \cup S) = \varnothing$). Any source messages that are available as side information to a relay node or terminal node are captured by the "has" sets

$$H = \bigcup_{v \in U \cup T} \{H_v\},$$

where $H_v \subseteq S$ for each $v \in U \cup T$. For each $v \in U \cup T$ and each $s \in H_v$, we model the direct availability of the source using an infinite capacity link going from node s to node v. These infinite capacity links are used not to represent real physical channels but instead to capture the notion that source information is available a priori to some subset of nodes in the network [1] . For each $v \in U \cup T$, we denote by

$$W_{H_v} = (W_s : s \in H_v) \in \prod_{s \in H_v} \mathcal{W}_s = \mathcal{W}_{H_v}$$

the vector of source random variables available to node v.

We consider a channel model where all nodes except the source nodes S are operated simultaneously in every time step. We refer to this as a "simultaneous" code schedule. That is, for n channel uses, the channel is operated over the same n time steps $\tau \in [n]$. Each relay node $u \in U$ transmits a channel input variable $X_{u,\tau} \in \mathcal{X}_u$ and receives a channel output variable $Y_{u,\tau} \in \mathcal{Y}_u$ at each time step $\tau \in [n]$. Each terminal node $t \in T$ receives a *channel output* variable $Y_{t,\tau} \in \mathcal{Y}_t$ at each time step $\tau \in [n]$ but transmits no network input. The transition probability $p(\mathbf{Y}|\mathbf{X})$ is a function that describes the probability at each time τ of observing the channel output variables

$$\mathbf{Y} = \mathbf{Y}_\tau = (Y_{v,\tau}, v \in U \cup T)$$

given that the channel input variables are

$$\mathbf{X} = \mathbf{X}_\tau = (X_{v,\tau}, v \in U);$$

this probability is independent of τ by assumption. It is also assumed to be memoryless, so the channel output at time τ is conditionally independent of both the channel inputs and the channel outputs at prior times, given the channel input at time τ. An example appears in Figure 2.1.

Given a rate vector $\mathbf{R} = (R_1, \ldots, R_k)$ and a blocklength n, a $(2^{n\mathbf{R}}, n)$ communication code $\mathcal{C}$ is a mechanism for simultaneous transmission of a rate R_i

[1] Making sources directly available to those nodes would be technically equivalent but notationally less convenient for our purposes.

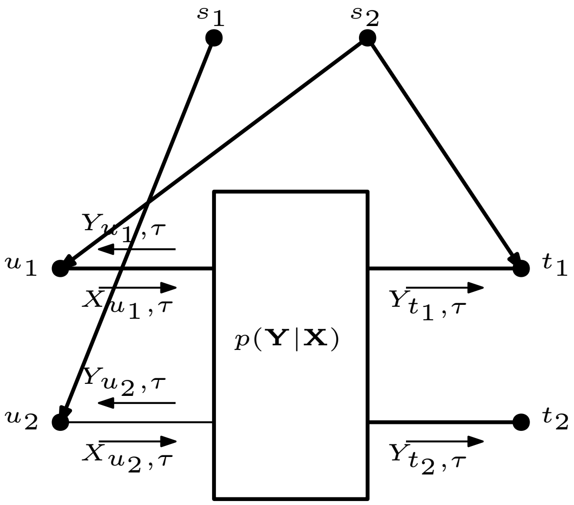


Figure 2.1: A 2 by 1 communication network with $S = \{s_1, s_2\}$, $T = \{t_1, t_2\}$, $U = \{u_1, u_2\}$ and $H_{u_1} = \{s_2\}, H_{u_2} = \{s_1\}, H_{t_1} = \{s_2\}, H_{t_2} = \varnothing$.

message from each source $s_i \in S$ to its corresponding terminals $t_{i,1}, \cdots, t_{i,l}$ over n uses of the network $\mathcal{I}$. Each source node $s \in S$ holds an nR_s-bit *source message* random variable

$$W_s \in \mathbb{F}_2^{nR_s}$$

that is uniformly distributed over its alphabet and independent of all other sources.

The communication code

$$\mathcal{C} = (\{f_{u,\tau} u \in U, \tau \in [n]\}, \{g_t, t \in T\})$$

consists of a set of encoders $\{f_{u,\tau}\}$ and a set of decoding functions $\{g_t\}$. Code $\mathcal{C}$ assigns n encoding functions $\{f_{u,\tau}, \tau \in [n]\}$ to each relay node $u \in U$, one for each time step $\tau \in [n]$, and a single *block* decoding function d_t for each terminal $t \in T$. For $v \in T \cup U$ and for each $s \in H_v$, we assume the entire source message W_s is available to v, thus no encoding function is required for source nodes.

The time-τ output at any node can rely only on inputs to the same node at prior timesteps. Thus, for each $u \in U$ and $\tau \in [n]$, the relay node encoding function

$$f_{u,\tau} : \mathcal{Y}_u^{\tau-1} \times \prod_{s \in H_u} \mathbb{F}_2^{nR_s} \to \mathcal{X}_u$$

maps the *previously received* symbols $\mathbf{Y}_u^{\tau-1} = (Y_{u,1}, \cdots, Y_{u,\tau-1})$ and the available source messages $W_{H_u} = (W_s, s \in H_u)$ to the message $X_{u,\tau}$ transmitted by node u at time τ.

For each $t \in T$, the decoding function

$$g_t : \mathcal{Y}_t^n \times \prod_{s \in H_t} \mathbb{F}_2^{nR_s} \rightarrow \mathbb{F}_2^{nR_t}$$

maps the *complete* vector of received symbols at the end of n channel uses and the available source messages $W_{H_t} = (W_s, s \in H_t)$ to a reproduction $\widehat{W_t}$ of the message W_t desired by terminal t.

For any non-source node $v \in U \cup T$, denote by

$$Z_v = (\mathbf{Y}_v^n, W_{H_v}) \in \mathcal{Y}_v^n \times \mathcal{W}_{H_v} = \mathcal{Z}_v$$

the total information available to node v at the end of n channel uses. Throughout, when $s = s_i$, we use notation W_s and W_i, $\mathcal{W}_s$ and $\mathcal{W}_i$, and R_s and R_i interchangeably; that is $W_s = W_i$, $\mathcal{W}_s = \mathcal{W}_i$, and $R_s = R_i$ when $s = s_i$. Similarly, when $t = t_{i,j}$, $\mathcal{W}_t = \mathcal{W}_i$, $R_t = R_i$, and $W_t = W_i$.

The performance of a communication code is characterized by its rate vector $\mathbf{R}$ and error probability $P_e^{(n)}$, where

$$P_e^{(n)} = \Pr\left(\bigcup_{t \in T} \{\widehat{W_t} \neq W_t\}\right)$$

is the probability that one or more terminal node decodes its desired source in error.

Remark 1. *There is no loss of generality in restricting attention to the canonical form of [13] used here. For any terminal t that sends a network input or demands q sources where $q > 1$, we add q new terminal nodes $t_1', \cdots t_q'$ such that each new terminal node t_i' receives the same channel output as t and has the same set of sources messages available to t. The ith new terminal t_i' now demands the ith source originally demanded by t; node t no longer demands any source and becomes a relay node. Similarly, if there is a source message W_i demanded by $0 < m < l$ terminals $t_{i,1}, \cdots, t_{i,m}$, we add $l - m + 1$ new terminals $t_1', \cdots, t_{l-m+1}'$ such that each new terminal node t_i' receives the same channel output as $t_{i,m}$ and has the same set of sources messages available to $t_{i,m}$. Each new terminal node t_i' demands W_i; node $t_{i,m}$ no longer demands any* *source and becomes a relay node. If $m = 0$, the message W_i is removed from the network. These modifications do not change the amount of information that is available to any of the old terminal nodes. Further, no new information is available to any of the new terminal nodes; hence the capacity region, defined below, remains unchanged.*

Code Feasibility and Capacity Regions

A communication network instance $\mathcal{I}$ is said to be $(\mathbf{R}, \epsilon, n)$-feasible if there exists a code $\mathcal{C}$ with blocklength n such that operation of code $\mathcal{C}$ on source message random variables $\mathbf{W} = (W_s : s \in S)$ uniformly distributed on $\mathcal{W} = \prod_{s\in S} \mathcal{W}_s = \prod_{s\in S} \mathbb{F}_2^{nR_s}$ yields error probability $P_e^{(n)} \leq \epsilon$.

We apply the notion of feasibility to define two notions of capacity. Here $\overline{A}$ denotes the closure of a set A. The *ϵ-error capacity region* of $\mathcal{I}$, denoted by $\mathcal{R}_\epsilon(\mathcal{I})$, captures the asymptotic notion of reliability as

$$\mathcal{R}_\epsilon(\mathcal{I}) = \overline{\{\mathbf{R} : \forall\epsilon > 0,\ \mathcal{I} \text{ is } (\mathbf{R}, \epsilon, n)\text{-feasible infinitely often in } n\}}.$$

The 0*-error capacity region* of $\mathcal{I}$, denoted by $\mathcal{R}_0(\mathcal{I})$, captures the notion of perfect reliability as

$$\mathcal{R}_0(\mathcal{I}) = \overline{\{\mathbf{R} : \mathcal{I} \text{ is } (\mathbf{R}, 0, n)\text{-feasible infinitely often in } n\}}.$$

Since any code with $P_e^{(n)} = 0$ also satisfies $P_e^{(n)} \leq \epsilon$ for all $\epsilon > 0$,

$$\mathcal{R}_0(\mathcal{I}) \subseteq \mathcal{R}_\epsilon(\mathcal{I}).$$

Linear Codes and Capacities

It is sometimes useful for practical reasons to restrict attention to low complexity coding modalities. In the discussion that follows, we consider both the general case, where codes may be arbitrary, and the case of linear codes, which are here defined to be codes with linear encoders and general decoders. In this work, we consider linearity over $\mathbb{F}_2$. Thus for a linear code, we require all input and output alphabets to be vectors over $\mathbb{F}_2$. Each encoder $f_{u,\tau}$ is represented by a matrix over $\mathbb{F}_2$, giving

$$X_{u,\tau} = (Y_u^{\tau-1}, W_{H_u}) f_{u,\tau}.$$

The decoding functions $\{g_t, t \in T\}$ are arbitrary (i.e., not necessarily linear) functions.

A network instance $\mathcal{I}$ is $(\mathbf{R}, \epsilon, n)$ *linearly-feasible* if it is $(\mathbf{R}, \epsilon, n)$ feasible using a linear code. The linear capacity regions are defined as

$$\begin{aligned} \mathcal{R}^L_\epsilon(\mathcal{I}) &= \overline{\{\mathbf{R} : \forall \epsilon > 0,\ \mathcal{I} \text{ is } (\mathbf{R}, \epsilon, n) \text{ linearly-feasible infinitely often in } n\}} \\ \mathcal{R}^L_0(\mathcal{I}) &= \overline{\{\mathbf{R} : \mathcal{I} \text{ is } (\mathbf{R}, 0, n) \text{ linearly-feasible infinitely often in } n\}}. \end{aligned}$$

2.2 Network Coding Networks

A network coding network is a communication network instance where nodes are connected by independent, point-to-point, noiseless communication links. These links are directed and each has a capacity value which describes the maximal rate of communication across each link. For a directed link (u, v) of capacity c bits that connects node u to node v, the channel input alphabet $\mathcal{X}_u$, channel output alphabet $\mathcal{Y}_v$ and the channel transition function $p(Y_v|X_u)$ are given by

$$\begin{aligned} \mathcal{X}_u &= \mathbb{F}_2^c, \\ \mathcal{Y}_v &= \mathbb{F}_2^c, \\ p(Y_v = y_v | X_u = x_u) &= \delta(x_u - y_v). \end{aligned}$$

A network coding network is therefore a communication network such that the channel transition function is a product of channel transition functions of point-to-point, noiseless links. We first give a description of a network coding instance before formally describing the restriction on the channel transition function.

By representing each of these directed links as an edge of a directed graph, a network coding network can be described by $\mathcal{I} = (G, S, T)$ where the graph $G = (V, E, C)$ is defined by a set of vertices V representing the nodes of the network, a set of directed edges $E \subseteq V^2$ representing communication links between these devices, and a vector $C = (c_e : e \in E)$ specifying the capacity for each edge. sets $S, T \subset V$ are the source nodes and terminal nodes, respectively. Each edge e is a noiseless channel of integer[2] capacity c_e from the edge's input node, here called $\text{In}(e)$, to its output node, here called $\text{Out}(e)$; for example, if $e = (u, v)$ and $C_e = 1$, then information travels from node $\text{In}(e) = u$ to node $\text{Out}(e) = v$ at a rate of $C_e = 1$ bit per transmission. Figure 2.2 shows an

[2] For an non-integral c_e, we may model each link to transmit $\lfloor(\tau c_e)\rfloor - \lfloor(\tau - 1)\rfloor c_e$ bits of information per time step. The same proofs presented in our work suffice to treat these cases.

example. Denote by $S^c = V \setminus S$ the set of non-source nodes in $\mathcal{I}$ and by

$$E_{S^c} = \{e \in E : \mathrm{In}(e) \in S^c\}$$

the set of edges that do not originate from source nodes. Similarly, denote by

$$E_S = \{e \in E : \mathrm{In}(e) \in S\}$$

the set of source edges. Similar to the setup of communication networks, any source edge $e \in E_S$ are infinite capacity edges.

A network coding network is therefore a communication network with a channel transition function that can be described by $\mathcal{I} = (G, S, T)$ and takes the following form: The relay nodes are given by $U = V \setminus (S \cup T)$. The channel alphabets $(\mathcal{X}_u, u \in U)$ and $(\mathcal{Y}_v, v \in T \cup U)$ are given by

$$\mathcal{X}_u = \prod_{v \in U \cup T : (u,v) \in E_{S^c}} \mathcal{X}_{(u,v)},$$
$$\mathcal{Y}_v = \prod_{u \in U : (u,v) \in E_{S^c}} \mathcal{Y}_{(u,v)},$$

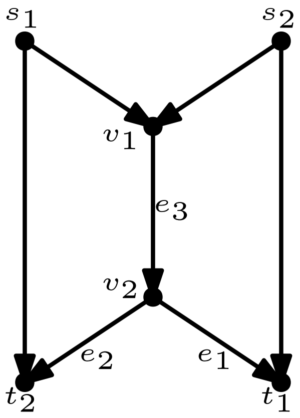
where for $e = (u, v)$, each $\mathcal{X}_{(u,v)} = \mathcal{Y}_{(u,v)} = \mathbb{F}_2^{c_e}$. The "has" sets $(H_v, v \in U \cup T)$ and channel transition function $p(\mathbf{Y}|\mathbf{X})$ are given by

$$H_v = \Big\{ s : (s, v) \in E_S \Big\}$$
$$p(\mathbf{Y}|\mathbf{X}) = \prod_{(u,v) \in E_{S^c}} \delta(x_{(u,v)} - y_{(u,v)}).$$

Again, there is no loss of generality in restricting attention to the canonical form. More specifically, in a canonical k by l network coding network, there are kl terminal nodes and each terminal $t_{i,j}, i \in [k], j \in [l]$ has no outgoing edge and demands W_i (see Remark 1).

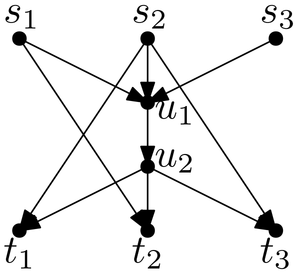
2.3 Index Coding Networks

An index coding network is network coding network with a graph G falling in a restricted class of possible network coding topologies. A k by l multiple multicast index coding network is a network coding network with a set of k source nodes $S = \{s_1, \cdots, s_k\}$, kl terminal nodes $T = \bigcup_{i \in [k]} \bigcup_{j \in [l]} \{t_{i,j}\}$ and two relay nodes $U = \{u_1, u_2\}$. Following our convention from canonical network coding instances, each s_i holds the ith source message variable and each $t_{i,j}$ is



Vertex set	$V = \{s_1, s_2, t_1, t_2, v_1, v_2\}$
Edge set	$E = \{(s_1, v_1), (s_2, v_1),$ $(s_1, t_2), (s_2, t_1), (v_1, v_2),$ $(v_2, t_2), (v_2, t_1)\}$
Capacity vector	$C = (\infty, \infty, \infty, \infty, 1, 1, 1)$
Source node	$S = \{s_1, s_2\}$
Terminal nodes	$T = \{t_1, t_2\}$

Figure 2.2: A network reminiscent of the "butterfly" network. All edges are of infinity capacity except for e_1, e_2, and e_3, which are of capacity 1. Each terminal t_i demands sources from s_i.



Source nodes	$S = \{s_1, s_2, s_3\}$
Terminal nodes	$T = \{t_1, t_2, t_3\}$
Has set	$H_{t_1} = \{s_2\}$, $H_{t_2} = \{s_1\}, H_{t_3} = \{s_2\}$
Broadcast capacity	c_B

Figure 2.3: Representation of an index coding instance (S, T, H, c_B) as a network coding problem (left). Node u_2 broadcasts a common message of rate c_B to all terminal nodes. For each terminal node t and each source node s in the has set H_t of t, there is an edge going from node s to node t. Edges (u_1, u_2), (u_2, t_1), (u_2, t_2), and (u_2, t_3) have capacity c_B. All other edges have infinite capacity.

the jth receiver of the ith source message. When expressed in the form of a network coding network, we have

$$V = \{u_1, u_2\} \cup S \cup T.$$

The set of links includes an infinite-capacity link from each source node to node u_1, a capacity c_B "*bottleneck link*" B from node u_1 to node u_2, a capacity

c_B link from node u_2 to each terminal, and a collection of infinite-capacity links from source nodes to terminal nodes. The source nodes connected to a given terminal node $t \in T$ are described by the "has" set H_t of terminal t. Thus,

$$\begin{aligned} E &= \left[\bigcup_{s\in S}\{(s,u_1)\}\right] \cup \{(u_1,u_2)\} \cup \left[\bigcup_{t\in T}\{(u_2,t)\}\right] \cup \left[\bigcup_{t\in T}\bigcup_{s\in H_t}\{(s,t)\}\right] \\ c_e &= \begin{cases} c_B & \text{if } \mathrm{In}(e) \in \{u_1,u_2\} \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

Given these restrictions, an instance $\mathcal{I} = (G,S,T)$ of a k by l canonical network coding network that falls in the sub-class of k by l index coding problems can be entirely described by a set of source nodes $S = \{s_1, \cdots, s_k\}$, a set of terminal nodes $T = \bigcup_{i\in[k]}\bigcup_{j\in[l]}\{t_{i,j}\}$, a set of has sets $H = \{H_t, t \in T\}$, and the capacity c_B of the bottleneck link. We therefore alternatively describe instance $\mathcal{I}$ as $\mathcal{I} = (S,T,H,c_B)$. An example appears in Figure 2.3.

Note that by [13] there is no loss of generality in restricting to index coding instances in canonical form (i.e., there are k sources, and each source is demanded by l terminals and each terminal only demands one source).

Chapter 3

PRELIMINARIES

In this chapter, we introduce two key concepts to our work: reduction and the edge removal statement.

In this work, we use reduction to understand when solutions and results for one type of network information theory can be used for another type of network information theory problem. In Section 3.1, we formally describe reduction in two different settings: reduction in code design and reduction in capacity region. We also discuss some of the existing work in these two regimes.

One of the major contributions of our work is to show that in some cases the connection between code reduction and capacity reduction hinges on the edge removal statement. Determining whether of not the edge removal statement holds may be considered a canonical open problem in information theory and is shown to be connected to other open problems in information theory. We introduce the edge removal statement in Section 3.2 and give some background on its development. We also introduce the cooperation facilitator and the broadcast facilitator, which are important components in the proofs of our work.

3.1 Reductions in Networks

Our work relies on a technique called *reduction.* When a problem $\mathcal{A}$ *reduces* to problem $\tilde{\mathcal{A}}$, it means that problem $\mathcal{A}$ can be solved by first *mapping* it to a *corresponding* problem $\tilde{\mathcal{A}}$, then applying a solution for $\tilde{\mathcal{A}}$, and finally mapping the solution for $\tilde{\mathcal{A}}$ back to a solution for $\mathcal{A}$ (See Figure 3.1.)

Reduction is a very powerful technique because it allows us to draw connections between problems even when the solutions to both $\mathcal{A}$ and $\tilde{\mathcal{A}}$ are unknown; if $\mathcal{A}$ reduced to $\tilde{\mathcal{A}}$, then solving problem $\mathcal{A}$ suffices to find a solution to $\mathcal{A}$. Reduction can be used to show that a communication problem is easy if it can be reduced to other problems for which solutions are well understood. Reduction can also be used to show that a problem $\tilde{\mathcal{A}}$ is hard by showing that a difficult open problem can be reduced to $\tilde{\mathcal{A}}$.

We employ two distinct notions of reduction in this work. In each, some class

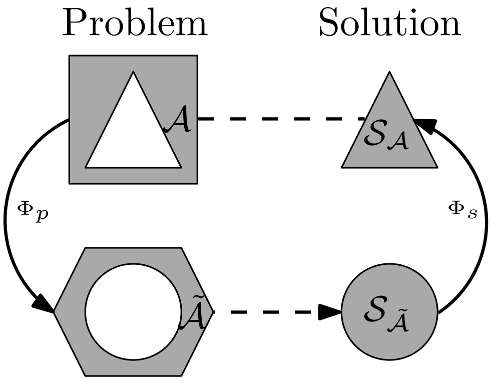


Figure 3.1: In this figure, two different problems and their respective solutions are represented by different shapes. If $\mathcal{A}$ reduces to $\tilde{\mathcal{A}}$, then there exist mappings Φ_p and Φ_s such that $\mathcal{A}$ can be solved by first mapping it to a corresponding problem $\tilde{\mathcal{A}} = \Phi_p(\mathcal{A})$, solving $\tilde{\mathcal{A}}$, and then mapping the solution $\mathcal{S}_{\tilde{\mathcal{A}}}$ for $\tilde{\mathcal{A}}$ back to a solution $\mathcal{S}_{\mathcal{A}} = \Phi_s(\mathcal{S}_{\tilde{\mathcal{A}}})$ for $\mathcal{A}$.

of problems on a family $\mathcal{P}$ of network coding instances is shown to be solvable through solution to the same class of problems on a different family $\tilde{\mathcal{P}}$ of network coding instances. We begin by describing both types of reductions and then give a brief background of the edge removal statement, which plays a central role in our derivation of new reduction results.

Code and Capacity Reduction

Consider two families of network coding instances, $\mathcal{P}$ and $\tilde{\mathcal{P}}$.

Definition 1 (Code Reduction)**.** *We say that* code design for $\mathcal{P}$ reduces to code design for $\tilde{\mathcal{P}}$ *if there exist the following two mappings:*

1. *a mapping from any instance $\mathcal{I} \in \mathcal{P}$ and code parameter triple $(\mathbf{R}, \epsilon, n)$ to an instance $\tilde{\mathcal{I}} \in \tilde{\mathcal{P}}$ and triple $(\tilde{\mathbf{R}}, \tilde{\epsilon}, \tilde{n})$ such that $\mathcal{I}$ is $(\mathbf{R}, \epsilon, n)$-feasible if and only if $\tilde{\mathcal{I}}$ is $(\tilde{\mathbf{R}}, \tilde{\epsilon}, \tilde{n})$-feasible.*

2. *a mapping from any $(\tilde{\mathbf{R}}, \tilde{\epsilon}, \tilde{n})$-feasible solution for $\tilde{\mathcal{I}}$ to a corresponding $(\mathbf{R}, \epsilon, n)$-feasible solution for $\mathcal{I}$.*

Thus if code design for $\mathcal{P}$ reduces to code design for $\tilde{\mathcal{P}}$, then solution of the code design problem for all networks in $\tilde{\mathcal{P}}$ would solve the code design problem for all networks in $\mathcal{P}$. Further, only one code design for one parameter vector is required in $\tilde{\mathcal{P}}$ to yield one code design for one parameter vector in $\mathcal{P}$. Theorems 3.1.1–3.1.3 give examples of code reductions. If the mappings described in 1) and 2) of Definition 1 are efficient then we say that the code reduction is efficient.

Multiple multicast networks are networks in which every source message is required by one or more receivers. Multiple unicast networks are a sub-class of multiple multicast networks in which every source is required by exactly one receiver. A network with k unicasts is also called a k-unicast network. Theorem 3.1.1 from [12] proves a code reduction from multiple multicast networks ($\mathcal{P}$) to multiple unicast networks ($\tilde{\mathcal{P}}$). This result proves that code design for the class of multiple multicast networks ($\mathcal{P}$) can be solved by solving code design for a subset of that class ($\tilde{\mathcal{P}} \subseteq \mathcal{P}$).

Theorem 3.1.1 (Code Reduction from Multiple Multicast to Multiple Unicast Network Coding[12, Theorem II.1])**.** *Multiple multicast network code design reduces to multiple unicast network code design.*

In some cases, code reduction results are known under restrictive assumptions on the parameters of either or both families of networks. For example, Theorem 3.1.2 describes both the linear [6] and general [7] forms of the code reduction from network coding to index coding. While Theorem 3.1.3 describes the reduction from k-unicast to 2-unicast network coding when $\epsilon = 0$, giving a zero-error code reduction result.

Theorem 3.1.2 (Code Reductions from Network Coding to Index Coding)**.**

1. *[6, Theorem 5] Linear acyclic network code design reduces to linear index code design.*
2. *[7, Theorem 1] Acyclic network code design reduces to index code design.*

Theorem 3.1.3 (Zero-error Code Reduction from k-Unicast to 2-Unicast Network Coding [14, Theorem 1])**.** *Zero-error k-unicast network code design reduces to zero-error 2-unicast network code design.*

Other examples also include the code reduction from network error correction to multiple-unicast network coding [21], [22], and the code reduction from secure network coding to multiple-unicast network coding [23].

Definition 2 (Capacity Reduction)**.** *We say that* capacity characterization for $\mathcal{P}$ reduces to capacity characterization for $\tilde{\mathcal{P}}$ *if there exists a mapping from* *any instance $\mathcal{I} \in \mathcal{P}$ and rate vector $\mathbf{R}$ to an instance $\tilde{\mathcal{I}} \in \tilde{\mathcal{P}}$ and rate vector $\tilde{\mathbf{R}}$ such that*

$$\mathbf{R} \in \mathcal{R}(\mathcal{I}) \Leftrightarrow \tilde{\mathbf{R}} \in \mathcal{R}(\tilde{\mathcal{I}}).$$

Here $\mathcal{R}(\cdot)$ is used as notational shorthand to describe a capacity region or bound for a capacity region; the type used in any particular result is specified in the result. Capacity reduction from $\mathcal{A}$ to $\tilde{\mathcal{A}}$ demonstrates how characterizing the capacity regions for all networks in $\tilde{\mathcal{A}}$ would characterize the capacity regions for all networks in $\mathcal{A}$. Further, solving a single question of the form "Is $\mathbf{R}$ in set $\mathcal{R}(\mathcal{I})$?" requires the solution of only a single question of the form "Is $\tilde{\mathbf{R}}$ in set $\mathcal{R}(\tilde{\mathcal{I}})$?". Similarly, if the mappings described in Definition 2 are efficient then we say that the capacity reduction is efficient. In some cases, reduction results are known under restrictive assumptions on the parameters or capacity regions of either or both families of networks. Theorem 3.1.4 proves a linear capacity reduction from multiple multicast index coding to multiple unicast index coding.

Theorem 3.1.4 (Linear Capacity Reduction from Multiple Multicast to Multiple Unicast Index Coding[13, Theorem 2])**.** *Multiple multicast index coding linear capacity calculation reduces to multiple unicast index coding linear capacity calculation.*

Does Code Reduction Imply Capacity Reduction?

The study of capacity reduction is motivated by the goal of understanding an efficient way to compute the capacity of $\mathcal{I}$ based on knowledge of how to compute the capacity region of $\tilde{\mathcal{I}}$. So far, we have seen quite a few reductions in code design [6], [7], [12], [14], but not all of these reductions have a corresponding known capacity reduction. A central question of this work is whether code reductions can be used to derive corresponding capacity reductions.

Consider the following region:

$$\mathcal{R}^*_\epsilon(\mathcal{I}) = \{\mathbf{R} : \forall\epsilon > 0, \mathcal{I} \text{ is } (\mathbf{R}, \epsilon, n)\text{-feasible infinitely often in } n\}.$$

By definition of the capacity region, $\mathcal{R}_\epsilon(\mathcal{I}) = \overline{\mathcal{R}^*_\epsilon(\mathcal{I})}$. If code reduction holds, then the knowledge of $\mathcal{R}^*_\epsilon(\tilde{\mathcal{I}})$ would imply the knowledge of $\mathcal{R}^*_\epsilon(\mathcal{I})$, whose closure is $\mathcal{R}_\epsilon(\mathcal{I})$. It is therefore tempting to believe that code reduction from $\mathcal{A}$ to $\tilde{\mathcal{A}}$ implies capacity reduction from $\mathcal{A}$ to $\tilde{\mathcal{A}}$, yet, no such result is known in general. One key obstacle is that the knowledge of $\mathcal{R}_\epsilon(\tilde{\mathcal{I}})$ does not provide enough information about $\mathcal{R}^*_\epsilon(\tilde{\mathcal{I}})$: $\mathbf{R}$ may get mapped to an $\tilde{\mathbf{R}}$ that falls on the boundary of $\mathcal{R}(\tilde{\mathcal{I}})$, which then leaves open the question of whether or not $\tilde{\mathbf{R}}$ is in $\mathcal{R}^*_\epsilon(\tilde{\mathcal{I}})$.

As a result, reductions in capacity characterization exist (see, for example, [24, Theorem 3]), but they remain relatively rare. Our work shows that for the scenarios to date where code reductions were not accompanied by corresponding capacity reductions, bridging the gap between code reduction and capacity reduction relies in some fundamental way on understanding how small changes in a network coding instance affect the capacity of that instance. The next section describes the edge removal statement, determining whether this statement holds in general is an example question in that domain.

3.2 The Edge Removal Statement

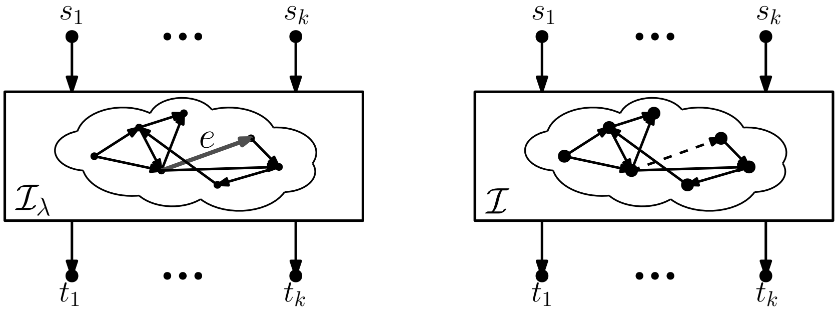


Figure 3.2: Networks $\mathcal{I}$ and $\mathcal{I}_\lambda$ differ by an edge e of capacity λ.

In this section, we focus on acyclic network coding instances[1]. The edge removal question studies the change in network coding capacity that results when a single edge of capacity λ is removed from a network coding instance [8]. Specifically, let $\mathcal{I}_\lambda = (G_\lambda, S, T)$ be a network coding instance containing an edge e_λ of capacity λ. Let $\mathcal{I} = (G, S, T)$ be the network coding instance that results when edge e_λ is removed from graph G_λ. The edge removal statement compares the capacity regions $\mathcal{R}(\mathcal{I})$ and $\mathcal{R}(\mathcal{I}_\lambda)$. In particular, the literature explores a variety of questions of the form

$$\text{Does } \mathbf{R} \in \mathcal{R}(\mathcal{I}_\lambda) \text{ imply } \mathbf{R} - f(\lambda) \in \mathcal{R}(\mathcal{I})?$$

[1] An acyclic network coding instance is a network coding instance with an underlying graph that does not contain any directed cycle.

Here, $f(\cdot)$ is some function of λ. This question has an increasingly rich history [4], [8], [19], [25]–[30], but remains unsolved in general. While this problem seems deceptively simple, it is deeply connected to many other fundamental properties of the capacity region. For example, in [19], the authors connected the edge removal statement with the "dependent source coding problem," which studies the change in capacity region when we allow the source messages to be dependent. In [25], the same authors show that determining whether or not the edge removal statement holds is equivalent to the "zero vs epsilon error problem," which studies the change in the capacity region when we require the source messages to be communicated without error. It is also known to be related to the strong converse problem [28].

Further, as we show in this work, the edge removal statement is also connected to a series of reduction results. Thus, deciding whether or not the edge removal statement is true may be considered a canonical problem in network coding in the sense that obtaining the answer to any one of them would yield answers to the rest. Results to date include complete solutions for a variety of special cases. Examples of two such results follow.

The edge removal question is solved in the case of linear codes on both acyclic networks and networks containing cycles [8]. In this case, $f(\lambda) = \lambda$; that is, $f(\lambda)$ is a vector with λ in every dimension, giving the following result.

Theorem 3.2.1 ([8, Section V.D])**.** *For any acyclic network coding instance $\mathcal{I}_\lambda$,*

$$\mathbf{R} \in \mathcal{R}^L_\epsilon(\mathcal{I}_\lambda) \quad \Rightarrow \quad \mathbf{R} - \lambda \in \mathcal{R}^L_\epsilon(\mathcal{I}).$$

For this work, we focus on an asymptotic version of the edge removal statement, where we seek to understand whether an edge of negligible capacity can have a non-negligible impact on network coding capacity. This variation is directly related to the vanishment conjecture [4], which studies the continuity of the capacity region with respect to the capacity of edges at value 0.

Definition 3 (Asymptotic Edge Removal Statement (AERS))**.** *For any acyclic network coding instances $\mathcal{I}_\lambda$ and $\mathcal{I}$ differing in a single edge e_λ of capacity λ,*

$$\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}).$$

The limit in the AERS is guaranteed to exist by Theorem 3.2.2.

Theorem 3.2.2. *The limit* $\lim\limits_{\lambda\to 0}\mathcal{R}_\epsilon(\mathcal{I}_\lambda)$ *exists.*

Proof. Let $\{\lambda_n\}_{n=1}^{\infty}$ be a monotonically decreasing sequence tending to zero, then

$$\lim_{\lambda\to 0}\mathcal{R}_\epsilon(\mathcal{I}_\lambda) = \lim_{n\to\infty}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_n}).$$

Since $\lambda_1 > \lambda_2 > \cdots$ implies $\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_1}) \supseteq \mathcal{R}_\epsilon(\mathcal{I}_{\lambda_2}) \supseteq \cdots \supseteq \mathcal{R}_\epsilon(\mathcal{I}_{\lambda_n})$, we have

$$\operatorname*{limsup}_{n\to\infty}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_n}) = \bigcap_{n\geq 1}\bigcup_{j\geq n}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_j}) = \bigcap_{n\geq 1}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_n})$$

and

$$\operatorname*{liminf}_{n\to\infty}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_n}) = \bigcup_{n\geq 1}\bigcap_{j\geq n}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_j}) = \bigcap_{j\geq 1}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_j}).$$

Therefore $\operatorname*{limsup}\limits_{n\to\infty}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_n}) = \operatorname*{liminf}\limits_{n\to\infty}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_n})$, and the limit exists and equals $\bigcap\limits_{n\geq 1}\mathcal{R}_\epsilon(\mathcal{I}_{\lambda_n})$. □

The asymptotic edge removal question is solved in the case of super-source networks and networks with co-located sources. A network with co-located sources is a network where all sources originate from a single source node[2]. A super-source network is one whose sources are not co-located but there is a super-node in the network that has both full knowledge of all the information present at the sources and low capacity outgoing edges connecting it with each and every one of the source nodes [26], [30] (See Figure 3.3). In the following, we define two variations of super source networks.

Cooperation Facilitator

Instance $\mathcal{I}_\lambda^{cf} = (G_\lambda^{cf}, S^{cf}, T^{cf})$ is obtained from $\mathcal{I} = (G, S, T,)$ by adding a cooperation facilitator to the network (Figure 3.3). Graph $G_\lambda^{cf} = (V^{cf}, E^{cf}, C_\lambda^{cf} = \{c_e^{cf}, e \in E^{cf}\})$ is obtained from $G = (V, E, C)$ by adding k new source nodes $s_1', \cdots, s_k'$, a super source node s_{su}, a relay node s_{re}, $2k$ infinite capacity links $\{(s_i', s_i), (s_i', s_{su})\}_{i\in[k]}$, k links $\{(s_{re}, s_i)\}_{i\in[k]}$ and a bottleneck link $b = (s_{su}, s_{re})$ of capacity λ. The source edges of the original instance $\mathcal{I}$ are replaced with

[2] While a network with co-located sources is not considered canonical, it can be easily converted to an equivalent canonical network [13, Footnote 5].

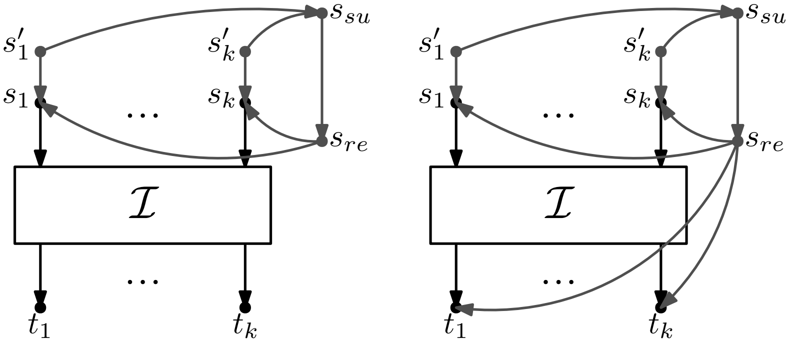


Figure 3.3: The left network shows $\mathcal{I}$ augmented with a cooperation facilitator ($\mathcal{I}_\lambda^{cf}$). The right network shows $\mathcal{I}$ augmented with a broadcast facilitator ($\mathcal{I}_\lambda^{bf}$).

links of capacity $\sum_{e'\in E_{S^c}} c_{e'}$. Thus, the resulting graph is defined by

$$\begin{aligned} V^{cf} &= V \cup \{s_{su}, s_{re}\} \cup \left[\bigcup_{i\in[k]} \{s_i'\}\right] \\ E^{cf} &= E \cup \{(s_{su}, s_{re})\} \cup \left[\bigcup_{i\in[k]} \{(s_i', s_{su}), (s_i', s_i), (s_{re}, s_i)\}\right] \\ c_e^{cf} &= \begin{cases} c_e & \text{if } e \in E_{S^c} \\ \sum_{e'\in E_{S^c}} c_{e'} & \text{if } e \in E_S \\ \lambda & \text{if } \mathrm{In}(e) \in \{s_{su}, s_{re}\} \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

The old source nodes no longer hold any source random variable; hence,

$$S^{cf} = \bigcup_{i\in[k]} \{s_i'\}.$$

The set of terminal nodes remains the same; hence,

$$T^{cf} = T.$$

Each terminal t_i now demands s_i' instead of s_i.

Broadcast Facilitator

Instance $\mathcal{I}_\lambda^{bf} = (G_\lambda^{bf}, S^{bf}, T^{bf})$ is obtained from $\mathcal{I} = (G, S, T)$ by adding a broadcast facilitator (Figure 3.3). Similar to the cooperation facilitator, the broadcast facilitator is a supersource node that receives all source messages and computes a function of them; however, instead of sending the computed value to the source nodes in S, it broadcasts it to *all* the nodes in V. Thus, $\mathcal{I}_\lambda^{bf}$ can also be also be obtained from $\mathcal{I}_\lambda^{cf}$ by adding $|V|-k$ broadcast links of capacity λ. Therefore, defining the instance $\mathcal{I}_\lambda^{bf}$ that contains the broadcast facilitator by comparison to the instance $\mathcal{I}_\lambda^{cf}$ that contains the cooperation facilitator, we have

$$\begin{aligned}
V^{bf} &= V^{cf} \\
E^{bf} &= E^{cf} \cup \left[\bigcup_{v \in V} \{(s_{re}, v)\} \right] \\
c_e^{bf} &= \begin{cases} c_e^{cf} & \text{if } e \in E^{cf} \\ \lambda & \text{otherwise.} \end{cases}
\end{aligned}$$

For a fixed bottleneck rate λ, the capacity region of $\mathcal{I}_\lambda^{bf}$ is a superset of that of $\mathcal{I}_\lambda^{cf}$. Theorem 3.2.3 summarizes what is known about the edge removal question in these scenarios.

Theorem 3.2.3 (Asymptotic Edge Removal Property for Co-located Source and Super-source Networks [26, Theorem 2, Proposition 5])**.** *For any acyclic network coding instances $\mathcal{I}_\lambda$ and $\mathcal{I}$ differing in a single edge e_λ of capacity λ, if $\mathcal{I}$ is a co-located source or super-source network, then*

$$\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}).$$

While [4, Definition 3.10] and [26, Conjecture 2] conjecture that the asymptotic edge removal statement is always true, even this very limited case of the edge removal question remains unproven in general. The question turns out to be intertwined with a variety of other open questions. For example, [19] shows that solving the asymptotic edge removal question would suffice to prove whether asymptotic and zero-error definitions of network coding capacity ever yield different rate regions. Similarly, [25] shows that solving the same question would determine whether asymptotically negligible source dependence can change the network capacity region.

Chapter 4

CAPACITY REDUCTION FROM MULTIPLE MULTICAST TO MULTIPLE UNICAST

The materials in this chapter are published in part as [31], [32].

In this chapter, we explore the question: how central is the demand type of a given network to the difficulty of characterizing its capacity region. For general networks, even the single unicast case remains open. For example, the single unicast relay channel [33] has only been solved for the degraded case [34]. For network coding, the single multicast capacity region has been solved [35]–[37], but the general capacity region remains elusive. Our work aims to identify a restrictive demand type that is representative of the general case in the realms of memoryless networks, network coding networks, and index coding networks.

Using the notion of capacity reduction, we show that multiple multicast demands reduce to multiple unicast demands. Our result implies that it suffices to study the capacity regions of multiple unicast networks to obtain full understanding of the capacity regions of multiple multicast networks. Our results hold for both general and linear coding capacity. As a corollary of our main result, we also obtain capacity reductions in the settings of network coding and index coding. As such our work generalizes and unifies previous works [12], [13] (Section 3.1.)

In what follows, we first describe the reduction mapping for the capacity reduction from multiple multicast to multiple unicast networks. The reductions for both the general capacity and the linear capacity region use the same reduction mapping. We present our main result in Section 4.2. The proof of the main theorem appears in Section 4.6. Our proof relies on a random linear channel outer coding argument; this technique is presented in Section 4.5.

4.1 Reduction Mapping Φ_1

One of the key challenges in proving a reduction result is to design a mapping from each instance $\mathcal{I} \in \mathcal{A}$ and rate vector $\mathbf{R}$ to its corresponding instance $\tilde{\mathcal{I}} \in \tilde{\mathcal{A}}$ and rate vector $\tilde{\mathbf{R}}$ in a way that preserves the properties needed for the desired reduction. For our purposes, those conditions are described in

Definition 2 in Section 3.1.

We begin by describing our mapping a multiple multicast instance $\mathcal{I}$ and its rate vector $\mathbf{R}$ to a multiple unicast instance $\tilde{\mathcal{I}}$ and its rate vector $\tilde{\mathbf{R}}$. We use tildes on all variables corresponding to the multiple unicast communication network instance in order to distinguish the two instances from each other. Our reduction mapping is inspired by but not identical to the constructions found in [12], [13]. The multiple unicast communication network instance

$$\tilde{\mathcal{I}} = (\tilde{S}, \tilde{T}, \tilde{U}, \tilde{H}, \tilde{p}(\tilde{\mathbf{Y}}|\tilde{\mathbf{X}}))$$

is constructed by augmenting $\mathcal{I} = (S, T, U, H, p(\mathbf{Y}|\mathbf{X}))$ with "butterfly" like structures. This is achieved by replacing source nodes S in $\mathcal{I}$ with kl new source nodes $\tilde{S}$ in $\tilde{\mathcal{I}}$ and replacing has set H in $\mathcal{I}$ with new has set $\tilde{H}$ in $\tilde{\mathcal{I}}$. In the following, we describe the formal construction of $(\tilde{\mathcal{I}}, \tilde{\mathbf{R}})$ and provide some intuition. An example appears in Figure 4.1.

Define an index mapping function $\beta(i,j) = (i-1)l + j$. For $\mathbf{R} = (R_1, \cdots, R_k)$, each old source node s_i is replaced by l new source nodes $\tilde{s}_{\beta(i,1)}, \cdots, \tilde{s}_{\beta(i,l)}$, where each $\tilde{s}_{\beta(i,1)}$ carries a rate-R_i independent source message variable $\tilde{W}_{\beta(i,j)}$. Thus

$$\tilde{S} = \bigcup_{i\in[k]} \bigcup_{j\in[l]} \left\{ \tilde{s}_{\beta(i,j)} \right\}.$$

Each old terminal node $t_{i,j}$ in $\mathcal{I}$ is relabeled as $\tilde{t}_{\beta(i,j)}$ in $\tilde{\mathcal{I}}$, it now demands source message $\tilde{W}_{\beta(i,j)}$; thus

$$\tilde{T} = \bigcup_{i\in[k]} \bigcup_{j\in[l]} \left\{ \tilde{t}_{\beta(i,j)} \right\}.$$

and

$$\tilde{\mathbf{R}} = (R_1^l, \cdots, R_k^l),$$

where each $R_i^l = (R_i, \cdots, R_i)$ is an l-dimensional vector with R_i in each component. Each relay node $u \in U$ is relabeled as $\tilde{u}$ in $\tilde{\mathcal{I}}$; thus,

$$\tilde{U} = \bigcup_{u\in U} \left\{ \tilde{u} \right\}.$$

The channel transition function for $\tilde{\mathcal{I}}$ remains unchanged apart from the nodes being relabeled, it is defined as follows,

$$\begin{aligned}\tilde{\mathcal{Y}}_{\tilde{u}} &= \mathcal{Y}_u,\ \forall u \in U\\ \tilde{\mathcal{Y}}_{\tilde{t}_{\beta(i,j)}} &= \mathcal{Y}_{t_{i,j}},\ \forall i \in [k], j \in [l]\\ \tilde{p}(\tilde{\mathbf{Y}} = \mathbf{y}|\tilde{\mathbf{X}} = \mathbf{x}) &= p(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x}),\ \forall \mathbf{x} \in \prod_{u\in U} \mathcal{X}_u, \mathbf{y} \in \prod_{v\in T\cup U} \mathcal{Y}_v.\end{aligned}$$

The remainder of $\tilde{\mathcal{I}}$ is designed to enable it to "reuse" the encoders and decoders from $\mathcal{I}$. For each relay node $u \in U$ and each source $s_i \in H_u$ available to node u, node u first combines the l independent sources $(\tilde{W}_{\beta(i,1)}, \cdots, \tilde{W}_{\beta(i,l)})$ into a single "*mixed source*" variable $\tilde{W}_i^{\text{sum}}$ before applying the encoder from $\mathcal{I}$ to the "mixed sources" variable $\tilde{W}_i^{\text{sum}}$. To enable the the combination of $(\tilde{W}_{\beta(i,1)}, \cdots, \tilde{W}_{\beta(i,l)})$ into $\tilde{W}_i^{\text{sum}}$, for each relay node $u \in U$ that has access to source from node s_i in $\mathcal{I}$, the corresponding relay node $\tilde{u}$ in $\tilde{\mathcal{I}}$ is given access to the source from nodes $\tilde{s}_{\beta(i,1)}, \cdots, \tilde{s}_{\beta(i,l)}$ in $\tilde{\mathcal{I}}$, giving

$$\tilde{H}_{\tilde{u}} = \bigcup_{i:s_i\in H_u} \bigcup_{j\in[l]} \{\tilde{s}_{\beta(i,j)}\}.$$

In the decoding phase, each terminal $\tilde{t} \in \tilde{T}$ uses the decoder from the corresponding node $t \in T$ to reconstruct the "mixed source." In order to enable node $\tilde{t}_{\beta(i,j)}$ to extract from the mixed source $\tilde{W}_i^{\text{sum}}$ its desired component $\tilde{W}_{\beta(i,j)}$, we provide each terminal $\tilde{t}_{\beta(i,j)}$ with just enough "*side information,*" $(\tilde{W}_{\beta(i,j')}, j' \in [l] \setminus \{j\})$, to solve for message $\tilde{W}_{\beta(i,j)}$. Thus, for each $\tilde{t} = \tilde{t}_{\beta(i,j)} \in \tilde{T}$,

$$\tilde{H}_{\tilde{t}} = \left[\bigcup_{i':s_{i'}\in H_{\tilde{t}}} \bigcup_{j'\in[l]} \{\tilde{s}_{\beta(i',j')}\}\right] \cup \left[\bigcup_{j''\in[l]\setminus\{j\}} \{\tilde{s}_{\beta(i,j'')}\}\right].$$

4.2 Main Result

Here, we state our main capacity reduction result for both the general and the linear capacity regions. The theorem relies on the mapping Φ_1 defined in Section 4.1.

Theorem 4.2.1 (Capacity Reduction from Multiple Multicast to Multiple Unicast Communication Networks)**.**

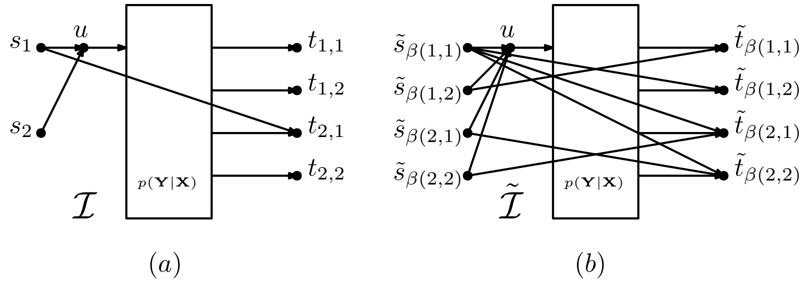


Figure 4.1: (a) A 2 by 2 communication network $\mathcal{I}$. (b) A 4-unicast communication network $\tilde{\mathcal{I}}$ corresponding to $\mathcal{I}$ in (a).

1. *Calculating the multiple multicast communication network capacity region reduces to calculating the multiple unicast communication network capacity region. That is, under mapping Φ_1, for any communication network instance $\mathcal{I}$ and rate vector $\mathbf{R}$,*

$$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}).$$

2. *Calculating the multiple multicast communication network linear capacity region reduces to calculating the multiple unicast communication network linear capacity region. That is, under mapping Φ_1, for any communication network instance $\mathcal{I}$ and rate vector $\mathbf{R}$,*

$$\tilde{\mathbf{R}} \in \mathcal{R}^L_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}^L_\epsilon(\mathcal{I}).$$

Proof. See Section 4.6. □

4.3 Implications for Network Coding Networks

Recall that a network coding network $\mathcal{I} = (G, S, T)$ is a communication network instance consisting of independent point-to-point noiseless links. Therefore, Theorem 4.2.1 may be applied directly to any multiple multicast network coding problem $\mathcal{I}$ to obtain a corresponding multiple unicast network $\tilde{\mathcal{I}}$.

Corollary 4.3.1 (Capacity Reduction from Multiple Multicast to Multiple Unicast Network Coding)**.**

1. *Calculating the multiple multicast network coding capacity region reduces to calculating the multiple unicast network coding capacity region. That is, under mapping Φ_1, for any network coding instance $\mathcal{I}$ and rate vector $\mathbf{R}$,*

$$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}).$$

2. *Calculating the multiple multicast network coding linear capacity region reduces to calculating the multiple unicast network coding linear capacity region. That is, under mapping Φ_1, for any network coding instance $\mathcal{I}$ and rate vector* $\mathbf{R}$*,*

$$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon^L(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon^L(\mathcal{I}).$$

Proof. Since the construction in Section 4.1 only modifies source nodes S and has sets H, $\mathcal{I}$ is a network coding instance then $\tilde{\mathcal{I}}$ is also a network coding network instance. Theorem 4.2.1 thus gives a reduction in capacity region (and linear capacity region) from multiple multicast network coding to multiple unicast network coding. □

4.4 Implications for Index Coding Networks

An index coding network $\mathcal{I} = (S, T, H, c_B)$ is a network coding network with a graph G falling in a restricted class of possible network coding topologies and is therefore a communication network. We therefore apply Theorem 4.2.1 to obtain the following corollary.

Corollary 4.4.1 (Capacity Reduction from Multiple Multicast to Multiple Unicast Index Coding)**.**

1. *Calculating the multiple multicast index coding capacity region reduces to calculating the multiple unicast index coding capacity region. That is, under mapping Φ_1, for any index coding instance $\mathcal{I}$ and rate vector* $\mathbf{R}$*,*

$$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}).$$

2. *Calculating the multiple multicast index coding linear capacity region reduces to calculating the multiple unicast index coding linear capacity region. That is, under mapping Φ_1, for any index coding instance $\mathcal{I}$ and rate vector* $\mathbf{R}$*,*

$$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon^L(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon^L(\mathcal{I}).$$

Proof. Since mapping Φ_1 only modifies the "has" set of $\mathcal{I}$, applying mapping Φ_1 to a multiple multicast index coding network $\mathcal{I}$ yields a multiple unicast index coding network $\tilde{\mathcal{I}}$. Thus, Theorem 4.2.1 holds for index coding networks. □

4.5 A Linear Code Reduction from Lossy Network Coding to Lossless Network Coding

The following result extends ideas from [24, Theorem 2] and [38, Section V.B]. Precisely, Lemma 4.5.1 shows that a lossy code can be mapped to a lossless code with a reduced rate using channel outer coding. Lemma 4.5.1 differs from [24, Theorem 2] and [38, Section V.B] in that we show the existence of a linear outer code rather than one that may be non-linear. The term lossy network code is here used to describe a set of blocklength-n encoders $\{f_{u,\tau}, u \in U\}$ for the relay nodes of a communication network. Rather than considering a sequence of codes that meets a certain asymptotic error constraint as in lossless network coding, we here consider a code that meets a constraint on the mutual information. Specifically, we consider the scenario where operation of the encoders $\{f_{u,\tau}, u \in U\}$ on a set of independent sources $\mathbf{W}$ yields a mutual information $I(W_t; Z_t) \geq nR_t$ for each terminal t and its desired source W_t where $Z_t = (Y_t^n, W_{H_t})$ is the channel output for terminal node t.

Lemma 4.5.1. *Let*

$$\mathcal{I} = (S, T, U, H, p(\mathbf{Y}|\mathbf{X}))$$

be a k by l communication network instance where the alphabet of the received information at $t_{i,j}$ is $\mathcal{Z}_{t_{i,j}} = \mathbb{F}_2^{nq_{i,j}}$, for all $(i,j) \in [k] \times [l]$. Suppose that there exist blocklength-n encoding functions $\{f_{u,\tau}\}_{\tau\in[n]}$ for each relay node $u \in U$ such that if sources W_i are uniformly distributed on $\mathcal{W}_i = \mathbb{F}_2^{nm_i}$ for each $i \in [k]$, then under the operation of $\{f_{u,\tau}\}_{u\in U}^{\tau\in[n]}$, the channel output $Z_{t_{i,j}} = (Y_{t_{i,j}}^n, W_{H_{t_{i,j}}})$ for the each of the terminals satisfies

$$I(W_i; Z_{t_{i,j}}) \geq nR_i$$

for each terminal $t_{i,j}$, $(i,j) \in [k] \times [l]$. Then for any rate vector $\mathbf{R}' = (R_1', ..., R_k')$ such that $R_i' < R_i$, for all i and any $\epsilon' > 0$, there exists a blocklength n' such that $\mathcal{I}$ is $(\mathbf{R}', \epsilon', n')$-feasible. Further, if the encoders $\{f_{u,\tau}\}_{u\in U}^{\tau\in[n]}$ are linear functions, then $\mathcal{I}$ is $(\mathbf{R}', \epsilon', n')$-linearly feasible.

Proof. As in [24], [38], this result is proved using a random channel outer coding argument, we extend this idea by applying a random *linear* outer coding argument instead to prove a linear code reduction when the encoders $\{f_{u,\tau}, u \in U\}$ are linear. This is achieved by applying the random *linear* outer coding argument for point-to-point channels from [39] simultaneously to all pairs of demands in $\mathcal{I}$.

In the following, we show that the multicast $(R_1', \cdots, R_k')$ rates can be achieved by a linear random channel outer coding argument. We begin with an outline of the proof before providing the details. We first maps source message W_i', $i \in [k]$, to a random blocklength N codeword $\mathbf{W}_i^N(W_i')$ using a set of random linear outer encoding functions. Each terminal $t_{i,j}$, $(i,j) \in [k] \times [l]$ then decodes a reconstruction $\widehat{W_i'} = w_i'$ if w_i is jointly typical with the received symbol $Z_{t_{i,j}}^N$. The overall blocklength is therefore nN. We show that there exists a set of linear outer encoding functions that yields a small error probability as N tends to infinity.

1. *Random linear code generation.* For a blocklength N, a random codeword $\mathbf{W}_i^N(w_i')$ for each source message realization $w_i' \in \mathbb{F}_2^{nNR_i'}$ is generated according to

$$\mathbf{W}_i^N(w_i') = w_i' M_i,$$

where w_i' is a length-nNR_i' row vector over $\mathbb{F}_2$ and M_i is an nNR_i' by nNm_i random matrix such that each entry is selected independently and uniformly at random from $\mathbb{F}_2$.

2. *Codeword distribution analysis.* Let r_j be the jth row of M_i. Then each r_j is a random row vector over $\mathbb{F}_2$. Denote by $w_{i,j}'$ the jth component of w_i' and denote by $B(w_i')$ the positions of 1's in w_i'; thus

$$B(w_i') = \Big\{ j \in [nNR_i'] : w_{i,j}' = 1 \Big\},$$
$$w_i' M_i = \sum_{j \in B(w_i')} r_j.$$

Thus, for any $w_i' \neq \mathbf{0}^{nNR_i'} = (0, ..., 0)$, the codeword for each w_i' is uniformly distributed over $\mathbb{F}_2^{nNm_i}$. For any distinct, non-zero w_i' and w_i^*, consider the following decomposition:

$$w_i' M_i = \sum_{j \in B(w_i') \cap B(w_i^*)} r_j + \sum_{j \in B(w_i') \backslash B(w_i^*)} r_j.$$

$$w_i^* M_i = \sum_{j \in B(w_i') \cap B(w_i^*)} r_j + \sum_{j \in B(w_i^*) \backslash B(w_i')} r_j.$$

Since $\sum_{j \in B(w_i') \backslash B(w_i^*)} r_j$ and $\sum_{j \in B(w_i^*) \backslash B(w_i')} r_j$ are independent and uniformly distributed in $\mathbb{F}_2^{nNm_i}$, the codewords for any distinct, nonzero pair of w_i and w_i' are therefore also (pairwise) independent and uniformly distributed in $\mathbb{F}_2^{nNR_i}$.

3. *Encoders for each $u \in U$.* Each relay node u first applies the random linear outer code in 1) to obtain $(\mathbf{W}_i^N(W_i'), s_i \in H_u)$. For each $i \in [k]$, divide codeword W_i^N into N chunks, i.e., $(W_{i,1}, \cdots, W_{i,N})$ such that each $W_{i,q}$ is in $\mathbb{F}_2^{nm_i}$ for each $q \in [N]$. Next, apply the blocklength n encoders $\{f_{u,\tau}\}_{\tau\in[n]}$ sequentially to each of the N chunks, namely, $(W_{i,q}, s_i \in H_u)$ for each $q \in [N]$. This takes a total of nN time steps.

4. *Joint typicality decoder.* At each terminal $t_{i,j}$, the decoded message estimate $\widehat{W'}_{i,j}$ equals w_i' if w_i' is non-zero (we address the case when $w_i' = \mathbf{0}$ separately since $\mathbf{0}M_i = \mathbf{0}$) and is the only source realization whose codeword is $\eta_{i,j}$-jointly typical with $\mathbf{Z}_{t_{i,j}}^N$, otherwise, the decoded message $\widehat{W'}_{i,j}$ equals the "error" symbol. Here, we pick $\eta_{i,j}$ such that $3\eta_{i,j} < R_i - R_i'$.

5. *Error analysis.* Each source message W_i' is drawn according to a uniform distribution over $\mathbb{F}_2^{nNR_i'}$. We use the jointly typical decoding as described above. Let $W_i' = w_i'$ where w_i' is a non-zero source message. Define the following events for $(i,j) \in [k] \times [l]$ and $w_i' \in \mathbb{F}_2^{nNR_i'}$:

$$E_{i,j}(w_i') = \{w_i' M_i \text{ is } \eta_{i,j}\text{-jointly typical with } \mathbf{Z}_{t_{i,j}}^N\}.$$

Let $\mathcal{E}_{i,j} = \{\widehat{W'}_{i,j} \neq W_i'\}$ denote the event that an error occur at $t_{i,j}$; then

$$\mathcal{E}_{i,j} = E_{i,j}^c(w_i') \cup \left[\bigcup_{w_i^* \in \mathbb{F}_2^{nNR_i'} \setminus \{\ \mathbf{0}, w_i'\}} E_{i,j}(w_i^*) \right].$$

We therefore have

$$\begin{aligned}\Pr(\mathcal{E}_{i,j} | W_i' = w_i') &= \Pr\left(E_{i,j}^c(w_i') \cup \left[\bigcup_{w_i^* \in \mathbb{F}_2^{nNR_i'} \setminus \{\ \mathbf{0}, w_i'\}} E_{i,j}(w_i^*) \right] \middle| W_i' = w_i' \right) \\ &\leq \Pr(E_{i,j}^c(w_i')) + \sum_{w_i^* \in \mathbb{F}_2^{nNR_i'} \setminus \{\ \mathbf{0}, w_i'\}} \Pr(E_{i,j}(w_i^*) | W_i' = w_i').\end{aligned}$$

The strong coding theorem for discrete memoryless channels [15, Theorem 5.6.2] upper bounds $\Pr(E_{i,j}^c(w_i'))$ by $2^{-nN\gamma_{i,j}}$ and $\Pr(E_{i,j}(w_i^*)|W_i' = w_i')$ by $2^{-I(\mathbf{W}_i^N;\mathbf{Z}_{t_{i,j}}^N)+nN3\eta_{i,j}}$, which gives

$$\Pr(\mathcal{E}_{i,j} | W_i' = w_i') \leq 2^{-nN\gamma_{i,j}} + 2^{nNR_i' - I(\mathbf{W}_i^N;\mathbf{Z}_{t_{i,j}}^N) + 3nN\eta_{i,j})}.$$

Note that this bound is independent of w_i' and is true for any non-zero w_i'. The error probability averaged over all code books is then computed as follows:

$$\begin{aligned}\Pr(\mathcal{E}_{i,j}) &= \sum_{w'_i \in \mathbb{F}_2^{nNR'_i}} \frac{1}{2^{nNR'_i}} \Pr(\mathcal{E}_{i,j} | W'_i = w'_i) \\ &= \frac{2^{nNR'_i} - 1}{2^{nNR'_i}} \Pr(\mathcal{E}_{i,j} | W'_i = w'_i) + \frac{1}{2^{nNR'_i}} \Pr(\mathcal{E}_{i,j} | W'_i = \mathbf{0}^{nNR'_i}) \\ &\leq \Pr(\mathcal{E}_{i,j} | W'_i = w'_i) + \frac{1}{2^{nNR'_i}}.\end{aligned}$$

By the union bound and for some positive constant η, the expected error probability for all kl decoders is at most $kl2^{-nN\eta}$, which goes to zero as N goes to infinity. This guarantees the existence of a good codebook that satisfies any error probability constraint.

□

4.6 Proof of Theorem 4.2.1

The proof of Theorem 4.2.1 relies on the mapping Φ_1 described in Section 4.1. We begin with a high level description of the proof.

The reductions for both the general capacity region and the linear capacity region use the same reduction network $\tilde{\mathcal{I}}$, which depends only on $\mathcal{I}$. To prove the "if and only if" statement in our theorem, we show that if a rate vector $\mathbf{R} = (R_1, \cdots, R_k)$ is in the capacity region of $\mathcal{I}$, then the corresponding rate vector $\tilde{\mathbf{R}} = (R_1^l, \cdots, R_k^l)$ is in the capacity region of $\tilde{\mathcal{I}}$. We then show the converse.

The proof employs a code reduction, in which we transform an $(\mathbf{R}, \epsilon, n)$ network code $\mathcal{C}$ for $\mathcal{I}$ into a rate $(\tilde{\mathbf{R}}(1-\rho), \tilde{\epsilon}, \tilde{n})$ network code $\tilde{\mathcal{C}}$ for $\tilde{\mathcal{I}}$ and vice versa. The loss in rate, δ, tends to zero as the blocklength, n, tends to infinity. By taking the closure of these rates, we get the desired results. We now present the proof of the assertion $\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}) \Leftrightarrow \tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}})$. The proof for linear capacity follows from that presented since our code reductions in both directions preserve linearity.

$\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}) \Rightarrow \tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}})$: Fix any $\epsilon, \delta > 0$. Our starting point is an $(\mathbf{R}(1-\rho), \epsilon, n)$-feasible network code $\mathcal{C}$ for $\mathcal{I}$. Our goal is to transform this code into an $(\tilde{\mathbf{R}}(1-\tilde{\rho}), \tilde{\epsilon}, \tilde{n})$-feasible network code $\tilde{\mathcal{C}}$ for $\tilde{\mathcal{I}}$. The idea is to augment $\mathcal{C}$ with a "butterfly" outer code. The outer code we use is linear. Thus, if we start with a $\mathcal{C}$ that is linear, the resulting $\tilde{\mathcal{C}}$ will also be linear. We formally describe $\tilde{\mathcal{C}}$ as follows (recall that $\beta(i,j) = i(l-1)+j$ and for any integer i, $[i] = \{1, \cdots, i\}$):

1. For each $u \in U$ and $s_i \in H_u$, relay node u combines the l sources $(\tilde{W}_{\beta(i,j)}, j \in [l])$ by applying an element-wise binary sum (denoted by operators $+$ and $\sum$). We denote each "combined" source by
$$\tilde{W}_i^{\text{sum}} = \sum_{j\in[l]} \tilde{W}_{\beta(i,j)}.$$

2. For the subsequent time step, each relay node in $\tilde{\mathcal{I}}$ operates the corresponding encoders from $\mathcal{C}$ using $(\tilde{W}_i^{\text{sum}}, i \in [k])$ as the source message in place of W_i. More precisely, if the encoders of $\mathcal{C}$ are
$$\bigcup_{u\in U}\bigcup_{\tau\in[n]}\Big\{f_{u,\tau}\Big\},$$
then the encoders of $\tilde{\mathcal{C}}$ for each $\tilde{u} \in \tilde{U} = U$ and each $\tau \in [n]$ are defined by
$$\tilde{X}_{\tilde{u},\tau} = f_{u,\tau}(\tilde{Y}_{\tilde{u}}^{\tau-1}, (\tilde{W}_i^{\text{sum}}, s_i \in H_u)).$$

3. At the end of n channel uses, each terminal $\tilde{t} = \tilde{t}_{\beta(i,j)} \in \tilde{T}$ first obtains a reconstruction $\widehat{\tilde{W}_i}^{\text{sum}}$ of $\tilde{W}_i^{\text{sum}}$ using the decoders of $t \in T$ from $\mathcal{C}$. Then terminal $\tilde{t}$ extracts the reconstruction $\widehat{\tilde{W}}_{\beta(i,j)}$ from $\widehat{\tilde{W}_i}^{\text{sum}}$ using side information $(\tilde{W}_{\beta(i,j')}, j' \in [l] \setminus j)$ and mixed sources $(\tilde{W}_i^{\text{sum}}, s_i \in H_t)$. More precisely, if the terminal decoders of $\mathcal{C}$ are
$$\Big\{g_t, t \in T\Big\},$$
then for each $\tilde{t} = \tilde{t}_{\beta(i,j)} \in \tilde{T}$, the decoder for terminal $\tilde{t}$ is given by
$$\widehat{\tilde{W}}_{\beta(i,j)} = d_t(Y_t^n, (\tilde{W}_i^{\text{sum}}, s_i \in H_t)) + \Bigg(\sum_{j'\in[l]\setminus\{j\}} \tilde{W}_{\beta(i,j')}\Bigg).$$

The resulting blocklength for $\tilde{\mathcal{C}}$ is $\tilde{n} = n$. Since $\mathcal{C}$ has error at most ϵ and source vector $(\tilde{W}_i^{\text{sum}}, i \in [k])$ is drawn from the same distribution and has the same support set as the source vector $(W_i, i \in [k])$ in $\mathcal{C}$, each terminal $\tilde{t} \in \tilde{T}$ can reconstruct $\widehat{\tilde{W}_i}^{\text{sum}}$ with error probability at most ϵ. Further, since the reconstruction of each $\widehat{\tilde{W}}_{\beta(i,j)}$ from $\widehat{\tilde{W}_i}^{\text{sum}}$ introduces no error, the error probability of $\tilde{\mathcal{C}}$ is thus bounded from above by the error probability $\tilde{\epsilon} = \epsilon$ from $\mathcal{C}$. Hence, $\tilde{\mathcal{I}}$ is $(\tilde{\mathbf{R}}(1-\rho), \epsilon, n)$-feasible. Since the rate of this code tends to $\tilde{\mathbf{R}}$ as δ tends to zero, we get the desired result.

$\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}) \Leftarrow \tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}})$: Fix any $\tilde{\epsilon}, \tilde{\rho} > 0$. We start with an $(\tilde{\mathbf{R}}(1-\tilde{\rho}), \tilde{\epsilon}, \tilde{n})$-feasible network code $\tilde{\mathcal{C}}$ for $\tilde{\mathcal{I}}$. Again, our goal is to transform $\tilde{\mathcal{C}}$ into an $(\mathbf{R}(1-\rho), \epsilon, n)$-feasible network code $\mathcal{C}$ for $\mathcal{I}$ by augmenting it with a linear channel outer code.

We begin by defining some notation. Denote by $\tilde{B}_{i,j}$ the side information source variables available to terminal $\tilde{t}_{\beta(i,j)}$ that share the same i subscript; thus

$$\tilde{B}_{i,j} = \left(\tilde{W}_{\beta(i,j')}, j' \in [l] \setminus j\right).$$

Denote by $\tilde{D}_i \in \mathbb{F}_2^{\tilde{n}lR_i}$ the l source message variables associated with s_i, giving

$$\tilde{D}_i = \left(\tilde{W}_{\beta(i,j')}, j' \in [l]\right).$$

Let $\tilde{A}_{i,j}$ denote the vector of output variables available at terminal $\tilde{t}_{\beta(i,j)}$ less the side information source variables $\tilde{B}_{i,j}$ at the end of $\tilde{n}$ transmissions, giving

$$\tilde{Z}_{\tilde{t}_{\beta(i,j)}} = (\tilde{B}_{i,j}, \tilde{A}_{i,j}).$$

Consider network $\mathcal{I}$, suppose that each source originating at s_i is the variable $W_i = \tilde{D}_i$. Consider applying the encoders from $\tilde{\mathcal{C}}$ to each relay node in $\mathcal{I}$. Since both $\mathcal{I}$ and $\tilde{\mathcal{I}}$ have the same channel transition function, the mutual information between each W_i and the variables received by terminal $t_{i,j}$ is then given by $I(\tilde{D}_i; \tilde{A}_{i,j})$. Note that

$$\begin{aligned} I(\tilde{D}_i; \tilde{A}_{i,j}) &= I(\tilde{B}_{i,j}; \tilde{A}_{i,j}) + I(\tilde{W}_{\beta(i,j)}; \tilde{A}_{i,j} | \tilde{B}_{i,j}) \\ &\geq I(\tilde{W}_{\beta(i,j)}; \tilde{A}_{i,j} | \tilde{B}_{i,j}) \\ &= I(\tilde{W}_{\beta(i,j)}; \tilde{A}_{i,j} | \tilde{B}_{i,j}) + I(\tilde{W}_{\beta(i,j)}; \tilde{B}_{i,j}) \qquad (4.1) \\ &= I(\tilde{W}_{\beta(i,j)}; \tilde{B}_{i,j}, \tilde{A}_{i,j}) \\ &\geq \tilde{n}R_i(1-\rho'), \qquad (4.2) \end{aligned}$$

where equation (4.1) follows from the fact that $\tilde{W}_{\beta(i,j)}$ and $\tilde{B}_{i,j}$ are independent, and equation (4.2) follows from Fano's inequality, with ρ going to zero as $\tilde{\epsilon}$ and $\tilde{\rho}$ go to zero.

By Lemma 4.5.1, we have that for any rate $R_i(1-\rho) < R_i(1-\rho')$ and $\epsilon > 0$, there exists blocklength n such that $\mathcal{I}$ is $(R_i(1-\rho), \epsilon, n)$-feasible. Hence, by closure of the capacity region, $\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})$. Furthermore, since linearity is preserved, $\tilde{\mathbf{R}} \in \mathcal{R}^L_\epsilon(\tilde{\mathcal{I}})$ implies that $\mathbf{R} \in \mathcal{R}^L_\epsilon(\mathcal{I})$.

Chapter 5

FROM CODE REDUCTION TO CAPACITY REDUCTION

The materials in this chapter are published in part as [40].

In this chapter, we describe some of the tools that enable us to derive capacity reduction results and to connect code reductions to their corresponding capacity reductions. In particular, we describe the tools that enable us to show that the code reduction from multiple unicast to 2-unicast network coding, and the code reduction from acyclic network coding to index coding, extend to corresponding capacity reductions if and only of the AERS holds. These tools also enable us to show that an entropic region outer bound for acyclic network coding is tight if and only if the AERS holds.

Since the above mentioned connections are derived for acyclic network coding networks, we first introduce the model for acyclic network coding in Section 5.1, which differs from the general network coding model in the description of a network code. We then introduce the concept of dependent sources in Section 5.2. Dependent sources are employed as a tool to study the change in capacity region when sources are correlated [25]. In [25], the authors show that the same rates achievable by sources with an asymptotically small correlation can be achieved by independent sources if and only if the AERS holds. Due to its close connection to the AERS, the concept of dependent sources serves as an important intermediate step in connecting reduction to the AERS.

In Section 5.3, we present techniques and useful lemmas that are used to prove our reduction results. In Section 5.4, we derive two equivalent formulation of the AERS by identifying a network that is representative of the edge removal statement. This enables us to study a particular network topology when exploring the AERS without losing generality. By focusing on these critical network topologies, we derive a sufficient condition for a capacity reduction to be equivalent to the AERS. This result is presented in Section 5.5.

5.1 Acyclic Network Coding

An acyclic network coding instance $\mathcal{I} = (G, S, T)$ is a network coding instance in which we restrict the underlying graph G to be acyclic.

The model for acyclic network coding differs from that of general network coding in the definition of network codes. Instead of the "simultaneous" schedule of encoders given in Section 2.2 where encoders are operated simultaneously at each time step, network codes for acyclic network coding networks are often given as block network codes where we assign a *single* block encoding function to each edge e. Under the schedule of block codes, the encoders operate in an order that ensures each node v operates only after all "upstream" nodes (nodes that have a directed path to v in G) have completed their operations, taking up to $n|E|$ time steps in total. As a result of this schedule, a valid sequence of transmissions exists if the function for each edge e can be computed locally by each node $\text{In}(e)$. This provides us with a clean single-function characterization of valid codes.

Since G is acyclic by assumption, any code under the "simultaneous" schedule can be converted to an equivalent block network code. While not all block codes can be converted to precisely equivalent codes (i.e., codes with the same blocklength) under a simultaneous schedule, they can be converted into related codes with asymptotically equivalent performance [41]. As a result, the capacity region does not depend on the schedule of the code [41]. For ease of notation and analysis, we use block network codes when dealing with acyclic network coding instances.

In the following, we give the notation of block codes for acyclic network coding as well as for index coding instances.

Block Network Code

Given a rate vector $\mathbf{R} = (R_1, \ldots, R_k)$ and a blocklength n, a $(2^{n\mathbf{R}}, n)$ block network code $\mathcal{C}$ is a mechanism for simultaneous transmission of a rate R_i message from each source $s_i \in S$ to its corresponding terminal $t_i \in T$ over n uses of the network G.

Similar to communication codes, for each $s \in S$, we use

$$W_s \in \mathcal{W}_s = \mathbb{F}_2^{nR_s}$$

to represent the source message originating at node s. Each element of $\mathbb{F}_2^m$ is represented by a length-m row vector over $\mathbb{F}_2$. Source messages are carried through the network using edge messages. The blocklength n message carried by edge $e \in E_{S^c}$ is

$$X_e \in \mathcal{X}_e = \mathbb{F}_2^{nc_e}.$$

Each source edge $e = (s, v) \in E_S$ is of infinite capacity. Again, infinite capacity edges are used to capture the notion that source W_s is available apriori to node v; thus,

$$X_e = W_s.$$

For any non-source node $v \in \overline{S}$, the information available to node v after all such transmissions is

$$Z_v = (Y_v, W_{H_v}) \in \mathcal{Y}_v \times \mathcal{W}_{H_v} = \mathcal{Z}_v.$$

Here, Y_v is the vector of messages delivered to node v on its incoming edges and W_{H_v} is the vector of sources available to node v. Thus,

$$\begin{aligned} Y_v &= \Big(X_{e'}, e' \in E_{S^c} \wedge (\mathrm{Out}(e') = v)\Big) \\ \mathcal{Y}_v &= \prod_{e' \in E_{S^c} \wedge (\mathrm{Out}(e')=v)} \mathcal{X}_{e'} \\ W_{H_v} &= (W_s, s \in H_v) \\ \mathcal{W}_{H_v} &= \prod_{s \in H_v} \mathcal{W}_s \end{aligned}$$

Each terminal $t \in T$ uses its available information Z_t to reproduce its desired source. We use

$$\widehat{W_t} \in \mathcal{W}_t = \mathbb{F}_2^{nR_t}$$

to represent the reproduction.

A $(2^{n\mathbf{R}}, n)$ network code $\mathcal{C}$ comprises an encoding function f_e for each edge $e \in E_{S^c}$, and a decoding function g_t for each terminal $t \in T$, giving $\mathcal{C} = (\{f_e\}, \{g_t\})$. For each $e \in E_{S^c}$, encoder f_e is a mapping

$$f_e : \mathcal{Z}_{\mathrm{In}(e)} \to \mathcal{X}_e$$

from the vector $Z_{\mathrm{In}(e)} \in \mathcal{Y}_{\mathrm{In}(e)} \times \mathcal{W}_{H_{\mathrm{In}(e)}}$ of information available to node $\mathrm{In}(e)$ to the value $X_e \in \mathcal{X}_e$ carried by edge e over its n channel uses; thus

$$X_e = f_e(Z_{\mathrm{In}(e)}).$$

By our assumption of the schedule of block network codes, the encoders operate in an order that ensures that the encoders for all edges incoming to node v operate before the encoders for all edges outgoing from the same node; this is possible since the network is acyclic by assumption.

We call f_e a *local encoding function* since it describes the local operation used to map the inputs of node $\mathrm{In}(e)$ to the output X_e transmitted across edge e. Since the network is deterministic, X_e can also be expressed as a deterministic function of the network inputs $\mathbf{W} = (W_s : s \in S)$. The resulting *global encoding function*

$$F_e : \prod_{s \in S} \mathcal{W}_s \to \mathcal{X}_e$$

takes as its input the source vector $\mathbf{W}$ and maps it directly to X_e. For each source edge $e \in E_S$, the global encoding function is given by

$$F_e(\mathbf{W}) = W_s.$$

Following the partial ordering on E, the global encoding function for each subsequent $e \in E$ is then a function of the global encoding functions for its inputs, giving

$$F_e(\mathbf{W}) = \begin{cases} W_{\mathrm{In}(e)} & \text{if } e \in E_S \\ f_e\Big(F_{e'}(\mathbf{W}) : e' \in E \wedge (\mathrm{Out}(e') = \mathrm{In}(e))\Big) & \text{if } e \in E_{S^c}. \end{cases}$$

Each decoder g_t, $t \in T$, is a mapping

$$g_t : \mathcal{Z}_t \to \mathcal{W}_t$$

from the vector $Z_t \in \mathcal{Y}_t \times \mathcal{W}_{H_t}$ of information available to node t to the reproduction $\widehat{W}_t$ of its desired source, giving

$$\widehat{W}_t = g_t(Z_t).$$

Block Index Code

An index code $\mathcal{C}$ is a block network code for the index coding network. Recall that an index coding problems is described by $\mathcal{I} = (S, T, H, c_B)$ in which a bottleneck link $B = (u1, u_2)$ that has access to all the sources broadcasts a common rate-c_B message to all the terminals. We assume without loss of generality that any edge with sufficient capacity to carry all information available to its input node carries that information unchanged; thus

$$f_e(Z_{\mathrm{In}(e)}) = Z_{\mathrm{In}(e)} \text{ for all } e \in E \wedge \mathrm{In}(e) = u_2.$$

As a result, specifying an index code's encoder requires specifying only the encoder f_B for its bottleneck link.

5.2 Dependent Sources

In the classical network coding setting, independent sources are considered. We employ dependent sources [25] to understand the rates achievable when dependent sources are allowed.

For a blocklength n, a vector of $n\delta$-dependent sources of rate $\mathbf{R} = (R_1, ..., R_{|S|})$ is a random vector $\mathbf{W} = (W_1, \cdots, W_k)$, where $W_i \in \mathbb{F}_2^{nR_i+n\delta}$ such that

$$\sum_{i\in[k]} H(W_i) - H(\mathbf{W}) \leq n\delta$$

and $H(W_i) \geq nR_i$ for all $i \in [k]$. For $\delta = 0$, the set of $n\delta$-dependent sources only includes random variables that are independent and uniformly distributed over their supports.

We also consider linearly dependent sources [42, Section 2.2]. A vector of $n\delta$-linearly-dependent sources are $n\delta$-dependent sources that are *pre-specified* linear combinations of underlying independent processes [42]. We denote such an underlying independent process by U, and we assume that U is uniformly distributed over $\mathbb{F}_2^{nR_U}$. Thus, each W_i can be expressed as

$$W_i = UT_i,$$

where each T_i is an $nR_U \times n(R_i + \delta)$ matrix over $\mathbb{F}_2$.

We now define communication with $n\delta$ dependent sources. Network $\mathcal{I}$ is $(\mathbf{R}, \epsilon, n, \delta)$-feasible if there exists a set of $n\delta$-dependent source variables $\mathbf{W}$ and a network code $\mathcal{C} = \{\{f_e\}, \{g_t\}\}$ with blocklength n such that operation of code $\mathcal{C}$ on source message random variables $\mathbf{W}$ yields error probability $P_e^{(n)} \leq \epsilon$. When the feasibility vector does not include the parameter for dependence (i.e., $(\mathbf{R}, \epsilon, n)$), we assume that the sources are independent.

5.3 Code Reduction Results

In Lemma 5.3.1, below, we present code reduction results that are useful in our proofs. In what follows, we map an epsilon-error code for dependent sources to a zero-error code for independent sources by introducing a cooperation facilitator to the network (Lemma 5.3.1(1),(3)). Similarly, we map an epsilon-error code for a network augmented with a broadcast facilitator to a zero-error code that can operate without the broadcast facilitator by using dependent sources (Lemma 5.3.1(2)).

Lemma 5.3.1. *For any network $\mathcal{I}$ and rate vector $\mathbf{R}$,*

1. *[43, Corollary 5.1] For any blocklength n and any $\epsilon, \delta \geq 0$, if $\mathcal{I}$ is $(\mathbf{R}, \epsilon, n, \delta)$-feasible, then by adding a cooperation facilitator, the network $\mathcal{I}_{\lambda}^{cf}$ is $(\mathbf{R} - \rho, 0, n, 0)$-feasible, where λ and ρ tend to zero as ϵ and δ tend to zero and n goes to infinity.*

2. *For any blocklength n and any $\epsilon, \lambda, \delta \geq 0$, if $\mathcal{I}_{\lambda}^{bf}$ is $(\mathbf{R}, \epsilon, n, \delta)$-feasible, then for dependent sources, $\mathcal{I}$ is $(\mathbf{R} - \rho, 0, n, \delta')$-feasible, where δ' and ρ tend to zero as $\epsilon, \lambda, \delta$ tend to zero.*

3. *For any blocklength n and any $\epsilon, \delta \geq 0$, if $\mathcal{I}$ is $(\mathbf{R}, \epsilon, n, \delta)$-linearly-feasible for an $n\delta$-linearly-dependent source, then by adding a cooperation facilitator, the network $\mathcal{I}_{\lambda}^{cf}$ is $(\mathbf{R} - \rho, \epsilon, n, 0)$-feasible, where λ and ρ tend to zero as δ tend to zero.*

Proof. (2) Let $\mathcal{I}_{\lambda}^{bf}$ be $(\mathbf{R}, \epsilon, n, \delta)$-feasible. By Lemma 5.3.1(1), $\mathcal{I}_{\lambda+\lambda^*}^{bf}$ is $(\mathbf{R} - \rho^*, 0, n, 0)$-feasible, where λ^* and ρ^* tend to zero as ϵ and δ tend to zero and n tends to infinity. (Adding a λ^* cooperation facilitator to $\mathcal{I}_{\lambda}^{bf}$ is equivalent to increasing the bottleneck capacity of $\mathcal{I}_{\lambda}^{bf}$ from λ to $\lambda + \lambda^*$.) Let $\lambda' = \lambda + \lambda^*$, and let $\mathcal{C}$ be an $(\mathbf{R}, 0, n, 0)$-feasible network code for $\mathcal{I}_{\lambda'}^{bf}$.

Define Z_b to be the value being sent on the bottleneck edge (s_{su}, s_{re}) of the broadcast facilitator in $\mathcal{I}_{\lambda'}^{bf}$ under the operation of $\mathcal{C}$. The conditional entropy $H(\mathbf{W}|Z_b)$ can be expanded as

$$H(\mathbf{W}|Z_b) = \sum_{\gamma' \in \mathbb{F}_2^{n\lambda'}} p(Z_b = \gamma') H(\mathbf{W}|Z_b = \gamma').$$

By an averaging argument, there exists a γ such that $H(\mathbf{W}|Z_b = \gamma) \geq H(\mathbf{W}|Z_b)$. We therefore have the following inequalities:

$$\begin{aligned} H(\mathbf{W}|Z_b = \gamma) &\geq H(\mathbf{W}|Z_b) \\ &\geq H(\mathbf{W}) - H(Z_b) \\ &\geq H(\mathbf{W}) - n\lambda'. \end{aligned} \tag{5.1}$$

For each $i \in [k]$,

$$\begin{aligned} H(W_i|Z_b = \gamma) &\geq H(\mathbf{W}|Z_b = \gamma) - H((W_j, j \in [k] \setminus \{i\})|Z_b = \gamma) \\ &\geq H(\mathbf{W}) - n\lambda - \sum_{j \in [k] \setminus \{i\}} \left(nR_j \right) \qquad (5.2) \\ &= H(W_i) - n\lambda'. \end{aligned}$$

Inequality (5.2) follows from bound on the support size of W_j and equation (5.1).

$$\begin{aligned} \sum_{i \in [k]} H(W_i|Z_b = \gamma) - H(\mathbf{W}|Z_b = \gamma) &\leq \sum_{i \in [k]} (nR_i) - H(\mathbf{W}) + n\lambda' \qquad (5.3) \\ &\leq n\lambda' \end{aligned}$$

Inequality (5.3) follows from bound on support size of $\mathbf{W}$.
Define the conditional random variable $\mathbf{W}^* = \mathbf{W}|_{Z_b=\gamma}$. The alphabet size of $\mathbf{W}^*$ is the same as $\mathbf{W}$, where for each $i \in [k]$, $|\mathcal{W}_i| = 2^{nR_i}$. We then have an $n\lambda'$-dependent source $\mathbf{W}^*$ such that $\mathcal{I}$ is $(\mathbf{R} - \lambda', 0, n, \lambda')$-feasible.

(3) Let $\mathbf{W} = (W_1, \cdots, W_k)$ denote the $n\delta$-linearly dependent source (See Section 5.2) with U as the underlying random process. Therefore, the source is the result of a linear transformation of U, namely,

$$\mathcal{L}_U : \mathbb{F}_2^{nR_U} \to \prod_{i \in [k]} \mathbb{F}_2^{nR_i}.$$

In particular, we can express each source W_i as

$$W_i = UL_i,$$

where each L_i is an nR_U by nR_i matrix over $\mathbb{F}_2$.

For any matrix M, denote by $M(i)$ the ith column of M and by $CS(M)$ the column space of M. For a row vector W_i, denote by $W_{i,j}$ the jth element of W_i. For vector spaces V and W, let $V + W$ denote the direct sum of V and W, (i.e., $\{v + w | v \in V, w \in W\}$). Let $B_i \subseteq [nR_i]$ be index sets such that for $B = \{(i, j) : i \in [k], j \in B_i\}$,

$$\bigcup_{(i,j) \in B} \{L_i(j)\}$$

is the minimum spanning set of the vector space $\sum_{i \in [k]} CS(F_i)$. Thus the rank of $\mathcal{L}_U$ equals $|B|$.

Let $\bar{B}_i = [nR_i] \setminus B_i$, then for each (i, j) such that $i \in [k], j \in \bar{B}_i$, we can express $UL_i(j)$ as linear combinations of the spanning set, i.e.,

$$UL_i(j) = \sum_{i' \in [k]} \sum_{j' \in B_{i'}} \eta_{iji'j'} UL_{i'}(j'),$$

where each $\eta_{iji'j'}$ is a coefficient in $\mathbb{F}_2$.

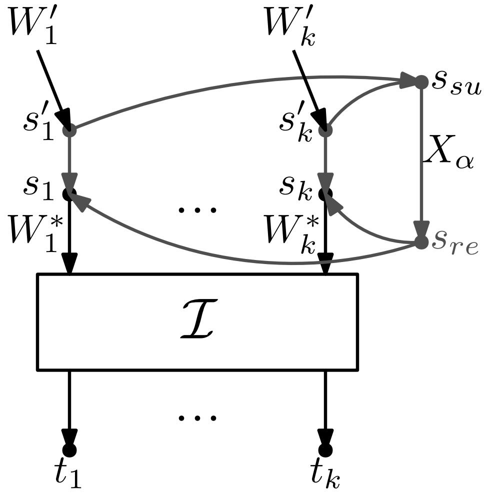


Figure 5.1: A schematic for generating dependent variables using a cooperation facilitator.

Next, we describe a scheme to turn the dependent source code for $\mathcal{I}$ into a code for $\mathcal{I}_\delta^{cf}$ that would operate on independent sources at the cost of a small loss in rate. Let $\mathbf{W}' = (W_1', \cdots, W_k')$ be a set of independent sources for $\mathcal{I}_\delta^{cf}$ where each W_i' is uniformly distributes in $\mathbb{F}_2^{|B_i|}$. With the help of the cooperation facilitator, we first transform $\mathbf{W}'$ into a set of dependent variables $\mathbf{W}^* = (W_1^*, \cdots, W_k^*)$ using a full rank linear transformation

$$\mathcal{L}^* : \prod_{i \in [k]} \mathbb{F}_2^{n|B_i|} \to \prod_{i \in [k]} \mathbb{F}_2^{nR_i}$$

before applying the encoders of $\mathcal{I}$ (See Figure 5.1). Due to the topology of $\mathcal{I}_\delta^{cf}$, we need a distributed scheme to apply $\mathcal{L}^*$. We therefore need a two-step procedure to generate $\mathbf{W}^*$. At the first step, the cooperation facilitator first broadcast a common message $X_\alpha = f_\alpha(\mathbf{W})$,

$$f_\alpha : \prod_{i \in [k]} \mathbb{F}_2^{n|B_i|} \to \mathbb{F}_2^{n\delta},$$

to each of the old source nodes $s_1, \cdots, s_k$ in $\mathcal{I}_\delta^{cf}$. In the second step, each s_i apply a local linear transformation

$$\mathcal{L}_i^* : \mathbb{F}_2^{n|B_i|} \times \mathbb{F}_2^{n\delta} \to \mathbb{F}_2^{nR_i},$$

that maps (W_i', X_α) to W_i^*.

To ensure correct operation of the code, $\mathcal{L}^*$ is designed such that the distribution and the support set of $\mathbf{W}^*$ equals that of $\mathbf{W}$. By assumption of the validity of the code, each terminal t_i will be able to reconstruct W_i^* with overall error probability less than ϵ. Finally, to allow for each terminal to reconstruct W_i' from W_i^*, we also require W_i' to be a function of W_i^*. Thus, an appropriate choice of $\mathcal{L}^*$ and λ will yield an $(\mathbf{R} - \delta, \epsilon, n, 0)$-linearly-feasible code for $\mathcal{I}_\delta^{cf}$.

In what follows, we give the definitions of f_α and $\{\mathcal{L}_i\}$ that satisfy the requirements.

- Let $\bar{B} = \{(i,j) : i \in [k], j \in \bar{B}_i\}$. Recall that $W_{i,j}$ denotes the jth bit of W_i. The function f_α is defined such that $X_\alpha = f_\alpha(\mathbf{W}') = (\alpha_{i,j}, (i,j) \in \bar{B})$, where
$$\alpha_{i,j} = \sum_{i' \in [k]} \sum_{j' \in B_i'} \eta_{iji'j'} W'_{i',j'}.$$
The rate of the CF required to broadcast this function can be bounded as follow,
$$|\bar{B}| = \sum_{i \in [k]} |B_i| - H(\mathbf{W}) \le \left(\sum_{i \in [k]} nR_i \right) - \left(\sum_{i \in [k]} nR_i \right) + n\delta = n\delta.$$

- For each $i \in [k]$, fix an arbitrary ordering of elements in B_i. define an index mapping function
$$\beta_i : B_i \to [|B_i|]$$
that maps j to i if j is the ith element in B_i. The linear transformation $\mathcal{L}_i^*$ that maps W_i' to W_i^* is defined as follows:
$$W_{i,j}^* = \begin{cases} W'_{i,\beta_i(j)} & j \in B_i \\ \alpha_{i,j} & j \in \bar{B}_i. \end{cases}$$
It can be verified that there exists a inverse map from W_i^* to W_i' for each $i \in [k]$, and the resulting $\mathcal{L}^*$ is full rank.

Finally, we show that $\mathbf{W}^*$ and $\mathbf{W}$ has the same distribution and support set. Observe that $\mathbf{W}$ is uniformly distributed over its support since it is a linear function of U which is uniformly distributed over $\mathbb{F}_2^{nR_U}$. This is due to the fact that the pre-image of any element in the co-domain of $\mathcal{L}_U$ has the same size. Similarly, since $\mathbf{W}'$ is uniformly distributed, $\mathbf{W}^*$ is also uniformly distributed over its support. Let $\mathbf{w} = (w_1, \cdots, w_k) \in \mathcal{W}_{\text{sp}}$, then $\mathbf{w}' = (w'_1, \cdots, w'_k)$ defined as follows satisfies $\mathcal{L}_U^*(\mathbf{w}') = \mathbf{w}$,

$$w'_{i,j} = \begin{cases} w_{i,j} & j \in B_i \\ \sum_{i' \in [k]} \sum_{j' \in B'_i} \eta_{iji'j'} w_{i,j} & j \in \bar{B}_i. \end{cases}$$

Thus, the support set of $\mathbf{W}$ is contained in that of $\mathbf{W}^*$. Finally, since both $\mathcal{L}_U$ and $\mathcal{L}^*$ has the same rank, the support set of $\mathbf{W}$ equals $\mathbf{W}^*$. $\mathcal{I}_\delta^{cf}$ is therefore $(\mathbf{R} - \delta, \epsilon, n, 0)$-linearly-feasible. □

5.4 Representative Topologies for The Edge Removal Statement

In Section 3.2, we introduce the cooperation facilitator (CF) and broadcast facilitator (BF) from [30] which can turn any network into a super-source network. Since the AERS is known to be true for networks super-source networks, adding any negligible edge to a super-source network will not impact its capacity region. This implies that the key to understanding the AERS is to understand the effect of adding a CF or BF to a network. Indeed, there is no loss in generality in restricting ourselves to special network topologies when studying the AERS. This observation is described in Theorem 5.4.1, which gives two equivalent formulations of the AERS.

Theorem 5.4.1. *The following statements are equivalent for any acyclic network $\mathcal{I}$.*

$$\text{(a) } \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda) \leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}).$$

$$\text{(b) } \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf}) \leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}).$$

$$\text{(c) } \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{cf}) \leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}).$$

Proof. This proof relies on Lemma 5.3.1.

(b) → (a): Let $\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda)$. Since the broadcast facilitator in $\mathcal{I}_\lambda^{bf}$ can compute edge e in $\mathcal{I}_\lambda$ and broadcast it to all nodes, we have $\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$.

By assumption of (a), this implies $\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})$. Since $\mathcal{R}_\epsilon(\mathcal{I}) \subseteq \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda)$, (b) is true.

(a) → (c): Let $\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{cf})$. Let $\mathcal{I}_c$ be obtained from $\mathcal{I}_\lambda^{cf}$ by removing the rate-λ bottleneck link in $\mathcal{I}_\lambda^{cf}$, then $\mathcal{R}_\epsilon(\mathcal{I}) = \mathcal{R}_\epsilon(\mathcal{I}_c)$. By assumption of (a), this implies $\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}_c)$, and therefore, $\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})$. Since $\mathcal{R}_\epsilon(\mathcal{I}) \subseteq \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{cf})$, (c) is true.

(c) → (b): Let $\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$, then for any $\epsilon, \lambda, \rho > 0$, $\mathcal{I}_\lambda^{bf}$ is $(\mathbf{R} - \rho, \epsilon, n, 0)$-feasible for some blocklength n. By Lemma 5.3.1(2), for all $\rho', \delta' > 0$, there exists blocklength n' such that $\mathcal{I}$ is $(\mathbf{R} - \rho', 0, n', \delta')$-feasible. By Lemma 5.3.1(1), for all $\rho'', \lambda'' > 0$, there exists blocklength n'' such that $\mathcal{I}_{\lambda''}^{cf}$ is $(\mathbf{R} - \rho'', 0, n'', 0)$-feasible. Thus, $\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{cf})$. By assumption of (c), we have $\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})$. Since $\mathcal{R}_\epsilon(\mathcal{I}) \subseteq \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$, (b) is true. □

Roughly speaking, equivalent form (b) in Theorem 5.4.1 describes the observation that the bottleneck edge in a broadcast facilitator is the strongest edge since it can compute any function of the sources and deliver it to any node, it is therefore representative of any other edge in a network. Equivalent form (c) in Theorem 5.4.1 describes the observation that even though the cooperation facilitator is not as strong as a broadcast facilitator, their difference (with respect to capacity region) becomes negligible when the bottleneck capacity becomes asymptotically small. Thus, studying the effect of the removal of the bottleneck edge of a cooperation facilitator (or broadcast facilitator) is representative of the AERS.

5.5 A sufficient Condition for Capacity Reduction

In this section, we derive sufficient conditions for a capacity reduction to be equivalent to the AERS or Linear AERS. The sufficient condition is presented in Theorem 5.5.1, below. One key observation of the capacity reductions mentioned in this paper is that they are only known to be true for super-source networks. Roughly speaking, Theorem 5.5.1 shows that if a broadcast facilitator allows one to prove a capacity reduction from $\mathcal{I}_\lambda^{bf}$ to $\tilde{\mathcal{I}}$, then the capacity reduction from $\mathcal{I}$ to $\tilde{\mathcal{I}}$ is equivalent to the AERS. Similarly, if one can prove a linear capacity reduction from $\mathcal{I}_\lambda^{bf}$ to $\tilde{\mathcal{I}}$, since edge removal is true for linear capacity regions (Theorem 3.2.1), a linear capacity reduction from $\mathcal{I}$ to $\tilde{\mathcal{I}}$ holds. This result is captured in Theorem 5.5.1.

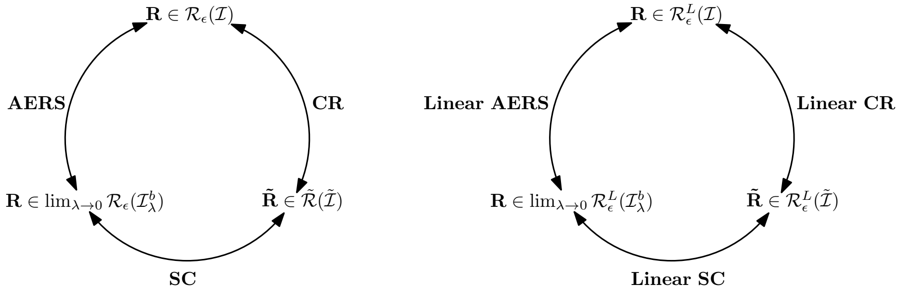

$\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})$
AERS
CR
$\mathbf{R} \in \lim_{\lambda\to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^b)$
$\tilde{\mathbf{R}} \in \tilde{\mathcal{R}}(\tilde{\mathcal{I}})$
SC
$\mathbf{R} \in \mathcal{R}_\epsilon^L(\mathcal{I})$
Linear AERS
Linear CR
$\mathbf{R} \in \lim_{\lambda\to 0} \mathcal{R}_\epsilon^L(\mathcal{I}_\lambda^b)$
$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon^L(\tilde{\mathcal{I}})$
Linear SC

Figure 5.2: The figure depicts the proof of Theorem 5.5.1 which proves a sufficient conditions for the equivalence between capacity reduction and AERS for both the general and linear case.

Theorem 5.5.1. *For a acyclic network coding instance $\mathcal{I}$, rate vector $\mathbf{R}$ and a reduction mapping Φ, let $\tilde{\mathcal{I}}$ and $\tilde{\mathbf{R}}$ be the corresponding network coding instance and rate vector; then*

1. $\left[\tilde{\mathbf{R}} \in \mathcal{R}(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \lim\limits_{\lambda\to 0}\mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})\right]$ *implies*

$$\left[\left(\tilde{\mathbf{R}} \in \mathcal{R}(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})\right) \Leftrightarrow \ \ \textbf{\textit{AERS}}\right].$$

2. $\left[\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon^L(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \lim\limits_{\lambda\to 0}\mathcal{R}_\epsilon^L(\mathcal{I}_\lambda^{bf})\right]$ *implies* $\left[\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon^L(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon^L(\mathcal{I})\right]$.

Proof. The proof idea is illustrated in Figure 5.2. Suppose that the sufficient condition (SC) is true; we show that capacity reduction (CR) is equivalent to the AERS. There are two directions to be proven. For the first direction, assume that the CR is true. By CR, $\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})$ if and only if $\mathbf{R} \in \tilde{\mathcal{R}}(\tilde{\mathcal{I}})$. By the SC, we have $\mathbf{R} \in \tilde{\mathcal{R}}(\tilde{\mathcal{I}})$ if and only if $\mathbf{R} \in \lim\limits_{\lambda\to 0}\mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$. This implies that AERS is true. For the second direction, assume that AERS is true. By AERS, $\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})$ if and only if $\mathbf{R} \in \lim\limits_{\lambda\to 0}\mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$. By the SC, we have $\mathbf{R} \in \lim\limits_{\lambda\to 0}\mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$ if and only if $\mathbf{R} \in \tilde{\mathcal{R}}(\tilde{\mathcal{I}})$. This implies that CR is true. The proof for the linear case is similar. □

Chapter 6

CAPACITY REDUCTION FROM MULTIPLE UNICAST TO 2-UNICAST

The materials of this chapter are published in part as [40].

Demand type plays a central role in characterizing the capacity region of network coding networks. For single-source network coding problems, the maximum rate at which information can be multicast has a simple characterization via the maximum flow of the underlying graph of the network coding problem. However, the k-source network coding problem is a non-trivial extension of the single-source case.

In the literature, results for k-source network coding problems are derived for various values of "k". For example, the authors of [44] show that for k equals 5, linear codes are insufficient to achieve the network coding capacity. The authors of [45] show that the linear programming outer bound [46] is loose when k equals 6. We have already seen from Chapter 4 that multiple multicast networks reduce to multiple unicast networks. One may ask the question how the difficulty of a k-unicast network coding problem depends on k. It is tempting to believe that the k-unicast problems are inherently easier for small values of k. For example, the authors of [47] show that the cut-set bound is tight for undirected k-unicast network coding problems for $k = 2$.

A surprising result in [14] proves a code reduction from k-unicast network coding to 2-unicast network coding. The authors of [14] pose the question of whether the reduction mapping in [14] (here we refer to as mapping Φ_2) can be used to prove a capacity reduction. We resolve this question partially by showing that under mapping Φ_2, the linear capacity calculation reduces from k-unicast to 2-unicast. Furthermore, the general capacity reduction from k-unicast to 2-unicast holds if and only if the AERS holds.

In the next section, we first describe mapping Φ_2. Our capacity reduction result is formally captured in Theorem 6.2.1 of Section 6.2, and its proof is given in Section 6.5. A useful lemma that maps a lossy code for dependent sources to a lossless network code for independent sources is given in Section 6.4.

6.1 Reduction Mapping Φ_2

We begin by describing the mapping Φ_2 used to prove the code reduction from k-unicast network coding to 2-unicast network coding in [14], modified slightly to fit our model.

Let $\mathcal{I} = ((V, E, C), S, T)$ and $\mathbf{R} = (R_1, \cdots, R_k)$. The corresponding network $\tilde{\mathcal{I}} = ((\tilde{V}, \tilde{E}, \tilde{C}), \tilde{S}, \tilde{T})$ and rate $\tilde{\mathbf{R}}$ is given below. The construction in [14] first combines demands $\{(s_1, t_1), \cdots, (s_k, t_k)\}$ of $\mathcal{I}$ into a single demand $(\tilde{s}_1, \tilde{t}_1)$ in $\tilde{\mathcal{I}}$. The instance $\tilde{\mathcal{I}}$ employs a modified butterfly structure for each demand (s_i, t_i), $i \in [k]$. The construction is illustrated in Figure 6.1.

These butterflies are connected to network $\mathcal{I}$ so that terminal node t_i in $\mathcal{I}$ acts as the right-side "source" of the i^{th} butterfly in $\tilde{\mathcal{I}}$. The left-side source node, a_i, of the same butterfly is connected to $\tilde{s}_2$, which carries an independent source $\tilde{W}_2$. We therefore have

$$\tilde{S} = \{\tilde{s}_1, \tilde{s}_2\}.$$

Finally, the left-side and right-side "terminal nodes" of the i^{th} butterfly are connected to terminal nodes $\tilde{t}_1$ and $\tilde{t}_2$ of $\tilde{\mathcal{I}}$, respectively, giving

$$\tilde{T} = \{\tilde{t}_1, \tilde{t}_2\}.$$

Since we are combining k sources, the new rate vector for $\tilde{\mathcal{I}}$ is

$$\tilde{\mathbf{R}} = \left(\sum_{i \in [k]} R_i, \sum_{i \in [k]} R_i \right).$$

The resulting graph is given by

$$\begin{aligned}
\tilde{V} &= V \cup \tilde{S} \cup \tilde{T} \cup \Big\{ u_1, u_2 \Big\} \cup \Big[\bigcup_{i \in [k]} \{a_i, b_i, c_i, f_i, h_i\} \Big] \\
\tilde{E}_i &= \{(u_1, s_i), (u_2, a_i), (f_i, \tilde{t}_1)(h_i, \tilde{t}_2), (t_i, b_i), (a_i, b_i), \\
&\quad (a_i f_i), (b_i, c_i), (c_i, h_i), (c_i, f_i), (s_i, h_i)\} \\
\tilde{E} &= E \cup \Big\{ (\tilde{s}_1, u_1), (\tilde{s}_2, u_2) \Big\} \cup \Big[\bigcup_{i \in [k]} \tilde{E}_i \Big] \\
\tilde{c}_e &= \begin{cases} \infty & \text{if } \mathrm{In}(e) \in \{\tilde{s}_1, \tilde{s}_2\} \\ R_i & \text{if } e \in \tilde{E}_i \vee \mathrm{In}(e) \in S \\ c_e & \text{if } e \in E. \end{cases}
\end{aligned}$$

Note that $\tilde{\mathcal{I}}$ depends on both $\mathcal{I}$ in its topology and $\mathbf{R}$ in its edge capacities. Here, for each $i \in [k]$, edges (u_1, s_i), (u_2, a_i) and each edge in $\tilde{E}_i$ is of capacity R_i. Infinite capacity edges $(\tilde{s}_1, u_1)$ and $(\tilde{s}_2, u_2)$ are added to capture the notion that the sources are available a priori to u_1 and u_2. Sources from $\tilde{s}_1$ and $\tilde{s}_2$ are demanded by nodes $\tilde{t}_1$ and $\tilde{t}_2$, respectively.

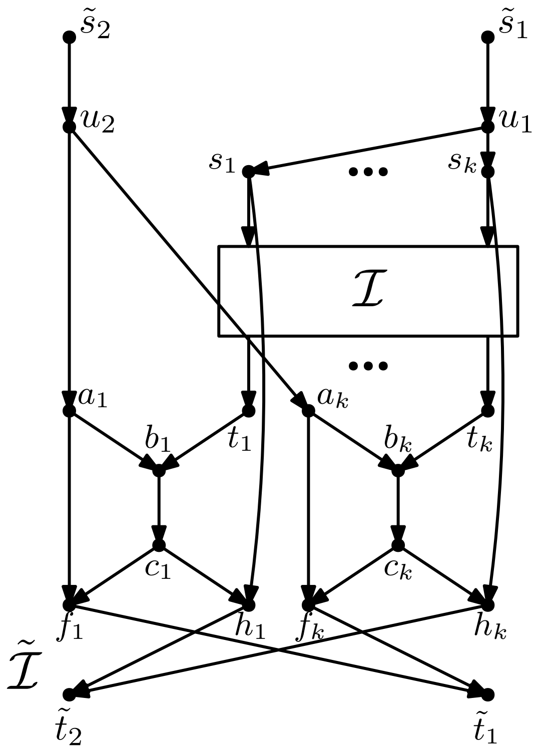


Figure 6.1: Graphical representation of corresponding network $\tilde{\mathcal{I}}$. Network $\tilde{\mathcal{I}}$ contains $\mathcal{I}$ as a sub-network and is augmented with k "butterfly" network structures.

6.2 Main Result

Theorem 6.2.1 gives a partial solution to the question of whether Φ_2 can be used to derive a reduction in capacity region, which is left open by authors of [14].

Theorem 6.2.1 (Capacity Reduction from k-Unicast to 2-Unicast)**.**

1. *Linear capacity characterization for k-Unicast network coding reduces to linear capacity characterization for 2-Unicast network coding. That is, under mapping Φ_2, for any acyclic network coding instance $\mathcal{I}$ and rate*

vector $\mathbf{R}$*,*

$$\tilde{\mathbf{R}} \in \mathcal{R}^L_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}^L_\epsilon(\mathcal{I}).$$

2. *Capacity characterization for k-Unicast network coding reduces to capacity characterization for 2-Unicast network coding under mapping* Φ_2 *if and only if the* ***AERS*** *holds. That is, for any acyclic network coding instance* $\mathcal{I}$ *and rate vector* $\mathbf{R}$*,*

$$\left(\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})\right) \textit{ if and only if the } \textbf{\textit{AERS}} \textit{ holds.}$$

Proof. See Section 6.5. □

6.3 Insufficiency of Linear Coding in 2-Unicast Network Coding

One application of capacity reduction results is to generate new results from existing ones that are proven for a different class of networks. In [44], the authors construct a 5-source multiple multicast network (call it $\mathcal{I}$) to demonstrate the insufficiency of linear coding in network coding networks. One application of capacity reduction is to reduce this example network to a 2-unicast network $\tilde{\mathcal{I}}$. The result in [44] proved that there exists $\mathbf{R}^*$ such that $\mathbf{R}^* \in \mathcal{R}_\epsilon(\mathcal{I})$ but $\mathbf{R}^* \notin \mathcal{R}^L_\epsilon(\mathcal{I})$.

In the proof of Theorem 6.2.1 (2), we show that

$$\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}^{bf}_\lambda) \leftrightarrow \tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}),$$

which gives

$$\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I}) \rightarrow \tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}).$$

By Theorem 6.2.1 (1), we have

$$\mathbf{R} \in \mathcal{R}^L_\epsilon(\mathcal{I}) \leftarrow \tilde{\mathbf{R}} \in \mathcal{R}^L_\epsilon(\tilde{\mathcal{I}}).$$

Thus, the resulting network $\tilde{\mathcal{I}}$ preserves the gap between linear and optimal codes. That is, $\tilde{\mathbf{R}}^* \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}})$, but $\tilde{\mathbf{R}}^* \notin \mathcal{R}^L_\epsilon(\tilde{\mathcal{I}})$. This yields a 2-unicast network coding network demonstrating the insufficiency of linear coding.

6.4 A Linear Code Reduction from Lossy Source Coding to Lossless Network Coding

We consider a lossy source coding scenario that is similar to that described in Section 4.5, except that the sources in this case are not independent. That is, we consider the scenario where operation of the encoders $\{f_{u,\tau}, u \in U\}$ on a set of dependent sources $\mathbf{W}$ yields a mutual information $I(W_t; Z_t) \geq nR_t$ for each terminal t and its desired source W_t where $Z_t = (Y_t^n, W_{H_t})$ is the channel output for terminal node t.

Since the sources are dependent, Lemma 4.5.1 cannot be applied directly because the random coding argument in its proof requires the sources to be independent. Lemma 6.4.1, below, describes a scheme that maps the lossy source code to a lossless network code by first applying linear Slepian-Wolf (SW) encoding scheme[48] (which enables terminals to reconstruct the sources losslessly) and then applying Lemma 5.3.1 (which enables independent sources of a reduced rate to be transmitted when $\mathcal{I}$ is augmented with a cooperation facilitator.) When the sources are linearly dependent and the encoders are linear, Lemma 6.4.1 yields a linear lossless network code.

Lemma 6.4.1. *Let $\mathcal{I} = (G, S, T)$ be a k-unicast network coding network. Let $\{f_e, e \in E\}$ be a set of blocklength n encoders and let $\mathbf{W}$ be a vector of rate $\mathbf{R} = (R_1, \cdots, R_k)$, $n\delta$-dependent source message random variables such that under the operation of $\{f_e, e \in E_{S^c}\}$ on $\mathbf{W}$, the information variable $Z_t = (Y_t^n, W_{H_t})$ received by each terminal $t \in T$ (See Figure 6.2(b)) satisfies*

$$\gamma = \max_{t \in T} \frac{1}{n} H(W_t|Z_t).$$

Then for any $\epsilon' > 0$, there exists a blocklength n' such that $\mathcal{I}_{\lambda'}^{cf}$ is $(\mathbf{R} - \rho', \epsilon', n', 0)$-feasible, where λ' and ρ' are positive numbers tending to zero as δ and γ tend to zero. Further, if $\mathbf{W}$ is linearly dependent and $\{f_e, e \in E_{S^c}\}$ are linear encoders, then for any $\epsilon' > 0$, there exists blocklength n' such that $\mathcal{I}_{\lambda'}^{cf}$ is $(\mathbf{R} - \rho', \epsilon', n', 0)$-linearly-feasible, where λ' and ρ' are positive numbers tending to zero as δ and γ tend to zero.

Proof. Consider the distributed Slepian-Wolf source coding set-up in Figure 6.2(b). Here SW coding is employed to describe sources drawn $i.i.d \sim p(Z_t, W_t)$, where $p(Z_t, W_t)$ is the distribution induced by the operation of $\{f_e, e \in E\}$ on $\mathbf{W}$. For a blocklength N, each terminal t has side-information $\mathbf{Z}_t^N$ and would like to reconstruct $\mathbf{W}_t^N$. By [48, Theorem 1], there exists a linear encoder

$$f_{t,\text{sw}} : \mathbb{F}_2^{nNR_t} \to \mathbb{F}_2^{nNR_t^*} \text{ for each } t \in T$$

such that for any $R_t^* > n\gamma$, each t can reconstruct $\mathbf{W}_t^N$ from the received codeword $f_{t,\text{sw}}(\mathbf{W}_t^N)$ (using a minimum entropy decoder [48]) with error probability ϵ^* which tends to zero as N tends to infinity.

Consider the following communication scheme for $\mathcal{I}$ to transmit $\mathbf{W}_t^N = (W_{t,j}, j \in [N])$ to each $t \in T$:

1. For the first nN time steps, apply the blocklength n encoders $\{f_e, e \in E_{S^c}\}$ a total of N times. For each $j \in [N]$, during the jth block of n timesteps, $\{f_e, e \in E_{S^c}\}$ is applied on $(W_{t,j}, t \in T)$. This yields the information variable $\mathbf{Z}_t^N$ that is received by each terminal $t \in T$ at the end of time nN.

2. For the rest of the time steps, compute and route each SW codeword $f_{t,\text{sw}}(\mathbf{W}_t^N)$ from source node s to terminal t. Do this sequentially for each $(s,t) \in \{(s_i,t_i), i \in [k]\}$ before each terminal reconstructs $\mathbf{W}_t^N$. Assuming we can send a single unicast rate of at least R'_t (via routing) across each source-destination pair (s,t), the total number of time steps needed for this phase is
$$nN\alpha = \sum_{t \in T} \frac{NR_t^*}{R'_t},$$
where $\alpha = \frac{R_t^*}{nR'_t}$ tends to zero as γ tends to zero.

This yields an $(\frac{\mathbf{R}}{1+\alpha}, \epsilon^*, nN(1+\alpha), \delta)$-feasible code for $\mathcal{I}$. By Lemma 5.3.1(1), $\mathcal{I}_\lambda^{cf}$ is $(\frac{\mathbf{R}}{1+\alpha} - \rho, 0, nN(1+\alpha), 0)$-feasible, where α, λ and ρ tend to zero as ϵ^*,γ and δ tend to zero and nN tends to infinity. If $\{f_e, e \in E_{S^c}\}$ are linear encoders and $\mathbf{W}$ is an $n\delta$-linearly-dependent, then the scheme above yields an $(\frac{\mathbf{R}}{1+\alpha}, \epsilon^*, nN(1+\alpha), \delta)$-linearly-feasible code for $\mathcal{I}$. By Lemma 5.3.1(3), $\mathcal{I}_\lambda^{cf}$ is $(\frac{\mathbf{R}}{1+\alpha} - \rho, \epsilon^*, nN(1+\alpha), 0)$-linearly-feasible, where α, λ, and ρ tend to zero as ϵ^*,γ, and δ tend to zero and nN tends to infinity. □

6.5 Proof of Theorem 6.2.1

Proof. We use variables without tildes for $\mathcal{I}$ and variables with tilde for $\tilde{\mathcal{I}}$. It suffices (Theorem 5.5.1) to show the following two "if and only if" statements:
$$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf}) \text{ and } \tilde{\mathbf{R}} \in \mathcal{R}_\epsilon^L(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon^L(\mathcal{I}_\lambda^{bf}).$$

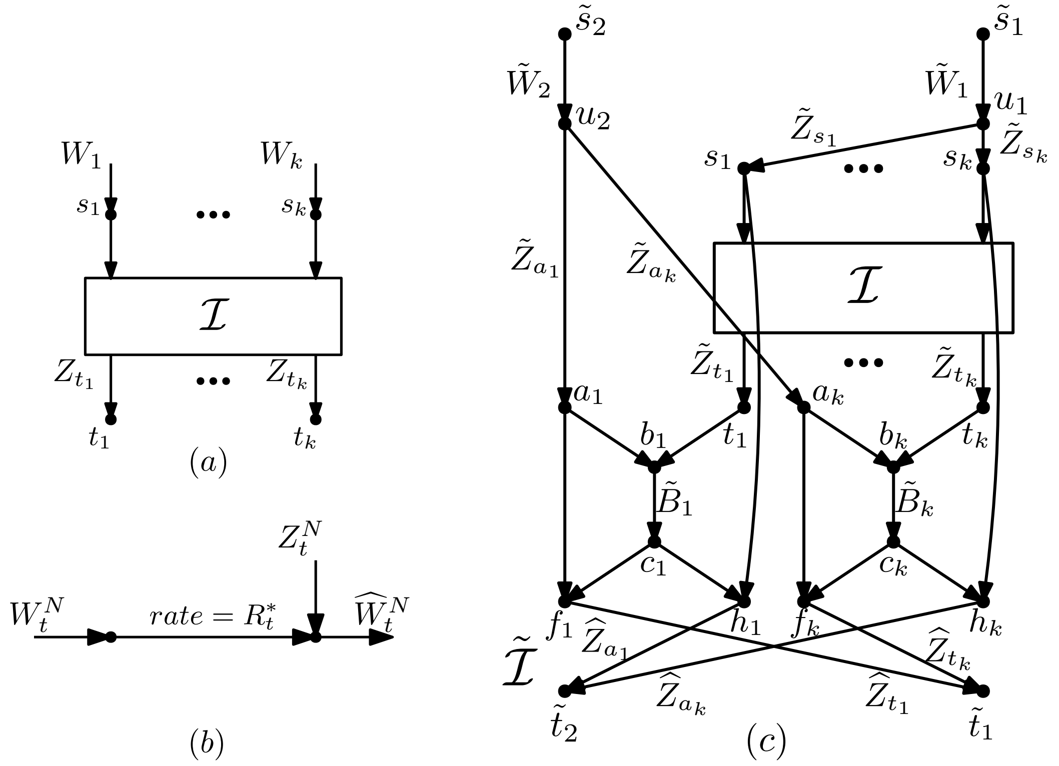


Figure 6.2: (a) A channel used to illustrate the channel in Lemma 6.4.1. (b) A Slepian-Wolf coding scheme for $\mathcal{I}$ to convert a lossy code to a lossless one. (c) Figure of $\tilde{\mathcal{I}}$ labeled with edge information variables.

We prove each statement in two parts, first showing that if a rate vector $\mathbf{R} = (R_1, \cdots, R_k)$ is in the capacity region of $\mathcal{I}_\lambda^{bf}$, then the corresponding rate vector $\tilde{\mathbf{R}} = (R_{\text{sum}}, R_{\text{sum}})$, where $R_{\text{sum}} = \sum\limits_{i \in [k]} R_i$, is in the capacity region of $\tilde{\mathcal{I}}$, and then showing the converse.

We now present the proof of the assertion $\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \lim\limits_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$. The proof for linear capacity follows from that presented since it uses the same reduction network $\tilde{\mathcal{I}}$ and our code reductions preserve code linearity. Parallel arguments for the proof for linear capacity are contained in square brackets (i.e., [...]).

$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Rightarrow \mathbf{R} \in \lim\limits_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$: Fix any $\tilde{\epsilon}, \tilde{\rho} > 0$. We start with an $(\tilde{\mathbf{R}} - \tilde{\rho}, \tilde{\epsilon}, \tilde{n}, 0)$-feasible network code $\tilde{\mathcal{C}}$ for $\tilde{\mathcal{I}}$. Under the operation of $\tilde{\mathcal{C}}$ on network $\tilde{\mathcal{I}}$, let the edge information variables be denoted (capital letters) as in Figure 6.2(c). In particular, let each $\tilde{Z}_s$ be the information variable received by the old source node $s \in S$ and each $\tilde{Z}_t$ be the information variable received by the old terminal node $t \in T$ in $\tilde{\mathcal{I}}$.

By Lemma 6.5.1, to be stated shortly, for each source-destination pair of $\mathcal{I}$, $(s,t) \in \{(s_i,t_i), i \in [k]\}$, we have

$$H(\tilde{Z}_s|\tilde{Z}_t) \leq \tilde{n}\tilde{\gamma},$$

where $\tilde{\gamma}$ tends to zero as $\tilde{\rho}$ and $\tilde{\epsilon}$ tend to zero.

Next, we bound the dependence among the variables $(\tilde{Z}_s, s \in S)$. We have

$$I(\tilde{W}_1; (\tilde{Z}_s, s \in S)) \overset{(a)}{\geq} I(\tilde{W}_1; (\tilde{Z}_t, t \in T)) \overset{(b)}{\geq} \tilde{n}(R_{\text{sum}} - \tilde{\rho} - \tilde{\gamma}')$$

for some $\tilde{\gamma}'$ that tends to zero as $\tilde{\epsilon}$ tends to zero, where (a) is due to the data processing inequality and (b) is due to Fano's inequality. This gives

$$H(\tilde{Z}_s, s \in S) \geq \tilde{n}(R_{\text{sum}} - \tilde{\rho} - \tilde{\gamma}') + H(\tilde{Z}_s, s \in S|\tilde{W}_1) = \tilde{n}(R_{\text{sum}} - \tilde{\rho} - \tilde{\gamma}').$$

Further, since each link carrying $\tilde{Z}_s$ has a capacity of R_s, the support size of each $\tilde{Z}_s$ is bounded above by $2^{\tilde{n}R_s}$, giving $H(\tilde{Z}_s) \leq \tilde{n}R_s$ for each $s \in S$. Hence

$$H(\tilde{Z}_s) \geq H(\tilde{Z}_{s'}, s' \in S) - \sum_{s' \in S \setminus \{s\}} H(\tilde{Z}_{s'}) \geq \tilde{n}(R_s - \tilde{\rho} - \tilde{\gamma}'),$$

$$\left(\sum_{s \in S} H(\tilde{Z}_s)\right) - H(\tilde{Z}_s, s \in S) \leq \tilde{n}(\tilde{\rho} + \tilde{\gamma}').$$

If we consider $(\tilde{Z}_s, s \in S)$ as source message variables, those source message variables would be $\tilde{n}\tilde{\delta}$-dependent for $\tilde{\delta} = \tilde{\rho} + \tilde{\gamma}'$.

By Lemma 6.4.1, for any $\epsilon > 0$, there exists a blocklength n such that $\mathcal{I}_\lambda^{cf}$ is $(\mathbf{R} - \rho, \epsilon, n, 0)$-feasible where ρ and λ tend to zero as $\tilde{\rho}$ and $\tilde{\epsilon}$ tend to zero. This implies that $\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$ as desired. [Note that if $\tilde{\mathcal{C}}$ were a linear code, then $(\tilde{Z}_s, s \in S)$ would be $\tilde{n}\tilde{\delta}$-linearly-dependent with $\tilde{W}_1$ as the underlying random process (See Section 5.2). Lemma 6.4.1 therefore yields a linear code for $\mathcal{I}_\lambda^{cf}$ which in turn implies that $\mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon^L(\mathcal{I}_\lambda^{bf})$.]

$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \leftarrow \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$: Fix any $\epsilon, \rho, \lambda > 0$. We start with an $(\mathbf{R} - \rho, \epsilon, n, 0)$-feasible network code $\mathcal{C}$ for $\mathcal{I}_\lambda^{bf}$. By Lemma 5.3.1(2), $\mathcal{I}$ is $(\mathbf{R} - \rho', 0, n', \delta')$-feasible (under code $\mathcal{C}'$) for some δ' and ρ' that tend to zero as ϵ, λ, and ρ tend to zero. Let $\mathbf{W} = (W_1, \cdots, W_k)$ be the corresponding set of $n\delta'$-dependent sources. [If $\mathbf{R} \in \mathcal{R}_\epsilon^L(\mathcal{I}_\lambda^{bf})$, then $\mathbf{R} - \rho \in \mathcal{R}_\epsilon^L(\mathcal{I})$ for some ρ that tends to zero as λ tends to zero (Theorem 3.2.1). Using the fact that $\mathcal{R}_\epsilon^L(\mathcal{I}) = \mathcal{R}_0^L(\mathcal{I})$ (Theorem 9.1.1), there exists a blocklength n' such that $\mathcal{I}$ is $(\mathbf{R} - \rho', 0, n', 0)$-linearly-feasible (under code $\mathcal{C}'$), where ρ' tends to zero as λ tends to zero.]

Since $\mathcal{C}'$ is a zero error code, any source realization in the support set $\mathcal{W}_{\text{sp}}$ of $\mathbf{W}$ can be transmitted to the terminals without error. Using this fact, consider the following communication scheme for $\tilde{\mathcal{I}}$, in which we transmit independent source messages $\tilde{W}_1$ and $\tilde{W}_2$:

1. At source node $\tilde{s}_2$, the source message $\tilde{W}_2 \in \mathbb{F}_2^{n' R_{\text{sum}}}$ is split into k chunks, giving $\tilde{W}_2 = (\tilde{Z}_{a_1}, \cdots, \tilde{Z}_{a_k})$, where each
$$\tilde{Z}_i \in \mathbb{F}_2^{n' R_i}$$
is transmitted on edge (u_2, a_i) and forwarded to nodes b_i and f_i.

2. We set $\tilde{W}_1$ to be in $\mathbb{F}_2^{n'(R_{\text{sum}} - k\rho' - \delta')}$. At source node $\tilde{s}_1$, the encoder
$$f_{\tilde{s}_1} : \mathbb{F}_2^{n'(R_{\text{sum}} - k\rho' - \delta')} \to \mathcal{W}_{\text{sp}}$$
maps the ith element in $\mathbb{F}_2^{n'(R_{\text{sum}} - k\rho' - \delta')}$ to the ith element in the support set of $\mathbf{W}$, with respect to an arbitrary but fixed ordering of $\mathbb{F}_2^{n'(R_{\text{sum}} - k\rho' - \delta')}$ and $\mathcal{W}_{\text{sp}}$. Note that this mapping is well defined since
$$\log_2 |\mathcal{W}_{\text{sp}}| \geq H(\mathbf{W}) \geq \left(\sum_{i \in [k]} H(W_i) \right) - n'\delta' \geq n'(R_{\text{sum}} - k\rho' - \delta').$$

3. Consider operating $\mathcal{C}'$ on the sub-network $\mathcal{I}$ that is contained in $\tilde{\mathcal{I}}$. That is, treating
$$f_{\tilde{s}_1}(\tilde{W}_1) = (\tilde{W}_{1,1}, \cdots, \tilde{W}_{1,k})$$
as the dependent sources and applying encoders of $\mathcal{C}'$ on nodes $V \setminus T$ and the decoders on each of the terminals $t \in T$. By assumption of the zero-error code $\mathcal{C}'$, each terminal t_i is able to obtain an error-free reconstruction of $\tilde{W}_{1,i}$.

4. The rest of the network applies a "butterfly" network code. For each $i \in [k]$, node t_i forwards $\tilde{W}_{1,i}$ to node b_i, which computes the element-wise binary sum (denoted by operator "+")
$$\tilde{W}_{1,i} + \tilde{W}_{2,i}$$

and forwards it to nodes c_i, f_i, and h_i.

5. Finally, each h_i receives variable $\tilde{W}_{1,i}$ from s_i and extracts $\tilde{W}_{2,i}$ from $\tilde{W}_{1,i} + \tilde{W}_{2,i}$, then transmits it to $\tilde{t}_2$. Similarly, each f_i computes $\tilde{W}_{1,i}$ and transmits it to $\tilde{t}_1$.

Since the butterfly network code introduces no error, this scheme yields a code $\tilde{\mathcal{C}}$ that is $(\tilde{\mathbf{R}} - k\rho' - \delta', 0, n', 0)$-feasible for $\tilde{\mathcal{I}}$. Since ρ' and δ' tend to zero as ϵ, ρ and λ tend to zero, $\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\mathcal{I})$. [Note that if we start with a linear code $\mathcal{C}'$ that is feasible for independent sources, the corresponding support set then becomes $\mathcal{W}_{\text{sp}} = \prod_{i\in[k]} \mathbb{F}_2^{n'(R_i - \rho')}$, the encoding function $f_{\tilde{s}_1}$ described in Step 2) of the scheme can then be made linear. Further, since the butterfly network code described in steps 4) and 5) is also linear, this yields a linear network code $\tilde{\mathcal{C}}$ for $\tilde{\mathcal{I}}$. Thus $\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon^L(\mathcal{I})$.]

□

Lemma 6.5.1. *(Based on [14]) Let* $\tilde{\mathbf{R}} = (\sum_{i=1}^k R_i, \sum_{i=1}^k R_i)$*. If* $\tilde{\mathcal{I}}$ *is* $(\tilde{\mathbf{R}} - \tilde{\rho}, \tilde{\epsilon}, \tilde{n}, 0)$*-feasible, then for all* $i \in [k], H(\tilde{Z}_{s_i}|\tilde{Z}_{t_i}) \le \tilde{n}\tilde{\gamma}$*, where* $\tilde{\gamma}$ *goes to zero as* $\tilde{\rho}$ *and* $\tilde{\epsilon}$ *goes to zero.*

Proof. This proof follows the proof idea from [14] and uses variables from Figure 6.2(c). We bound $I(\tilde{B}_i; \tilde{W}_2)$ as follows,

$$\begin{array}{ll}
I(\tilde{B}_i;\tilde{W}_2) & \\
= I(\tilde{B}_i;\tilde{W}_1,\tilde{W}_2) - I(\tilde{B}_i;\tilde{W}_1|\tilde{W}_2) & \text{By chain rule of mutual information.} \\
= I(\tilde{B}_i;\tilde{W}_1,\tilde{W}_2) - I(\tilde{B}_i,\tilde{W}_2;\tilde{W}_1) & \text{Independence of } \tilde{W}_1 \text{ and } \tilde{W}_2 \\
= I(\tilde{B}_i;\tilde{W}_1,\tilde{W}_2) - I(\tilde{B}_i,\widehat{Z}_{t_i},\tilde{W}_2;\tilde{W}_1) & \widehat{Y}_i \text{ is a function of } \tilde{W}_2, \tilde{B}_i \\
\le \tilde{n}R_i - I(\widehat{Z}_{t_i};\tilde{W}_1) & \\
= \tilde{n}R_i - I((\widehat{Z}_{t_j}, j\in[k]);\tilde{W}_1) & \\
\qquad +I((\widehat{Z}_{t_j}, j\in[k]\setminus\{i\});\tilde{W}_1|\widehat{Z}_{t_i}) & \text{By chain rule of mutual information.} \\
\le \sum_{j\in[k]} \tilde{n}R_j & \text{By decoding condition} \\
\qquad -((\sum_{j\in[k]} \tilde{n}R_j) - \tilde{n}\tilde{\delta}_i) = \tilde{n}\tilde{\delta}_i & \text{and Fano's inequality.}
\end{array}$$

Next, we bound $I(\tilde{B}_i; \tilde{W}_2, \tilde{Z}_{s_i})$,

$$\begin{aligned}
& I(\tilde{B}_i; \tilde{W}_2, \tilde{Z}_{s_i}) & \\
& \geq I(\tilde{B}_i; \tilde{W}_2 | \tilde{Z}_{s_i}) & \\
& = I(\tilde{B}_i, \tilde{Z}_{s_i}; \tilde{W}_2) & \text{Independence of } \tilde{W}_2 \text{ and } \tilde{Z}_{s_i}. \\
& = I(\tilde{B}_i, \tilde{Z}_{s_i}, \widehat{Z}_{a_i}; \tilde{W}_2) & \widehat{Z}_{a_i} \text{ is a function of } \tilde{B}_i, \tilde{Z}_{s_i} \\
& \geq I(\widehat{Z}_{a_i}; \tilde{W}_2) & \\
& = I((\widehat{Z}_{a_j}, j \in [k]); \tilde{W}_2) & \\
& \qquad -I((\widehat{Z}_{a_j}, j \in [k] \setminus \{i\}); \tilde{W}_2 | \widehat{Z}_{a_i}) & \text{By chain rule of mutual information.} \\
& \geq (\sum_{j \in [k]} \tilde{n} R_j - \tilde{n} \tilde{\delta}_i) & \text{By decoding condition} \\
& \qquad - \sum_{j \in [k] \setminus \{i\}} \tilde{n} R_j = \tilde{n} R_i - \tilde{n} \tilde{\delta}_i & \text{and Fano's inequality.}
\end{aligned}$$

Finally we bound $H(\tilde{Y}_{s_i} | \tilde{Y}_{t_i})$,

$$\begin{aligned}
& H(\tilde{Z}_{s_i} | \tilde{Z}_{t_i}) & \\
& = H(\tilde{Z}_{s_i} | \tilde{Z}_{t_i}, \tilde{W}_{\tilde{s}_2}, (\tilde{Z}_{a_j}, j \in [k])) & \text{By independence of} \\
& & (\tilde{W}_{\tilde{s}_2}, (\tilde{Z}_{a_j}, j \in [k]) \text{ with } (\tilde{Z}_{s_i}, \tilde{Z}_{t_i}) \\
& = H(\tilde{Z}_{s_i} | \tilde{Z}_{t_i}, \tilde{W}_2, (\tilde{Z}_{a_j}, j \in [k]), \tilde{B}_i) & \text{Since } \tilde{B}_i \text{ is a function of } \tilde{Z}_{a_i}, \tilde{Z}_{t_i}. \\
& \leq H(\tilde{Z}_{s_i} | \tilde{B}_i, \tilde{W}_2) & \text{Conditioning reduces entropy.} \\
& = H(\tilde{Z}_{s_i} | \tilde{W}_2) - I(\tilde{B}_i; \tilde{Z}_{s_i} | \tilde{W}_2) & \text{Expansion of } I(\tilde{B}_i; \tilde{Z}_{s_i} | \tilde{W}_2). \\
& = H(\tilde{Z}_{s_i} | \tilde{W}_2) & \\
& \qquad -I(\tilde{B}_i; \tilde{W}_2, \tilde{Z}_{s_i}) + I(\tilde{B}_i; \tilde{W}_2) & \text{By chain rule of mutual information.} \\
& \leq 2 \tilde{n} \tilde{\delta}_i &
\end{aligned}$$

Here each $\tilde{\delta}_i$ goes to zero as $\tilde{\rho}$ and $\tilde{\epsilon}$ goes to zero. It suffices to set $\tilde{\gamma} = \max_{i \in [k]} 2\tilde{\delta}_i$. □

Chapter 7

REDUCTION FROM NETWORK CODING TO INDEX CODING

The materials of this chapter are published in part as [49].

The index coding problem [5] is a special case of the network coding problem that can be interpreted as a "broadcast with side information" problem: a broadcast node has access to all sources and wishes to communicate with several terminals, each having and desiring to reconstruct potentially different sets of sources.

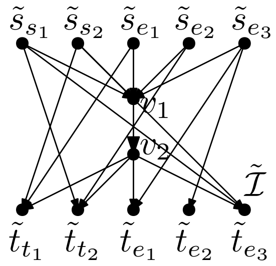
A code reduction from acyclic network coding to index coding is derived in [7]. Thus, any efficient scheme that solves all index coding problems would yield an efficient scheme that solves all acyclic general network coding problems. Although the connection between network coding and index coding presented in [7] is very general, it does not resolve the question of whether the network coding capacity region can be obtained by solving the capacity region of a corresponding index coding problem.

In this work, we show that the capacity reduction from acyclic network coding to index coding is equivalent to the AERS. Section 7.1 describes the mapping Φ_3 from an acyclic network coding instance to an index coding instance. We describe the main result in Section 7.2 and give its proof in Section 7.3.

7.1 Reduction Mapping $\Phi3$

We begin by describing the reduction from $\mathcal{I}$ to $\tilde{\mathcal{I}}$ employed in [6], modified slightly here in order to fit our model. Note that in the reduction of [6], the instance $\tilde{\mathcal{I}}$ depends only on $\mathcal{I}$ (and not on the parameters n and ϵ as permitted by Definition 1).

Given network coding instance $\mathcal{I} = (G, S, T)$ with topology $G = (V, E, C)$ and given any rate vector $\mathbf{R}$, we define index coding problem $\tilde{\mathcal{I}} = (\tilde{S}, \tilde{T}, \tilde{H}, \tilde{c}_B)$ and rate vector $\tilde{\mathbf{R}}$ as follows. The source set $\tilde{S}$ contains one source node $\tilde{s}_s$ for each source node $s \in S$ and one source node $\tilde{s}_e$ for each edge e that is not a



Source Nodes	$\tilde{S} = \{\tilde{s}_{s_1}, \tilde{s}_{s_2}, \tilde{s}_{e_1}, \tilde{s}_{e_2}, \tilde{s}_{e_3}\}$
Destination Nodes	$\tilde{t} = \{\tilde{t}_{t_1}, \tilde{t}_{t_2}, \tilde{t}_{e_1}, \tilde{t}_{e_2}, \tilde{t}_{e_3}\}$
Side Information Sets	$\tilde{H}_{\tilde{t}_{t_1}} = \{\tilde{s}_{s_2}, \tilde{s}_{e_1}\}$ $\tilde{H}_{\tilde{t}_2} = \{\tilde{s}_{s_1}, \tilde{s}_{e_2}\}$ $\tilde{H}_{e_3} = \{\tilde{s}_{s_1}, \tilde{s}_{s_2}\}$ $\tilde{H}_{e_1} = \{\tilde{s}_{e_3}\}, \tilde{H}_{e_2} = \{\tilde{s}_{e_3}\}$
Broadcast Capacity	$\tilde{c}_B = 3$
Rate	$\tilde{\mathbf{R}} = (\mathbf{R}, 1, 1, 1)$

Figure 7.1: The index coding network corresponding to the "butterfly" like network in Figure 2.2 in Section 2.2.

source edge, giving

$$\tilde{S} = \{\tilde{s}_s : s \in S\} \cup \{\tilde{s}_e : e \in E_{S^c}\}.$$

Similarly, terminal set $\tilde{T}$ has one terminal $\tilde{t}_t$ for each terminal $t \in T$ and one terminal $\tilde{t}_e$ for each edge e that is not a source edge, giving

$$\tilde{T} = \{\tilde{t}_t : t \in T\} \cup \{\tilde{t}_e : e \in E_{S^c}\}.$$

The "has" set $\tilde{H}_{\tilde{t}}$ for terminal $\tilde{t}$ varies with the terminal type. When $\tilde{t} = \tilde{t}_e$ for some edge $e \in E_{S^c}$, $\tilde{H}_{\tilde{t}}$ includes the source nodes $\tilde{s}_{e'}$ for all edges e' incoming to e and source nodes $\tilde{s}_s$ for all source edge $e' = (s, \mathrm{In}(e)) \in E$ incoming to e in G; when $\tilde{t} = \tilde{t}_t$ for some terminal $t \in T$, $\tilde{H}_{\tilde{t}}$ includes the source nodes $\tilde{s}_{e'}$ for all edges e' incoming to t and source nodes $\tilde{s}_s$ for all source edge $e' = (s, t) \in E$ incoming to t in G. Thus

$$\tilde{H}_{\tilde{t}} = \begin{cases} \{\tilde{s}_{e'} : \mathrm{Out}(e') = \mathrm{In}(e)\} \cup \{\tilde{s}_s : (s, \mathrm{In}(e)) \in E\} & \text{if } \tilde{t} = \tilde{t}_e \text{ for some } e \in E_{S^c} \\ \{\tilde{s}_e : \mathrm{Out}(e) = t\} \cup \{\tilde{s}_s : (s, t) \in E\} & \text{if } \tilde{t} = \tilde{t}_t \text{ for some } t \in T. \end{cases}$$

The bottleneck capacity $\tilde{c}_B$ is set to the sum of all finite edge capacities in $\mathcal{I}$, giving

$$\tilde{c}_B = \sum_{e \in E_{S^c}} c_e.$$

The rate vector $\mathbf{R}$ is mapped to

$$\tilde{\mathbf{R}} = (\mathbf{R}, (c_e : e \in E_{S^c})).$$

An example is shown in Figure 7.1, which gives the index coding network $\tilde{\mathcal{I}}$ corresponding to the butterfly network from Figure 2.2 of Chapter 2.

7.2 Main Result

The authors in [7] pose the question of whether code reduction Φ_3 can be used to derive a corresponding capacity reduction. Theorem 7.2.1 gives a partial solution to that question.

Theorem 7.2.1 (Capacity Reduction from Network Coding to Index Coding)**.**

1. *Linear capacity characterization for network coding reduces to linear capacity characterization for index coding. That is, under mapping Φ_3, for any acyclic network coding instance $\mathcal{I}$ and rate vector $\mathbf{R}$,*

$$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon^L(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon^L(\mathcal{I}).$$

2. *Capacity characterization for network coding reduces to capacity characterization for index coding under mapping Φ_3 if and only if the **AERS** holds. That is, for any acyclic network coding instance $\mathcal{I}$ and rate vector $\mathbf{R}$,*

$$\left(\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})\right) \text{ if and only if the } \textbf{AERS} \text{ holds.}$$

Since the question of whether the edge removal statement is always true or sometimes false is unresolved, the question of capacity reduction between network coding and index coding remains open in the general case. However, our result provides another way to understand the capacity region of network coding problems via the edge removal statement.

7.3 Proof of Theorem 7.2.1

Proof. We first give a high level description of the proof. Throughout this proof, we use "untilded" variables for $\mathcal{I}$ and "tilded" variables for $\tilde{\mathcal{I}}$. It suffices (Theorem 5.5.1) to show the following two "if-and-only-if" statements:

$$\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \lim_{\lambda\to 0}\mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf}) \text{ and } \tilde{\mathbf{R}} \in \mathcal{R}_\epsilon^L(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \lim_{\lambda\to 0}\mathcal{R}_\epsilon^L(\mathcal{I}_\lambda^{bf}).$$

To prove each of the above two statements, we present two proof "directions". In the first direction, we show that if a rate vector $\mathbf{R} = (R_1, \cdots, R_k)$ is in the capacity region of $\mathcal{I}_\lambda^{bf}$, then the corresponding rate vector $\tilde{\mathbf{R}} = (\mathbf{R}, (c_e, e \in E_{S^c}))$ is in the capacity region of $\tilde{\mathcal{I}}$. We show the converse in the second direction.

The proof uses the idea of code reduction, in which we transform an $(\mathbf{R}(1-\rho), \epsilon, n)$ network code $\mathcal{C}$ for $\mathcal{I}_\lambda^{bf}$ into a rate $(\tilde{\mathbf{R}}(1-\tilde{\rho}), \tilde{\epsilon}, \tilde{n})$ network code $\tilde{\mathcal{C}}$ for $\tilde{\mathcal{I}}$ and vice versa. The $\tilde{\rho}$-loss in rate tends to zero as ρ, ϵ tends to zero and the blocklength $\tilde{n}$ tends to infinity. By taking the closure of these rates, we get the desired result. We now present the proof of the assertion $\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftrightarrow \mathbf{R} \in \lim\limits_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$. The proof for linear capacity follows from that presented since it uses the same reduction network $\tilde{\mathcal{I}}$ and our code reductions preserve code linearity. Parallel arguments for the proof for linear capacity will be contained in square brackets (i.e., [...]).

First direction: $\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Rightarrow \mathbf{R} \in \lim\limits_{\lambda \to 0} \mathcal{R}_\epsilon(\mathcal{I}_\lambda^{bf})$. Define $\tilde{\mathbf{W}}_S = (\tilde{W}_s, s \in S)$ and $\tilde{\mathbf{W}}_{E_{S^c}} = (\tilde{W}_e, e \in E_{S^c})$. Fix any $\epsilon, \rho > 0$, we start with a code $\tilde{\mathcal{C}}$ that is $(\tilde{\mathbf{R}}(1-\tilde{\rho}), \tilde{\epsilon}, \tilde{n})$-feasible for $\tilde{\mathcal{I}}$. For any non-source node $v \in \overline{S}$, denote by

$$\tilde{\mathbf{W}}_{H_v} = \Big((\tilde{W}_{e'}, e' \in E : \text{Out}(e') = v), (\tilde{W}_{s'}, s' \in S : (s', v) \in E_S) \Big)$$

the vector of source and edge messages in the has set $\tilde{H}_v$ of v in $\tilde{\mathcal{I}}$. Let the broadcast encoder be denoted by

$$\tilde{f}_B(\tilde{\mathbf{W}}_S, \tilde{\mathbf{W}}_{E_{S^c}})$$

and the decoders be denoted by,

$$\tilde{g}_{\tilde{t}_e}(\tilde{X}_B, \mathbf{W}_{H_{\text{In}(e)}})$$
$$\tilde{g}_{\tilde{t}_t}(\tilde{X}_B, \mathbf{W}_{H_t}).$$

We design a code $\mathcal{C}$ that operates on $\mathcal{I}_\lambda^{bf}$ in two phases, reusing these functions:

1. In the first phase, the super-node s_{su} broadcasts an overhead message

   $$X_\alpha = \tilde{f}_B(\mathbf{W}, F_{E_{S^c}}(\mathbf{W}))$$

   to nodes in $\overline{S}$ of $\mathcal{I}_\lambda^{bf}$, where

   $$F_{E_{S^c}} : \prod_{s \in S} \mathbb{F}_2^{\tilde{n} R_s (1-\tilde{\rho})} \to \prod_{e \in E_{S^c}} \mathbb{F}_2^{\tilde{n} c_e (1-\tilde{\rho})}$$

   is a suitable function that maps a realization of $\tilde{\mathbf{W}}_S$ to a realization of $\tilde{\mathbf{W}}_{E_{S^c}}$.

2. In the second phase, the sources $\mathbf{W}$ are transmitted through the rest of $\mathcal{I}^{bf}_{\lambda}$ by having the following encoding function for each $e \in E_{S^c}$,

   $$f_e(Z_{\mathrm{In}(e)}) = \tilde{g}_{\tilde{t}_e}(X_\alpha, Z_{\mathrm{In}(e)}).$$

   Each terminal t implements the decoding function

   $$\widehat{W}_t = \tilde{g}_{\tilde{t}_t}(X_\alpha, Z_t).$$

   Note that if the original index code operates without error on a message realization $(\tilde{\mathbf{W}}_S, \tilde{\mathbf{W}}_{E_{S^c}}) = (\tilde{\mathbf{w}}_S, F_{E_{S^c}}(\tilde{\mathbf{w}}_S))$, then by induction on the topological order of G, $\mathcal{C}$ will also be able to operate without error on message realization $\mathbf{W} = \tilde{\mathbf{w}}_S$.

By Lemma 7.3.1, stated shortly, there exists a function $f_{E_{S^c}}$ so that the super-node in $\mathcal{I}^{bf}_{\lambda}$ can broadcast the overhead message X_α using only a small alphabet Σ and that the coding scheme described above operates with error at most $2\tilde{\epsilon}$, where $\tilde{n}^{-1}\log|\Sigma|$ goes to zero as $\tilde{n}$ goes to infinity and $\tilde{\rho}$ goes to zero. Therefore, $\forall \lambda, \tilde{\rho}, \tilde{\epsilon} > 0$, $\mathcal{I}^{bf}_{\lambda}$ is $(\mathbf{R}(1-\tilde{\rho}), 2\tilde{\epsilon}, \tilde{n})$-feasible for large enough $\tilde{n}$, which also means that $\mathbf{R} \in \lim\limits_{\lambda\to 0}\mathcal{R}_\epsilon(\mathcal{I}^{bf}_{\lambda})$. [Note that if $\tilde{\mathcal{C}}$ were linear, then Lemma 7.3.2 implies a linear encoder $F_{E_{S^c}}$ (of a rate that tends to zero as $\tilde{\rho}$ tends to zero) for the broadcast facilitator, which will yield a linear $\mathcal{C}$. Hence $\mathbf{R} \in \lim\limits_{\lambda\to 0}\mathcal{R}^L_\epsilon(\mathcal{I}^{bf}_{\lambda})$.]

Second direction: $\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}) \Leftarrow \mathbf{R} \in \lim\limits_{\lambda\to 0}\mathcal{R}_\epsilon(\mathcal{I}^{bf}_{\lambda})$. Fix any $\epsilon, \rho > 0$. We start with an $(\mathbf{R}(1-\rho), \epsilon, n)$-feasible code for $\mathcal{I}^{bf}_{\lambda}$. Denote this code by

$$\mathcal{C}_\lambda = (\{f_e\}_{e\in E_\lambda}, \{g_t\}_{t\in T}),$$

where E_λ is the set of edges in instances $\mathcal{I}$ and $\mathcal{I}^{bf}_{\lambda}$, respectively. Let $\tilde{\mathcal{I}}$ be the corresponding index coding instance for $\mathcal{I}$ and let $\tilde{\mathcal{I}}_\lambda$ be the index coding instance obtained from $\tilde{\mathcal{I}}$ by adding an extra λ to the capacity of the bottleneck link. Let $\{F_e\}_{e\in E_\lambda}$ be the set of global encoding functions corresponding to $\mathcal{C}_\lambda$ and let α be the bottleneck edge of the broadcast facilitator in $\mathcal{I}^{bf}_{\lambda}$. Following [7], we construct an index code $\tilde{\mathcal{C}}$ for $\tilde{\mathcal{I}}_\lambda$ by reusing the code for $\mathcal{I}$, concatenating it with a linear outer code as follow.

1. Let "+" denote the element-wise binary addition operator. We decompose the broadcast encoder $\tilde{f}_B(\mathbf{W}_S, \mathbf{W}_{E_{S^c}}) = \tilde{X}_B$ into components and

define each component,

$$\begin{aligned}\tilde{X}_B &= (\tilde{X}_{B,e}, e \in E_{S^c} \cup \{\alpha\}) \\ \tilde{X}_{B,e} &= \begin{cases} \tilde{W}_e + F_e(\tilde{\mathbf{W}}_S) & e \in E_{S^c} \\ F_\alpha(\tilde{\mathbf{W}}_S) & e = \alpha. \end{cases}\end{aligned}$$

2. At the decoders, for each edge terminal $\tilde{t} = \tilde{t}_e$, each decoder $\tilde{g}_{\tilde{t}}$ first computes $F_e(\tilde{\mathbf{W}}_S)$ using its side information and the broadcast message,

$$F_e(\tilde{\mathbf{W}}_S) = f_e((\tilde{X}_{B,e'} + \tilde{W}_{e'}, e' : \text{Out}(e') = \text{In}(e)), \tilde{\mathbf{W}}_{H_{\text{In}(e)}})$$

and then finally obtains $\widehat{\tilde{W}}_e = \tilde{X}_{B,e} + F_e(\tilde{\mathbf{W}}_S)$. Similarly, for each $\tilde{t} = \tilde{t}_t$ that demands $\tilde{W}_s$, each decoder $\tilde{d}_{\tilde{t}}$ outputs a reconstruction of $\tilde{W}_t$ by applying the decoders from $\mathcal{C}_\lambda$

$$\widehat{\tilde{W}}_s = g_t((\tilde{X}_{B,e'} + \tilde{W}_{e'}, e' : \text{Out}(e') = t), \tilde{\mathbf{W}}_{H_t}).$$

Note that this index code operates without error on inputs $(\tilde{\mathbf{W}}_S, \tilde{\mathbf{W}}_{E_{S^c}}) = (\tilde{\mathbf{w}}_S, \tilde{\mathbf{w}}_{E_{S^c}})$ if and only if the original network code operates without error on inputs $\mathbf{W} = \tilde{\mathbf{w}}_S$.

Therefore, for all $\lambda, \rho, \epsilon > 0$, there exists blocklength n such that $\tilde{\mathcal{I}}_\lambda$ is $(\tilde{\mathbf{R}}(1-\rho), \epsilon, n)$-feasible. Thus, $\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}}_\lambda)$. By Lemma 7.3.3, stated shortly, we have $\tilde{\mathbf{R}}(1 - \frac{\lambda}{c_B}) \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}})$ for all λ. Since capacity regions are closed, $\tilde{\mathbf{R}} \in \mathcal{R}_\epsilon(\tilde{\mathcal{I}})$. [If we start with a linear $\mathcal{C}_\lambda$, the resulting $\tilde{\mathcal{C}}_\lambda$ will also be linear since the outer code described above is linear. Thus $\tilde{\mathcal{I}}_\lambda$ is $(\tilde{\mathbf{R}}(1-\rho), \epsilon, n)$-linearly-feasible. By Theorem 3.2.1, $\tilde{\mathcal{I}}$ is $(\tilde{\mathbf{R}}(1-\tilde{\rho}), \epsilon, n)$-linearly-feasible where $\tilde{\rho}$ tends to zero as ρ and λ tends to zero, which implies $\tilde{\mathbf{R}} \in \mathcal{R}^L_\epsilon(\tilde{\mathcal{I}})$.] □

Lemma 7.3.1. *([7, Claim 1]) Let $\mathcal{I} = (G, S, T)$ be a network coding instance and let $\tilde{\mathcal{I}} = (\tilde{S}, \tilde{T}, \tilde{H}, \tilde{c}_B)$ be its corresponding index coding instance according to Φ_3. For any index code that is $(\tilde{\mathbf{R}}(1-\tilde{\rho}), \tilde{\epsilon}, \tilde{n})$-feasible on $\tilde{\mathcal{I}}$, there exists a function*

$$F_{E_{S^c}} : \prod_{s\in S} \mathbb{F}_2^{\tilde{n}R_s(1-\tilde{\rho})} \to \prod_{e \in E_{S^c}} \mathbb{F}_2^{\tilde{n}c_e(1-\tilde{\rho})}$$

and a set $\Sigma \subset \mathbb{F}_2^{\tilde{n}\tilde{c}_B}$ satisfying $|\Sigma| \leq 4\tilde{n}(1-\tilde{\rho})(\sum_{s\in S} R_s)2^{n\tilde{\rho}\tilde{c}_B}$ such that at least a $(1-2\tilde{\epsilon})$ fraction of source realizations $\tilde{\mathbf{w}}_S \in \prod_{s\in S} \mathbb{F}_2^{\tilde{n}R_s(1-\tilde{\rho})}$ satisfy

$$\tilde{f}_B(\tilde{\mathbf{w}}_S, F_{E_{S^c}}(\tilde{\mathbf{w}}_S)) \in \Sigma$$

and the index code operate without error on message realization

$$(\tilde{\mathbf{W}}_S, \tilde{\mathbf{W}}_{E_{S^c}}) = (\tilde{\mathbf{w}}_S, F_{E_{S^c}}(\tilde{\mathbf{w}}_S)).$$

Lemma 7.3.2. *Let $\mathcal{I} = (G, S, T)$ be a network coding instance and let $\tilde{\mathcal{I}} = (\tilde{S}, \tilde{T}, \tilde{H}, \tilde{c}_B)$ be its corresponding index coding instance according to Φ_3. For any linear index code that is $(\tilde{\mathbf{R}}(1-\tilde{\rho}), 0, \tilde{n})$-feasible on $\tilde{\mathcal{I}}$, there exists a linear transformation matrix $F_{E_{S^c}}$,*

$$F_{E_{S^c}} : \prod_{s\in S} \mathbb{F}_2^{\tilde{n}R_s(1-\tilde{\rho})} \to \prod_{e\in E_{S^c}} \mathbb{F}_2^{\tilde{n}c_e(1-\tilde{\rho})},$$

and a linear subspace $\Sigma \subset \mathbb{F}_2^{\tilde{n}\tilde{c}_B}$ satisfying dim$(\Sigma) \leq \tilde{n}\tilde{\rho}\tilde{c}_B$ such that all $\tilde{\mathbf{w}}_S \in \prod_{s\in S} \mathbb{F}_2^{\tilde{n}R_s(1-\tilde{\rho})}$ satisfy

$$\tilde{f}_B(\tilde{\mathbf{w}}_S, \tilde{\mathbf{w}}_S F_{E_{S^c}}) \in \Sigma.$$

Proof. Let

$$\tilde{f}_B(\tilde{\mathbf{w}}_S, \tilde{\mathbf{w}}_{E_{S^c}}) = \tilde{\mathbf{w}}_S \tilde{f}_S + \tilde{\mathbf{w}}_{E_{S^c}} \tilde{f}_{E_{S^c}},$$

where $\tilde{f}_S$ and $\tilde{f}_{E_{S^c}}$ are $\sum_{s\in S} \tilde{n}R_s(1-\tilde{\rho}) \times \tilde{n}\tilde{c}_B$ and $\tilde{n}\tilde{c}_B(1-\tilde{\rho}) \times \tilde{n}\tilde{c}_B$ matrices over $\mathbb{F}_2$, respectively.

For any matrix M, denote by $M(i)$ row i of M and by $RS(M)$ the rowspace of M. For vector spaces V and W, let $V + W = \{v + w | v \in V, w \in W\}$. Let $\tilde{B}_{E_{S^c}} \subseteq [\tilde{n}\tilde{c}_B(1-\tilde{\rho})]$ such that $\{\tilde{F}_{E_{S^c}}(i)\}_{i\in\tilde{B}_{E_{S^c}}}$ forms a basis for $RS(\tilde{f}_{E_{S^c}})$. Let $\tilde{B}_{S1} \subseteq [\sum_{s\in S} \tilde{n}R_s(1-\tilde{\rho})]$ such that

$$\left[\bigcup_{i\in\tilde{B}_{S1}} \{\tilde{f}_S(i)\}\right] \cup \left[\bigcup_{i\in\tilde{B}_E} \{\tilde{f}_{E_{S^c}}(i)\}\right]$$

forms a basis for $RS(\tilde{f}_S) + RS(\tilde{f}_{E_{S^c}})$. Since the index code is assumed to be zero-error, $\tilde{f}_{E_{S^c}}$ is full rank, we must have $|\tilde{B}_{S1}| \leq \tilde{n}\tilde{\rho}\tilde{c}_B$ or we would have more than $\tilde{n}\tilde{c}_B$ independent vectors in $\mathbb{F}_2^{\tilde{n}\tilde{c}_B}$. Therefore, $\tilde{f}_S(i)$ can be decomposed as follows:

$$\tilde{f}_S = \tilde{f}_{S1} + \tilde{f}_{S2},$$

where for each $j \in [\sum_{s\in S} \tilde{n}R_s]$, $\tilde{f}_{S1}(j)$ is in the linear span of $\{\tilde{f}_S(i)\}_{i\in\tilde{B}_{S1}}$ and $\tilde{f}_{S2}(j) \in RS(\tilde{f}_{E_{S^c}})$.

We observe that rank$(\tilde{f}_{S1}) \leq \tilde{n}\tilde{\rho}\tilde{c}_B$. Since $RS(\tilde{f}_{S2}) \subseteq RS(\tilde{f}_{E_{S^c}})$, we can find a matrix $F_{E_{S^c}}$ that satisfies

$$F_{E_{S^c}} \tilde{f}_{E_{S^c}} = -\tilde{f}_{S2}.$$

We therefore have for any $\tilde{\mathbf{w}}_S \in \prod_{s \in S} \mathbb{F}_2^{\tilde{n}R_s(1-\tilde{\rho})}$

$$\tilde{f}_B(\tilde{\mathbf{w}}_S, \tilde{\mathbf{w}}_S F_{E_{S^c}}) = \tilde{\mathbf{w}}_S \tilde{f}_S + \tilde{\mathbf{w}}_S F_{E_{S^c}} \tilde{f}_{E_{S^c}} = \tilde{\mathbf{w}}_S \tilde{f}_{S1}.$$

□

Lemma 7.3.3. *[24, Lemma 4] For any $0 < \kappa < 1$ and network coding instance $\mathcal{I}$, let $\mathcal{I}(\kappa)$ be obtained from $\mathcal{I}$ by multiplying the capacity value of each edge of $\mathcal{I}$ by κ. Then for any rate vector $\mathbf{R}$, $\mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})$ implies $\mathbf{R}\kappa \in \mathcal{R}_\epsilon(\mathcal{I}(\kappa))$.*

Chapter 8

THE TIGHTNESS OF THE YEUNG NETWORK CODING OUTER BOUND

The materials of this chapter are published in part as [50].

An outer bound [15, Theorem 15.9] and an exact characterization [16] of the capacity region for network coding are known, whether these regions differ remains an open problem. These bounds are derived based on the notion of entropic vectors and the entropic region Γ^* (Section 8.1). So far, and there exists no full characterization on the entropic region Γ^*, there is no known algorithm to evaluate these bounds. Hence these results are known as implicit characterizations.

Throughout the paper, we shall refer to the network coding outer bound developed in [15, Theorem 15.9] as the Yeung outer bound. The Yeung outer bound is tight when all sources are colocated [17] and for single-source network coding. Whether the outer bound is tight in general remains open. Computationally efficient outer bounds are developed in [18].

The tightness of the Yeung outer bound can be expressed in a form that is similar to the definition of a capacity reduction (Theorem 8.3.1). We apply tools from Chapter 5 and show that the Yeung outer bound is tight if and only of the AERS holds. Before describing these bounds, we begin by defining the entropic region.

8.1 Entropic Regions

An approach to characterize the network coding capacity region is to find all information inequalities. For a long time, the knowledge of information inequalities are limited to the basic inequalities due to Shannon [1] (Shannon inequalities). Shannon inequalities has the advantage of being able to be verified by a linear program [46]. Discovery of non-Shannon inequities [51] calls for the definition of the entropic region, which in principle is capable of verifying all information inequalities. Unfortunately, this region cannot be computed explicitly.

In the following, we first define the entropic region $\Gamma^*(\cdot)$. For consistency, we employ notation from [15], [16]. For a given network coding problem $\mathcal{I}$, let

$$\mathcal{N}(\mathcal{I}) = \{W_s, s \in S; X_e, e \in E\} = \{\mathbf{W}_S, \mathbf{X}_E\}$$

be a collection of discrete random variables corresponding to the source message random variables and the edge information random variables. When the underlying network $\mathcal{I}$ is clear, we simply denote $\mathcal{N}(\mathcal{I})$ by $\mathcal{N}$. Let $\mathcal{H}_{\mathcal{N}}$ be the $|2^{|\mathcal{N}|} - 1|$-dimensional Euclidean space where $\mathbf{h} \in \mathcal{H}_{\mathcal{N}}$ consists of entries h_A labeled by $A \subseteq \mathcal{N}$, for $A \neq \varnothing$. A vector $\mathbf{h} \in \mathcal{H}_{\mathcal{N}}$ is called an entropic vector (or an entropic function) if there exists a set of $|\mathcal{N}|$ random variables such that $\forall A \subseteq \mathcal{N}, A \neq \varnothing$,

$$h_A = H(A).$$

The set of entropic vectors corresponding to $\mathcal{N}$ is denoted by $\Gamma^*(\mathcal{N})$. When the set $\mathcal{N}$ is implied, we simplify $\Gamma^*(\mathcal{N})$ to Γ^*.

Quasi-uniform and Individually-uniform Entropic Region

In this section, we introduce two subsets of the entropic region, namely the quasi-uniform and the individually-uniform entropic region. The quasi-uniform entropic region is employed in [17] to show that the Yeung outer bound is tight when sources are collocated. Our definition of individually-uniform entropic region is inspired by [17]. The individually-uniform entropic region enables us to derive an implicit characterization of the zero-error capacity region (Chapter 9).

We first give the definitions of quasi-uniform entropic vectors and entropic region. A set of random variable $\{X_1, ..., X_n\}$ is quasi-uniform if for any subset $\alpha \subseteq [n]$, $(X_\alpha) = (X_i, i \in \alpha)$ is uniformly distributed over its support ($\mathrm{s}p(X_\alpha)$), or equivalently,

$$H(X_\alpha) = \log |\mathrm{s}p(X_\alpha)|. \tag{8.1}$$

Define $\Gamma^*_Q(\mathcal{N})$ to be the set of all entropic vectors that correspond to random vectors $\mathcal{N}$ that are quasi-uniform. When the set $\mathcal{N}$ is implied, we simplify $\Gamma^*_Q(\mathcal{N})$ to Γ^*_Q.

The requirement of (8.1) allows a quasi-uniform variable X_i to be transmitted across an edge of capacity $H(X_i)$ by sending the indices of $\mathrm{s}p(X_i)$. This is a useful property in mapping an entropic vector directly to a network code in the proof of Theorem 8.3.1.

In the derivation of the zero-error network coding region, one crucial step in the proof is to map a particular network code to a vector in the entropic region. Since not all network code corresponds to a quasi-uniform random variable, we relax the condition in (8.1) and define *individually uniform* variables. A set of random variables $\{X_1, ..., X_n\}$ is individually-uniform if for any $i \in [n]$, the random variable X_i is uniform over its support, or equivalently,

$$H(X_i) = \log |\mathrm{s}p(X_i)|.$$

Define $\Gamma^*_U(\mathcal{N})$ to be the set of all entropic vectors that correspond to random vectors $\mathcal{N}$ that are individually-uniform. When the set $\mathcal{N}$ is implied, we simplify $\Gamma^*_U(\mathcal{N})$ to Γ^*_U.

The definition of individually-uniform random variables relaxes that of quasi-uniform random variables by allowing variables with joint distributions that are not uniform. Since $\Gamma^*_Q \subseteq \Gamma^*_U \subseteq \Gamma^*$, [17, Proposition 2] implies that Γ^*_U is also dense in Γ^* (and $\overline{\Gamma^*}$) in the sense that for every element $\mathbf{h} \in \Gamma^*$ and every $\epsilon > 0$ there is an element $\mathbf{h}' \in \Gamma^*_U$ within Euclidean distance ϵ of $\mathbf{h}$.

8.2 The Yeung Outer Bound $\mathcal{R}_{\mathbf{out}}$

We define the following sub-spaces that will be used to describe the Yeung outer bound [15] and the zero-error capacity region in Chapter 9. We denote linear sub-spaces of $\mathcal{H}_{\mathcal{N}}$ by L_i and the intersection of $L_1, L_2, ..., L_m$ by $L_{12...m}$. For $A, B \subset \mathcal{N}$, define $h_{A|B} = h_{A\cup B} - h_B$.

- The sub-space

$$L_1(\mathcal{I}) = \left\{\mathbf{h} \in \mathcal{H}_{\mathcal{N}} : h_{\mathbf{W}_S} - \sum_{s\in S} h_{W_s} = 0\right\}$$

  describes the set of entropic vectors that corresponds to independent source message variables $\{h_{W_s}, s \in S\}$.

- For $e \in E$, define $Z_{\mathrm{In}(e)} = (X_{e'}, e' \in E, \mathrm{Out}(e') = \mathrm{In}(e))$. The sub-spaces

$$L_2(\mathcal{I}) = \left\{\mathbf{h} \in \mathcal{H}_{\mathcal{N}} : \forall e = (s, v) \in E, s \in S, h_{X_e|W_s} = 0\right\}$$
$$L_3(\mathcal{I}) = \left\{\mathbf{h} \in \mathcal{H}_{\mathcal{N}} : \forall e \in E, \mathrm{In}(e) \notin S, h_{X_e|Z_{\mathrm{In}(e)}} = 0\right\}$$

  describe the set of entropic vectors whose edge information variables match the topology of $\mathcal{I}$. That is, the edge information sent on edge e must be a function of the edge variables entering $\mathrm{In}(e)$ or of the source message variable originating at $\mathrm{In}(e)$.

- The sub-space
$$L_4(\mathcal{I}) = \left\{\mathbf{h} \in \mathcal{H}_{\mathcal{N}} : \forall e, h_{X_e} \leq c_e\right\}$$
describes the set of entropic vectors whose edge information variables satisfy each edge capacity constraint (i.e., c_e).

- For $t \in T$, define $Y_t = (\mathbf{X}_{e'}, e' \in E, \mathrm{Out}(e') = t)$. The sub-space
$$L_5(\mathcal{I}) = \left\{\mathbf{h} \in \mathcal{H}_{\mathcal{N}} : \forall t_i \in T, h_{W_{s_i}|Z_{t_i}} = 0\right\}$$
describes the set of entropic vectors whose edge information variables satisfy the decoding condition at each terminal t. That is, the edge information variables entering t must contain enough information to reconstruct its desired sources.

The authors of [17] observed a connection between the tightness of the Yeung outer bound and the AERS. They proved that the outer bound is tight if the AERS holds. Inspired by their work, we extend the result to an "if-and-only-if" relationship. We begin by defining the Yeung outer bound.

Theorem 8.2.1 (Yeung Network Coding Entropic Function Outer Bound [15, Theorem 15.9])**.** *For a given network coding problem $\mathcal{I}$, an outer bound to its capacity region is given by*

$$\mathcal{R}_{out} = \Omega(Proj_{\mathbf{W}_S}(\overline{D(\Gamma^*)} \cap L_{12345}))).$$

The definitions of the functions appearing in the above expression are given by

Convex combination with origin	$D(B) = \{\alpha\mathbf{h} : 0 \leq \alpha \leq 1, \mathbf{h} \in B\}$
Projection function	$Proj_{\mathbf{W}_S}(B) = \{\{h_{W_s}\}_{s \in S} : \mathbf{h} \in B\}$
Inferior set function	$\Omega(B) = \{\mathbf{h} : \boldsymbol{0} \leq \mathbf{h} \leq \mathbf{h}', \mathbf{h}' \in B\}$

8.3 Main Result

Theorem 8.3.1 (Tightness of the Yeung Outer Bound)**.** *The Yeung entropic region outer bound is tight if and only if the **AERS** holds. Namely, for any acyclic network coding instance $\mathcal{I}$ and rate vector $\mathbf{R}$,*

$$\Big(\mathbf{R} \in \mathcal{R}_{out}(\mathcal{I}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_\epsilon(\mathcal{I})\Big) \textit{ if and only if the } \boldsymbol{AERS} \textit{ holds.}$$

Proof. The tightness of the Yeung outer bound can be expressed in a form that is similar to capacity reduction, namely

$$\mathbf{R} \in \mathcal{R}_{out}(\mathcal{I}) \Leftrightarrow \mathbf{R} \in \mathcal{R}_{\epsilon}(\mathcal{I}). \tag{8.2}$$

We show that (8.2) holds if and only if the asymptotic edge removal statement holds. The proof of this result is given in Section 8.4. □

A full characterization of the capacity region $\mathcal{R}(\mathcal{I})$ appears in [16] and is equal to

$$\Omega(\mathrm{Proj}_{\mathbf{W}_S}(\overline{D(\Gamma^* \cap L_{123})} \cap L_{45}))).$$

Nevertheless, due to its relative simplicity, $\mathcal{R}_{out}$ has seen various studies. A variant of the outer bound $\mathcal{R}_{out}$ is shown to be equal to the ϵ-error capacity region of a network when dependence is allowed among the sources [52].

8.4 Proof of Theorem 8.3.1

Proof. This proof uses notation from Section 8.2. By Theorem 5.5.1, it suffices to show that $\mathbf{R} \in \mathcal{R}_{out}(\mathcal{I}) \Leftrightarrow \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_{\epsilon}(\mathcal{I}_{\lambda}^{bf})$. Here the mapping from $(\mathcal{I}, \mathbf{R})$ to $(\tilde{\mathcal{I}}, \tilde{\mathbf{R}})$ is the identity map (i.e., $\tilde{\mathcal{I}} = \mathcal{I}$ and $\tilde{\mathbf{R}} = \mathbf{R}$). To prove this "if and only if" statement, we present two proof directions,

$$\mathbf{R} \in \mathcal{R}_{out}(\mathcal{I}) \Rightarrow \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_{\epsilon}(\mathcal{I}_{\lambda}^{bf}) \text{ and } \mathbf{R} \in \mathcal{R}_{out}(\mathcal{I}) \Leftarrow \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_{\epsilon}(\mathcal{I}_{\lambda}^{bf}).$$

First direction: $\mathbf{R} \in \mathcal{R}_{out}(\mathcal{I}) \Rightarrow \mathbf{R} \in \lim_{\lambda \to 0} \mathcal{R}_{\epsilon}(\mathcal{I}_{\lambda}^{bf})$. Let $\mathbf{R}' \in \mathcal{R}_{out}(\mathcal{I})$, then there exists an entropic vector $\mathbf{h} \in \overline{D(\Gamma^*)} \cap L_{12345}$ and a rate vector $\mathbf{R}$ such that

$$\mathbf{R}' \leq \mathbf{R} = \mathrm{P}roj_{\mathbf{W}_S}(\mathbf{h}).$$

By Lemma 8.4.1, stated shortly, there exists a sequence of quasi-uniform random variables $\{(\mathbf{W}_S^{(m)}, \mathbf{X}_E^{(m)})\}$ with corresponding entropic vector $\mathbf{h}^{(m)} \in \Gamma_Q^*$, a sequence of integers $\{n_m\}$, and a sequence of positive numbers $\{\delta_m\}$ such that the following are satisfied:

$$\begin{array}{lcl}
L_2(\mathcal{I}) & : & \forall e \in E_S, h^{(m)}_{X_e|W_{\mathrm{In}(e)}} = 0, \\
L_3(\mathcal{I}) & : & \forall e \in E_{S^c}, h^{(m)}_{X_e|Z_{\mathrm{In}(e)}} = 0, \\
L_5(\mathcal{I}) & : & \forall t \in T, h^{(m)}_{W_t|Z_t} = 0, \\
\text{Dependence} & : & \sum_{s \in S} h^{(m)}_{W_s} - h^{(m)}_{\mathbf{W}} \leq n_m \delta_m, \\
\text{Edge capacity} & : & \forall e \in E, h^{(m)}_{X_e} \leq n_m(c_e + \delta_m), \\
\text{Rate} & : & \forall s \in S, h^{(m)}_{W_s} \geq n_m(R_s - \delta_m),
\end{array}$$

where $\lim_{m\to\infty} n_m = \infty$, $\lim_{m\to\infty} \delta_m = 0$, and $\lim_{m\to\infty} \frac{\mathbf{h}^{(m)}}{n_m} = \mathbf{h}$.

Since the sequence of variables are quasi-uniform, the edge capacity constraint implies that

$$h^{(m)}_{X_e} = \log |sp(X^{(m)}_e)|,$$

a code can be obtained by sending the indices of $sp(X^{(m)}_e)$ for each e, where each index is an element of $[\mathbb{F}_2^{c_e n_m + \delta_m n_m}]$ by our convention. For a block length of n_m, an edge of capacity c_e can only carry $2^{n_m c_e}$ messages, to account for the larger message size of $2^{c_e n_m + \delta_m n_m}$, we increase the block length by $\frac{n_m \delta_m}{c^*}$, where c^* is the smallest edge capacity in $\mathcal{I}$ (i.e., $h^{(m)}_{X_e} \leq n_m(c_e + \delta_m) \leq n_m(1 + \frac{\delta_m}{c^*})c_e$). For small enough δ_m,

$$\frac{1}{1 + \delta_m/c^*} > 1 - \delta_m/c^*,$$

$\mathcal{I}$ is therefore $(\mathbf{R}(1-\delta_m/c^*), 0, n_m(1+\frac{\delta_m}{c^*}), \delta_m)$-feasible. Since δ_m tends to zero, by Lemma 5.3.1(1), $\forall \rho, \lambda > 0$ there exists n such that $\mathcal{I}^{cf}_\lambda$ is $(\mathbf{R} - \rho, 0, n, 0)$-feasible.

Since $\mathcal{R}_0$ is closed and $\forall \lambda' < \lambda, \mathcal{R}_0(\mathcal{I}^{cf}_{\lambda'}) \subseteq \mathcal{R}_0(\mathcal{I}^{cf}_\lambda)$, we have $\mathbf{R} \in \mathcal{R}_0(\mathcal{I}^{cf}_\lambda), \forall \lambda > 0$. This implies that $\mathbf{R} \in \lim_{\lambda\to 0} \mathcal{R}_0(\mathcal{I}^{cf}_\lambda)$. Finally, since $\mathcal{R}_0(\mathcal{I}^{cf}_\lambda) \subseteq \mathcal{R}_0(\mathcal{I}^{bf}_\lambda)$ and $\mathcal{R}_0(\mathcal{I}^{bf}_\lambda) \subseteq \mathcal{R}_\epsilon(\mathcal{I}^{bf}_\lambda)$, we have $\mathbf{R} \in \lim_{\lambda\to 0} \mathcal{R}_\epsilon(\mathcal{I}^{bf}_\lambda)$.

Second direction: $\mathbf{R} \in \mathcal{R}_{out}(\mathcal{I}) \Leftarrow \mathbf{R} \in \lim_{\lambda\to 0} \mathcal{R}_\epsilon(\mathcal{I}^{bf}_\lambda)$ Let $\mathbf{R} \in \lim_{\lambda\to 0+} \mathcal{R}_\epsilon(\mathcal{I}^{bf}_\lambda)$.

For any $\lambda, \rho, \epsilon > 0$, there exists a blocklength n such that $\mathcal{I}^{bf}_\lambda$ is $(\mathbf{R} - \rho, \epsilon, n, 0)$-feasible. By Lemma 5.3.1(2), there exists a monotone sequences of positive numbers $\{\delta_m\}$ tending to 0 and positive integers $\{n_m\}$ tending to infinity such that for each m, there exists an $(\mathbf{R} - \delta_m, 0, n_m, \delta_m)$ code for hence $\mathcal{I}$. For each m, the code induces random variables $\mathcal{N} = \{W^{(m)}_s, s \in S; X^{(m)}_e, e \in E\}$ with entropic vector $\mathbf{h}^{(m)}$ satisfying:

$$\begin{array}{lll}
\text{Dependence} & : & \sum_{s\in S} h^{(m)}_{W_s} - h^{(m)}_{\mathbf{W}} \leq n_m\delta_m. \\
\text{Encoding 1} & : & \forall e \in E_S, h^{(m)}_{X_e|W_{\text{In}(e)}} = 0. \\
\text{Encoding 2} & : & \forall e \in E_{S^c}, h^{(m)}_{X_e|Z_{\text{In}(e)}} = 0. \\
\text{Decoding error} & : & \forall t \in T, h^{(m)}_{W_t|Z_t} = 0. \\
\text{Edge Capacity} & : & \forall e \in E, h^{(m)}_{X_e} \leq n_m c_e. \\
\text{Rate requirement} & : & \forall s \in S, h^{(m)}_{W_s} \geq n_m(R_s - \delta_m).
\end{array}$$

Let $\mathbf{h}^{(m)*} = n_m^{-1}\mathbf{h}^{(m)}$, then $\mathbf{h}^{(m)*} \in D(\Gamma^*)$. Since $\mathbf{h}^{(m)*}$ is bounded and $\delta_m \to 0$, there is a converging subsequence $\{\mathbf{h}^{(m_i)*}\}_{i=1}^{\infty}$ with limit such that $\mathbf{h} = \lim_{i\to\infty} \mathbf{h}^{(m_i)*} \in \overline{D(\Gamma^*)} \cap L_{12345} = \mathcal{R}_{\text{out}}(\mathcal{I})$. □

Lemma 8.4.1 ([17, Proposition 2]). *For any $\mathbf{h} \in \overline{\Gamma}_n^*$, there exists a sequence of quasi-uniform random variables $\{\mathcal{N}^{(m)}\}$ and corresponding $\mathbf{h}^{(m)} \in \Gamma^*$ and a sequence of positive integers n_m such that*

1) $\lim_{m\to\infty} n_m = \infty$,

2) $\lim_{m\to\infty} \mathbf{h}^{(m)}/n_m = \mathbf{h}$,

3) For all positive integers m, $h_{\beta|\alpha}^{(m)} = 0$ for all $\beta \subset \mathcal{N}^{(m)}$ and $\alpha \subseteq \mathcal{N}^{(m)}$ such that $h_{\beta|\alpha} = 0$.

Chapter 9

ZERO-ERROR VERSUS EPSILON-ERROR CAPACITY REGION

The materials in this chapter are published in part as [50].

In this chapter, we study the potential gap between the zero-error and the epsilon-error network coding capacity region. For the single multicast scenario, it is known that the cut-set bound can be achieved using zero-error linear codes [36], [53]. Thus, these two notions of capacity region are equal in this setting. In fact, due to the structured properties of linear functions, it can be shown that the zero-error network coding capacity region equals the epsilon-error capacity region under the restriction of linear codes. We give the formal proof of this observation in Section 9.1.

Since linear codes are insufficient to achieve the general capacity region, this observation does not resolve the question of whether there is a gap between the zero-error and epsilon-error capacity region for general codes. This potential gap remains an intriguing open question. In [19], the authors show that this question is closely related to a variation of the edge removal statement.

Motivated by the work of [16] which presents an implicit characterization of the epsilon-error capacity region using notions of entropic vectors, we follow a similar approach and derive an implicit characterization of the zero-error capacity region for acyclic network coding networks. Our characterization is based on a dense subset of the entropic region, which is defined in Section 8.1. This result is given in Section 9.2.

9.1 Zero Versus Epsilon-Error Linear Network Coding Capacity Region

In the theorem below, we show that zero-error linear codes achieve the epsilon-error linear capacity region of network coding networks.

Theorem 9.1.1. *For any network coding network $\mathcal{I} = (G, S, T)$, $\mathcal{R}^L_\epsilon(\mathcal{I}) = \mathcal{R}^L_0(\mathcal{I})$.*

Proof. Fix any $0 < \epsilon < 1/2$, we start with an $(\mathbf{R}, \epsilon, n)$-linearly-feasible code $\mathcal{C}$

for $\mathcal{I}$. We show that replacing the decoder of $\mathcal{C}$ with a maximum likelihood decoder will yield an $(\mathbf{R}, 0, n)$-linearly-feasible code for $\mathcal{I}$, thus proving our assertion.

We denote by W_t the source message variable demanded by t and by $W_{\bar{t}}$ the remaining source message variables. Then for any terminal $t \in T$, the received information variable Z_t (under the operation of $\mathcal{C}$) can be written as a function of the sources $\mathbf{W} = (W_t, W_{\bar{t}})$. Thus,

$$Z_t = \mathbf{W}F = W_tF_t + W_{\bar{t}}F_{\bar{t}},$$

where Z_t and $\mathbf{W} = (W_t, W_{\bar{t}})$ are row vectors and $F = \begin{bmatrix} F_t \\ F_{\bar{t}} \end{bmatrix}$ are matrices over $\mathbb{F}_2$.

Denote by $\mathcal{N}_t$ and $\mathcal{N}_{\bar{t}}$ the left null space of F_t and $F_{\bar{t}}$, respectively (i.e., $\mathcal{N}_t = \{w_t \in \mathcal{W}_t | w_tF_t = \mathbf{0}\}$.) For any $v \in \mathcal{V}_t, w_t \in \mathcal{W}_t$, we have

$$\Pr(W_t = w_t | W_tF_t = v) = \frac{1}{2^{\text{rank}(\mathcal{N}_t)}}.$$

If $\text{rank}(\mathcal{N}_t) > 0$, the probability of error for any terminal receiving W_tF_t will be at least $1 - 2^{-\text{rank}(\mathcal{N}_t)}$. Since

$$W_t \to W_tF_t \to Z_t$$

forms a Markov chain, a decoder receiving W_tF_t will perform no worse than one that receives Z_t if a maximum likelihood decoder is used in both scenarios. Therefore F_t must be full rank.

With the assumption that $\text{rank}(\mathcal{N}_t) = 0$, denote by $\mathcal{V}_t$, $\mathcal{V}_{\bar{t}}$ the row spaces of F_t and $F_{\bar{t}}$, respectively. Denote by $\mathcal{V}_{t \cap \bar{t}}$ the intersection of $\mathcal{V}_t$ and $\mathcal{V}_{\bar{t}}$. Let $\mathcal{P}_t \subset \mathcal{W}_t$ denote the pre-image of $\mathcal{V}_{t \cap \bar{t}}$ with respect to F_t, (i.e., $\mathcal{P}_t = \{w_t \in \mathcal{W}_t | w_tF_t \in \mathcal{V}_{\bar{t}}\}$). Then for any $v_t \in \mathcal{P}_t$, the set

$$\mathcal{S}(v_t) = \{(v_t, w_{\bar{t}}) \in \mathcal{W}_t \times \mathcal{W}_{\bar{t}} | v_tF_t + w_{\bar{t}}F_{\bar{t}} = \mathbf{0}\}$$

has cardinality equal to $2^{\text{rank}(\mathcal{N}_{\bar{t}})}$. Let $\mathcal{N}$ and $\mathcal{N}_{\bar{t}}$ denote the left null space of F and $F_{\bar{t}}$, respectively. Thus, for any $w_t \in \mathcal{W}_t$, $w_{\bar{t}} \in \mathcal{W}_{\bar{t}}$ and $v_t \in \mathcal{P}_t$,

$$\begin{aligned}
&\Pr(W_t = w_t + v_t | Y_t = w_tF_t + w_{\bar{t}}F_{\bar{t}}) \\
&\quad = \frac{|\{(w_t + v_t, w'_{\bar{t}}) \in \mathcal{W}_t \times \mathcal{W}_{\bar{t}} | (w_t + v_t)F_t + w'_{\bar{t}}F_{\bar{t}} = w_tF_t + w_{\bar{t}}F_{\bar{t}}\}|}{|\{(w'_t, w'_{\bar{t}}) \in \mathcal{W}_t \times \mathcal{W}_{\bar{t}} | w'_tF_t + w'_{\bar{t}}F_{\bar{t}} = w_tF_t + w_{\bar{t}}F_{\bar{t}}\}|} \\
&\quad = \frac{|\mathcal{S}(v_t)|}{2^{\text{rank}(\mathcal{N})}} = \frac{2^{\text{rank}(\mathcal{N}_{\bar{t}})}}{2^{\text{rank}(\mathcal{N})}}.
\end{aligned}$$

The decoding error of a maximum likelihood decoder at t then equals $1 - 2^{\text{rank}(\mathcal{N}_{\bar{t}})-\text{rank}(\mathcal{N})}$. For any $\epsilon < 1/2$, we therefore require $\text{rank}(\mathcal{N}) = \text{rank}(\mathcal{N}_{\bar{t}})$, giving

$$\Pr(W_t = w_t | Y_t = w_t F_t + w_{\bar{t}} F_{\bar{t}}) = 1,$$

which implies that using the encoders in $\mathcal{C}$ and a maximum likelihood decoder at each of the terminals yields an $(\mathbf{R}, 0, n)$-linearly-feasible code for $\mathcal{I}$. □

9.2 An Implicit Characterization for the Zero-Error Capacity Region

In [20], bounds for the 0-error capacity region are derived by relaxing the edge capacity constraint from a strict (worse case) requirement to an average requirement. In this section, we provide an exact (implicit) characterization for the 0-error network coding capacity region using a dense subset of the entropic region under a strict edge capacity constraint.

We derive this result in a form that is similar to a capacity reduction. More precisely, we show that for any acyclic network coding instance $\mathcal{I}$,

$$\tilde{\mathcal{R}}(\tilde{\mathcal{I}}) = \mathcal{R}_0(\mathcal{I}),$$

where we will describe $\tilde{\mathcal{R}}(\tilde{\mathcal{I}})$ and the mapping from $\mathcal{I}$ to $\tilde{\mathcal{I}}$ shortly. We rely on definitions of the individually uniform entropic vectors from Section 8.1.

We first describe mapping Φ_4 that maps any acyclic network coding instance $\mathcal{I}$ to a corresponding network coding instance $\tilde{\mathcal{I}}$. Given a network $\mathcal{I} = (G, S, T)$, network $\tilde{\mathcal{I}} = (\tilde{G}, \tilde{S}, \tilde{T})$ is obtained from $\mathcal{I}$, described below.

Graph $\tilde{G} = (\tilde{V}, \tilde{E}, \tilde{C})$ is obtained from G by adding a new source node s' and, for each edge $e = (u, v) \in E_{\overline{S}}$, a pair of new edges $(s', u), (s', v)$ with infinite capacity. Thus,

$$\begin{aligned}
\tilde{S} &= S \cup \{s'\} \\
\tilde{V} &= V \cup \{s'\} \\
\tilde{E} &= E \cup \{(s', v), v \in \overline{S}\} \\
\tilde{C}_e &= \begin{cases} c_e & \text{if } e \in E_{\overline{S}} \\ \infty & \text{otherwise.} \end{cases}
\end{aligned}$$

Source s' provides a common randomness to all nodes in the network and is not demanded by any node. The demands for the rest of the network and the set of terminal nodes remain the same, and thus $\tilde{T} = T$.

Theorem 9.2.1 gives a characterization of the zero error network coding capacity region. The linear subspaces L_i are based on the definitions given for the Yeung outer bound in Section 8.2.

Theorem 9.2.1. *For any acyclic network coding instances $\mathcal{I}$,*

$$\tilde{\mathcal{R}}(\tilde{\mathcal{I}}) = \mathcal{R}_0(\mathcal{I}),$$

where $\tilde{\mathcal{R}}(\tilde{\mathcal{I}}) = \Omega(Proj_{\mathbf{W}_{\tilde{S}\setminus\{s'\}}}(\overline{D(\Gamma_U^* \cap L_{12345})}))$ and the mapping from $\mathcal{I}$ to $\tilde{\mathcal{I}}$ is given by Φ_4.

Proof. In Theorem 9.2.1, we wish to show that the existence of an entropic vector for $\mathcal{N}(\mathcal{I})$ implies the existence of a 0-error code for $\mathcal{I}$ of the same rate (achievability proof). Compared to the methodology of [16], new argument is required here since the random coding argument in [16] does not necessarily imply a 0-error code. Building on the idea of constructing 0-error codes from quasi-uniform random variables [17], we rely on Γ_U^*, which is a relaxation of the quasi-uniform entropic region (Γ_Q^*) with similar desirable properties. Any random vector $\mathcal{N}$ with an entropic vector that falls in Γ_U^* and satisfies functional constraints L_{12345} can be used to construct a 0-error code. Specifically, a code can be constructed by sending the index of X_e in the support set of X_e ($\mathrm{s}p(X_e)$), since the size of the support satisfies the edge capacity constraint $\log|\mathrm{s}p(X_e)| \leq 2^{c_e n+\gamma n}$.

The use of Γ_U^* in the achievability result creates a problem in the converse. Any entropy vector in $\Gamma_U^* \cap L_{12345}$ corresponds to a zero-error code, but not all zero-error code corresponds to a random vector $\mathcal{N}$ with an entropic vector in Γ_U^*. We overcome this challenge by augmenting $\mathcal{I}$ to form $\tilde{\mathcal{I}}$, which allows us to convert any code in $\mathcal{I}$ to one whose entropic vector is in Γ_U^*. We follow techniques from [16].

Achievability ($\tilde{\mathcal{R}}(\tilde{\mathcal{I}}) \subseteq \mathcal{R}_0(\mathcal{I})$): Let $\mathbf{R}' \in \tilde{\mathcal{R}}(\tilde{\mathcal{I}})$. Then there exists a vector

$$\mathbf{h} \in \overline{D(\Gamma_U^* \cap L_{12345})}$$

such that

$$\mathbf{R}' \leq \mathbf{R} = \mathrm{Proj}_{\mathbf{W}_{\tilde{S}\setminus\{s'\}}}(\mathbf{h}).$$

There exists a sequence of entropic vectors $\{\alpha_m \mathbf{h}^{(m)}\}$, a sequence of coefficients $\{\alpha_m\}, \alpha_m \in [0,1]$, a sequence of blocklengths $\{n_m = \lceil\frac{1}{\alpha_m}\rceil\}$, and a sequence of slackness parameters $\{\epsilon_m\}$ such that

$$\begin{array}{lcl}
L_1(\tilde{\mathcal{I}}) & : & \sum_{s\in\tilde{S}} h^{(m)}_{\tilde{W}_s} - h^{(m)}_{\tilde{\mathbf{W}}_{\tilde{S}}} = 0, \\
L_2(\tilde{\mathcal{I}}) & : & \forall e \in \tilde{E}, \mathrm{In}(e) \in \tilde{S}, h^{(m)}_{\tilde{X}_e|\tilde{W}_{\mathrm{In}(e)}} = 0, \\
L_3(\tilde{\mathcal{I}}) & : & \forall e \in \tilde{E}, \mathrm{In}(e) \notin \tilde{S}, h^{(m)}_{\tilde{X}_e|\tilde{Z}_{\mathrm{In}(e)}} = 0, \\
L_4(\tilde{\mathcal{I}}) & : & \forall e \in \tilde{E}, h^{(m)}_{\tilde{X}_e} \leq n_m c_e, \\
L_5(\tilde{\mathcal{I}}) & : & \forall t \in \tilde{T}, h^{(m)}_{\tilde{W}_t|\tilde{Z}_t} = 0, \\
\text{Rates} & : & \forall s \in \tilde{S} \setminus \{s'\}, h^{(m)}_{\tilde{W}_s} \geq n_m(R_s - \epsilon_m),
\end{array}$$

where $\mathbf{h}^{(m)} \in \Gamma^*_U \cap L_{12345}$, $\lim_{m\to\infty} \alpha_m \mathbf{h}_m = \mathbf{h}$ and $\lim_{m\to\infty} \epsilon_m = 0$.

Since edge variables are uniform over their supports (i.e., $h^{(m)}_{\tilde{X}_e} = \log|sp(\mathcal{X}_e)|$) for each e, we obtain a sequence of codes for $\tilde{\mathcal{I}}$ that are $(\mathbf{R}-\epsilon_m, 0, n_m)$-feasible by sending the indices of the alphabets of X_e over each edge.

Each of these codes for $\tilde{\mathcal{I}}$ can be further converted to a code for $\mathcal{I}$ with the same rate by fixing a source realization $\tilde{W}_{s'} = \tilde{w}_{s'}$. Now we apply the same encoding and decoding functions to $\mathcal{I}$ by assuming that $\tilde{W}_{s'} = \tilde{w}_{s'}$. Every node in $\mathcal{I}$ may therefore compute the message being sent on the outgoing links of node s' in $\tilde{\mathcal{I}}$ even though these links are not available in $\mathcal{I}$. Thus, we get a sequence of codes for $\mathcal{I}$ that is $(\mathbf{R} - \epsilon_m, 0, n_m)$-feasible. This implies that $\mathbf{R} \in \mathcal{R}_0(\mathcal{I})$.

Converse ($\tilde{\mathcal{R}}(\tilde{\mathcal{I}}) \supseteq \mathcal{R}_0(\mathcal{I})$): Let $\mathbf{R} \in \mathcal{R}_0(\mathcal{I})$, there exists a sequence of codes $\{\mathcal{C}_m\}$ so that for each m, $\mathcal{C}_m$ is $(\mathbf{R} - \epsilon_m, 0, n_m)$-feasible code for $\mathcal{I}$. Let X_e be the edge message sent in code $\mathcal{C}_m$ for each $e \in E$. Next we convert each $\mathcal{C}_m$ into a code $\tilde{\mathcal{C}}_m$ for $\tilde{\mathcal{I}}$ in which each individual edge message $\tilde{X}_e$ in $\tilde{\mathcal{I}}$ is uniformly distributed over its support. Note that the alphabet of each edge e is the set $\mathbb{F}_2^{c_e n_m}$.

For each edge $e = (u, v) \in E_{\overline{S}}$, the new code $\tilde{\mathcal{C}}_m$ sends a random variable M_e that is uniformly distributed in $\mathbb{F}_2^{c_e n_m}$ via edges (s', u) and (s', v) to both node u and node v. For the rest of the edges $(u, v) \in E_{\overline{S}}$, $\tilde{\mathcal{C}}_m$ sends a uniformly distributed message $\tilde{X}_e = X_e \oplus M_e$ on each edge e.

The decoding function of $\tilde{\mathcal{C}}_m$ for each $t \in \tilde{T}$ first subtracts M_e from $\tilde{X}_e$ for each e such that $\mathrm{Out}(e) = t$, and then applies the same decoding function from $\mathcal{C}_m$.

Each edge message $\tilde{X}_e$ is therefore uniformly distributed on the set $\mathbb{F}_2^{c_e n_m}$. Each $\tilde{\mathcal{C}}_m$ therefore induces a set of entropic vectors $\mathbf{h}^{(m)} \in \Gamma^*_U$ such that

$$\begin{array}{rcl}
L_1(\tilde{\mathcal{I}}) & : & \sum_{s\in\tilde{S}} h^{(m)}_{\tilde{W}_s} - h^{(m)}_{\tilde{W}_{\tilde{S}}} = 0, \\
L_2(\tilde{\mathcal{I}}) & : & \forall e \in \tilde{E}, \mathrm{In}(e) \in \tilde{S}, h^{(m)}_{\tilde{X}_e|\tilde{W}_{\mathrm{In}(e)}} = 0, \\
L_3(\tilde{\mathcal{I}}) & : & \forall e \in \tilde{E}, \mathrm{In}(e) \notin \tilde{S}, h^{(m)}_{\tilde{X}_e|\tilde{Z}_{\mathrm{In}(e)}} = 0, \\
L_4(\tilde{\mathcal{I}}) & : & \forall e \in \tilde{E}, h^{(m)}_{\tilde{X}_e} \leq n_m c_e, \\
L_5(\tilde{\mathcal{I}}) & : & \forall t \in \tilde{T}, h^{(m)}_{\tilde{W}_t|\tilde{Z}_t} = 0, \\
\text{Rates} & : & \forall s \in S \setminus \{s'\}, h^{(m)}_{\tilde{W}_s} \geq n_m(R_s - \epsilon_m),
\end{array}$$

and $\lim_{m\to\infty} \epsilon_m = 0$. Therefore, $\forall m$, $\mathbf{h}^{(m)} \in \Gamma^*_U \cap L_{12345}$ and $n_m^{-1}\mathbf{h}^{(m)} \in D(\Gamma^*_U \cap L_{12345})$. Since $\{n_m^{-1}\mathbf{h}^{(m)}\}$ is a bounded sequence, by the Bolzano-Weierstrass theorem, there exists a convergent sub-sequence $\{n_m^{-1}\mathbf{h}^{(m_i)}\}_{i=1}^{\infty}$ converging to $\mathbf{h} \in \overline{D(\Gamma^*_U \cap L_{12345})}$. □

Chapter 10

CODE REDUCTION FROM STREAMING NETWORK CODING TO STREAMING INDEX CODING

The materials of this chapter are published in part as [54].

While the network coding problem [35], [53] has been studied extensively in recent years, it remains a challenging problem even when restricted to acyclic networks. In this chapter, we consider a streaming network coding model [10], [11], where each demand has to satisfy both rate and streaming delay constraints. Motivated by the goal of extending tools and results derived for acyclic networks to the more practically relevant domain of cyclic networks, and inspired by [6], which demonstrates a code equivalence between network coding and index coding problems, we here derive a code equivalence between general network codes and index codes under the streaming model.

The reduction of [6] and its extensions [7], [49] are proven for acyclic network problems and do not easily extend to networks containing cycles. A first step towards addressing this challenge is to understand scenarios where this reduction can be extended. Restricting our attention to the streaming case enables us to overcome the key challenges for the cyclic case and apply techniques similar to those in [6] to prove a code equivalence between streaming network coding and streaming index coding problems for both acyclic and cyclic networks. A consequence of this reduction is that under the streaming model, there is no fundamental difference between acyclic and cyclic networks. We can therefore restrict our attention to index coding problems when working with the streaming variant of the network coding problem.

10.1 Streaming Network Coding Model

The instance description for streaming network coding is identical to that of the classical network coding from Section 2.2. Recall that an instance is given by $\mathcal{I} = (G, S, T)$, where sets $S, T \subset V$ are the source nodes and terminal nodes, respectively. We assume a canonical model where there are k sources $S = \{s_1, \cdots, s_k\}$ and kl terminal nodes $T = \{t_{1,1}, \cdots, t_{1,l}, \cdots, t_{k,l}\}$ and each source s_i is demanded by exactly l terminal nodes $\{t_{i,1}, \cdots, t_{i,l}\}$. The graph

$G = (V, E, C)$ is defined by a set of vertices V representing communicating devices, a set of directed edges $E \subseteq V^2$ representing communication channels between these devices, and a vector $C = (c_e : e \in E)$ describing the maximal rate of communication across each edge.

Each non-source edge $e \in E_{S^c} = \{e' \in E : \text{In}(e') \in V \setminus S\}$ is a noiseless channel of integer capacity c_e from the edge's input node, here called $\text{In}(e)$, to its output node, here called $\text{Out}(e)$; for example, if $e = (u, v)$ and $C_e = 1$, then information travels from node $\text{In}(e) = u$ to node $\text{Out}(e) = v$ at a rate of $C_e = 1$ bit per transmission. Each edge $e \in E_S = \{e' \in E : \text{In}(e') \in S\}$ has infinite capacity. Again, these *source* edges are used to capture the notion that source message variables from $\text{In}(e)$ are *streaming* to node $\text{Out}(e)$. For each time step τ, each node $v \in V \setminus (S \cup T)$ receives a c_e–bit edge information variable $X_{e,\tau} \in \mathbb{F}_2^{c_e}$ for each edge $e \in E$ such that $\text{Out}(e) = v$. Denote by $X_{e,\tau}$ the information variable on each edge e at time τ. For a rate vector $\mathbf{R} = (R_1, \cdots, R_k)$, each source node $s \in S$ holds a stream of source message random variables $[W_{s,\tau}]_{\tau \in \mathbb{N}}$ where each $W_{s,\tau}$ is uniformly distributed in $\mathcal{W}_s = \mathbb{F}_2^{R_s}$. We assume each R_s is an integer. Given a source delay requirement function

$$d_S : S \times (V \setminus S) \to \mathbb{Z}_{\geq 0},$$

each node $v \in V \setminus (S \cup T)$ receives an R_s–bit time-delayed source message variable

$$W_{s,\tau - d_S(s,v)} \in \mathbb{F}_2^{R_s}$$

for each time step τ and for each $s \in S$ such that $(s, v) \in E$.

For $v \in V \setminus S$, denote by $S_G(v) = \{s \in S : (s, v) \in E\}$ and $E_G(v) = \{e' \in E_{S^c} : \text{Out}(e') = v\}$ the set of sources and non-source edges entering v. For a set A, denote by $\mathbf{X}_A = (X_a, a \in A)$ the vector of variables of X with subscript in A. The edge information and source message variables received by v at time τ is therefore given by $X_{E_G(v),\tau} = (X_{e',\tau}, e' \in E_G(v))$. and $W_{S_G(v),\tau} = (W_{s,\tau - d_S(s,v)} : s \in S_G(v))$, respectively.

For time indices τ_1 and τ_2, denote by $[X]_{\tau_1}^{\tau_2} = (X_{\tau_1}, \cdots, X_{\tau_2})$ the vector of variables of X_τ from time index τ_1 to τ_2. Given rate vector $\mathbf{R} = (R_1, \cdots, R_k)$, a source delay requirement function d_S, a decoding delay requirement function

$$d_T : T \to \mathbb{Z}_{\geq 0},$$

and a memory parameter q, a streaming network code $(\mathcal{F}, \mathcal{G}, \mathcal{K})$ for $\mathcal{I}$ includes a set of encoding functions $\mathcal{F} = \{f_e\}_{e \in E_{S^c}}$, a set of decoding functions $\mathcal{G} = \{g_t\}_{t \in T}$ and a set of constants $\mathcal{K} = \{[\mathbf{x}_E]^0_{-d_{\max}-q}, ([\mathbf{w}_S]^0_{-d_{\max}-q})\}$ for initialization, where $d_{\max}$ denotes the maximum delay requirement in d_S and d_T.

We assume that there is a single unit time delay at each node. The network code therefore operates under a strict causality constraint where each encoding and decoding function can only be a function of the past q received message. At each time step τ, each edge (or terminal) applies a single function f_e (or d_t) over a "sliding window" of size q (explained in detail below). Input variables in the first q sliding windows are initialized by $\mathcal{K}$ at each time step τ.

The network code is said to satisfy decoding delay requirements d_T if the following are true:

- For each $e \in E_{S^c}$ and each time index $\tau \in \mathbb{N}$,

$$f_e : \mathbb{F}_2^{c_e q} \times \left(\prod_{e' \in E_G(\mathrm{In}(e))} \mathbb{F}_2^{c_{e'} q} \right) \times \left(\prod_{s' \in S_G(\mathrm{In}(e))} \mathbb{F}_2^{R_{s'} q} \right) \to \mathbb{F}_2^{c_e}$$

  is a sliding window encoding function that takes as input the past q source message variables $[W_{S_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q}$, the past q edge variables $[X_{E_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q})$ received by $\mathrm{In}(e)$ and the past q edge variables $[X_e]^{\tau-1}_{\tau-q}$ transmitted across edge e and transmits as output the information variable

$$X_{e,\tau} = f_e([X_e]^{\tau-1}_{\tau-q}, [X_{E_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q}, [W_{S_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q})$$

  at time τ on edge e.

- For each demand pair (s, t), let $d_T(s, t)$ be the corresponding delay constraint. For each time index $\tau > d_T(t)$,

$$g_t : \mathcal{W}_s^q \times \left(\prod_{e' \in E_G(t)} \mathbb{F}_2^{c_{e'} q} \right) \times \left(\prod_{s' \in S_G(t)} \mathbb{F}_2^{R_{s'} q} \right) \times \to \mathcal{W}_s$$

  is a sliding window decoding function that takes as input the past q source message variables $[W_{S_G(t)}]^{\tau-1}_{\tau-q}$, the past q edge variables $[X_{E_G(t)}]^{\tau-1}_{\tau-q}$ received by terminal t and the past q source variables $[W_s]^{\tau-d_T(t)-1}_{\tau-d_T(t)-q}$ decoded at node t and outputs the source message variable

$$W_{s,\tau-d_T(t)} = g_t([W_s]^{\tau-d_T(t)-1}_{\tau-d_T(t)-q}, [X_{E_G(t)}]^{\tau-1}_{\tau-q}, [W_{S_G(t)}]^{\tau-1}_{\tau-q}), \tag{10.1}$$

Streaming Index Coding Model

A streaming index coding network is a streaming network coding network $\mathcal{I}$ with graph G falling in a restricted topology as described in Section 2.3. A k by l multiple multicast streaming index coding instance is a streaming network coding instance with $S = \{s_1, \cdots, s_k\}$, $T = \{t_{1,1}, \cdots, t_{1,l}, \cdots, t_{k,l}\}$ and $V = \{u_1, u_2\} \cup S \cup T$. Node u_1 is the broadcast node and has access to all the source node. Node u_2 is the relay node and has no connection to any of the sources. The source nodes connected to a given terminal node $t \in T$ are described by the "has" set H_t of terminal t, giving

$$\begin{aligned} E &= \left[\bigcup_{s\in S}\{(s,u_1)\}\right] \cup \{(u_1,u_2)\} \cup \left[\bigcup_{t\in T}\{(u_2,t)\}\right] \cup \left[\bigcup_{t\in T}\bigcup_{s\in H_t}\{(s,t)\}\right] \\ c_e &= \begin{cases} c_B & \text{if } \mathrm{In}(e) \in \{u_1,u_2\} \\ \infty & \text{otherwise.} \end{cases} \end{aligned}$$

We therefore alternatively describe instance $\mathcal{I}$ as

$$\mathcal{I} = (S, T, H = \{H_t, t \in T\}, c_B).$$

We assume without loss of generality that any edge with sufficient capacity to carry all information available to its input node at a certain time step carries that information unchanged; thus

$$X_{e,\tau} = f_e([X_{E_G(\mathrm{In}(e))}]_{\tau-q}^{\tau-1}, [X_e]_{\tau-q}^{\tau-1}) = X_{E_G(\mathrm{In}(e)),\tau-1} \text{ for all } e \in E_{S^c} \setminus \{(u_1,u_2)\}.$$

As a result, specifying an index code's encoder requires specifying only the encoder f_B for its bottleneck link.

Streaming Code Feasibility

We define a *zero-error* streaming code here. A network $\mathcal{I}$ is $(\mathbf{R}, d_S, d_T)$–streaming feasible if there exists a streaming network code $(\mathcal{F}, \mathcal{G}, \mathcal{K})$ with some memory parameter $q \in \mathbb{N}$ such that when $(\mathcal{F}, \mathcal{G}, \mathcal{K})$ is operated on $\mathcal{I}$ under a set of streaming sources satisfying the rate requirements $\mathbf{R}$ and streaming delay requirements d_S, each of the terminals is able to output the source messages, satisfying the delay requirements d_T with error probability 0.

10.2 Reduction Mapping Φ_5

We begin by describing our mapping Φ_5 Given a streaming network coding instance $\mathcal{I} = (G, S, T)$, the construction of $\tilde{\mathcal{I}} = (\tilde{S}, \tilde{T}, \tilde{H}, \tilde{c}_B)$ is given by mapping Φ_3 of Section 7.1. We describe the code parameter $(\tilde{\mathbf{R}}, \tilde{d}_S, \tilde{d}_T)$ that corresponds to $(\mathbf{R}, d_S, d_T)$.

The rate for $\tilde{\mathcal{I}}$ is $\tilde{\mathbf{R}} = (\mathbf{R}, (c_e)_{e \in E_{S^c}})$, where each $\tilde{w}_{s,\tau}$ is set to R_s, while the rate for each $\tilde{w}_{e,\tau}$ for each $e \in E_{S^c}$ is set to c_e.

The source delay requirement $\tilde{d}_{\tilde{S}}$ is defined as follows:

$$\tilde{d}_{\tilde{S}}(s, v) = \begin{cases} 0 & \text{if } v = u_1 \\ d_S(s,t) + 3 & \text{if } v = \tilde{t}_t \text{ for some } t \in T \\ 3 & \text{otherwise.} \end{cases}$$

The decoding delay requirement for each $\tilde{t}_t$ is set to $d_t + 3$ and the rest is set to 3, thus

$$\tilde{d}_{\tilde{T}}(\tilde{t}) = \begin{cases} d_T(t) + 3 & \text{if } v = \tilde{t}_t \text{ for some } t \in T \\ 3 & \text{otherwise.} \end{cases}$$

Main Result

Theorem 10.2.1. *Streaming code design for network coding reduces to streaming code design for index coding. That is, under mapping Φ_5, for any network coding instance $\mathcal{I}$ and code parameter $(\mathbf{R}, d_S, d_T)$,*

1. *$\mathcal{I}$ is $(\mathbf{R}, d_S, d_T)$ streaming-feasible if and only if $\tilde{\mathcal{I}}$ is $(\tilde{\mathbf{R}}, \tilde{d}_{\tilde{S}}, \tilde{d}_{\tilde{T}})$ streaming-feasible.*
2. *Any $(\tilde{\mathbf{R}}, \tilde{d}_{\tilde{S}}, \tilde{d}_{\tilde{T}})$ solution for $\tilde{\mathcal{I}}$ may be efficiently mapped to an $(\mathbf{R}, d_S, d_T)$ solution for $\mathcal{I}$.*

Proof. Proof of 1) is given in Section 10.3. The proof is 2) is implied by the proof of 1). □

This result shows that under the streaming model, the network coding and index coding problems are equivalent. The result holds even when the network coding instance contains cycles. This implies that under the streaming model, the complexity of a general network coding instance is completely captured by an index coding network, which is acyclic and has a much simpler topology. While the question of whether networks containing cycles can be reduced to acyclic networks under the classical setting without delays remains open, this result allows us to better understand the gap between these two cases.

10.3 Proof of Theorem 10.2.1

Proof. Consider a streaming network coding instance $\mathcal{I} = (G, S, T)$ and an $(\mathbf{R}, d_S, d_T)$-streaming code. Let $\tilde{\mathcal{I}} = (\tilde{S}, \tilde{T}, \tilde{H}, \tilde{c}_B)$ and the corresponding feasibility parameters $(\tilde{\mathbf{R}}, \tilde{d}_{\tilde{S}}, \tilde{d}_{\tilde{T}})$ be as described above. There are two directions to be proven.

Direction 1: If $\mathcal{I}$ is $(\mathbf{R}, d_S, d_T)$ streaming feasible then $\tilde{\mathcal{I}}$ is $(\tilde{\mathbf{R}}, \tilde{d}_{\tilde{S}}, \tilde{d}_{\tilde{T}})$ streaming feasible.

Consider an $(\mathbf{R}, d_S, d_T)$ streaming code $(\mathcal{F}, \mathcal{G}, \mathcal{K})$ for $\mathcal{I}$. Let the memory parameter for this code be q, the encoding functions be $\mathcal{F} = \{f_e\}_{e \in E_{S^c}}$, the decoding functions be $\mathcal{G} = \{g_t\}_{t \in T}$ and the initialization variables be

$$\mathcal{K} = \{[\mathbf{x}_{E_{S^c}}]^{0}_{-d_{\max}-q}, [\tilde{\mathbf{w}}_S]^{0}_{-d_{\max}-q}\}.$$

We first define a set of initialization constants $\tilde{\mathcal{K}} = \{[\tilde{x}_B]^{0}_{-\tilde{d}_{\max}-q}, [\tilde{\mathbf{w}}_{\tilde{S}}]^{0}_{-\tilde{d}_{\max}-q}\}$. These constants will ensure suitable inputs are provided to the encoding functions for the initial time steps. We break down each $\tilde{X}_{B,\tau}$ into $|E_{S^c}|$ components $(\tilde{X}_{B(e),\tau})_{e \in E_{S^c}}$. Let $[\tilde{\mathbf{w}}_S]_1^{\tilde{d}_{\max}+q}$ be any source realization for $[\tilde{\mathbf{W}}_S]_1^{\tilde{d}_{\max}+q}$. Let $[\mathbf{x}_{E_{S^c}}]_1^{\tilde{d}_{\max}+q}$ be the corresponding edge information realization for $\mathcal{I}$ when the code is operated on the sources $[\tilde{\mathbf{w}}_S]_1^{\tilde{d}_{\max}+q}$ and the initialization constants $\mathcal{K} = \{[\mathbf{x}_{E_{S^c}}]^{0}_{-d_{\max}-q}, [\tilde{\mathbf{w}}_S]^{0}_{-d_{\max}-q}\}$ for $\mathcal{I}$. Assign the initialization constants for $\tilde{\mathcal{K}}$ as follows.

$$\begin{aligned} [\tilde{X}_{B(e)}]^{0}_{-\tilde{d}_{\max}-q} &= [x_e]_0^{\tilde{d}_{\max}+q} \qquad \forall\, e \in E_{S^c}, \\ [\tilde{\mathbf{w}}_S]^{0}_{-\tilde{d}_{\max}-q} &= [\tilde{\mathbf{w}}_S]_0^{\tilde{d}_{\max}+q}, \\ [\tilde{\mathbf{w}}_{E_{S^c}}]^{0}_{-\tilde{d}_{\max}-q} &= \mathbf{0}. \end{aligned}$$

For $v \in V \setminus S$, define $\tilde{w}_{S_d(v),\tau} = (\tilde{w}_{s,\tau-d_S(s,v)}, s \in S_G(v))$. Next, we define a single sliding-window broadcast function $\tilde{f}_B$ for $\tilde{\mathcal{I}}$ with memory parameter $q+1$ in terms of the edge encoding functions $\mathcal{F} = \{f_e\}_{e \in E_{S^c}}$ for $\mathcal{I}$.

$$\begin{aligned} \tilde{X}_{B,\tau} &= \tilde{f}_B([\tilde{w}_{E_{S^c}}]^{\tau-1}_{\tau-q-1}, [\tilde{w}_S]^{\tau-1}_{\tau-q-1}, [\tilde{X}_B]^{\tau-1}_{\tau-q-1}) && (10.2) \\ &= (\tilde{X}_{e,\tau-1} \oplus \tilde{w}_{e,\tau-1})_{e \in E_{S^c}}, \\ \tilde{X}_{e,\tau} &= f_e([\tilde{X}_e]^{\tau-1}_{\tau-q}, [\tilde{X}_{E_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q}, [(\tilde{w}_{S_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q}). && (10.3) \end{aligned}$$

Finally, we show that each of the terminals can decode its request using the broadcast function defined above; recall that $\tilde{d}_{\tilde{T}}(\tilde{t}_e) = 2$, $\tilde{d}_{\tilde{T}}(\tilde{t}_t) = d_T(t) + 2$ and $\tilde{d}_{\tilde{S}}(\tilde{s}_s, \tilde{t}_t) = d_S(s,t) + 2$ and $\tilde{d}_{\tilde{S}}(\tilde{s}_e, \tilde{t}_t) = 2$.

- For each $e \in E_{S^c}$, at time step τ, terminal $\tilde{t}_e$ can use the following information for decoding
$$([\tilde{w}_e]_{\tau-q-4}^{\tau-4}, [\tilde{w}_{E_G(\mathrm{In}(e))}]_{\tau-q-4}^{\tau-4}, [\tilde{w}_{S_G(\mathrm{In}(e))}]_{\tau-q-4}^{\tau-4}, [\tilde{X}_B]_{\tau-q-2}^{\tau-2}).$$
It first extracts $([\tilde{X}_{E_G(\mathrm{In}(e))}]_{\tau-q-3}^{\tau-4})$ from $[\tilde{X}_B]_{\tau-q-2}^{\tau-3}$ according to (10.2) and then computes $\tilde{X}_{e,\tau-3}$ according to (10.3). Finally, it extracts $\tilde{w}_{e,\tau-3}$ from $\tilde{X}_{B,\tau-2}$ using (10.2).

- For each demand pair (s,t) with delay $d_T(t)$ in $\mathcal{I}$, terminal $\tilde{t}_t$ has information
$$([\tilde{w}_s]_{\tau-d_T(t)-q-4}^{\tau-d_T(t)-4}, [\tilde{w}_{E_G(t)}]_{\tau-q-4}^{\tau-4}, [\tilde{w}_{S_G(t)}]_{\tau-q-4}^{\tau-4}, [\tilde{X}_B]_{\tau-q-2}^{\tau-2}).$$
It first extracts $[\tilde{X}_{E_G(t)}]_{\tau-q-3}^{\tau-4}$ from $[\tilde{X}_B]_{\tau-q-2}^{\tau-3}$ according to (10.2). The terminal then obtains $\tilde{w}_{s,\tau-d_T(t)-3}$ by applying the decoding function g_t on the values of $([\tilde{w}_s]_{\tau-d_T(t)-q-3}^{\tau-d_T(t)-4}, [\tilde{w}_{S_G(t)}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_{\mathrm{In}(t)}]_{\tau-q-3}^{\tau-4})$. This must be true by the feasibility assumption of the network code and (10.1).

Note that by the feasibility assumption of the streaming network code, the initialization for $\tilde{\mathcal{K}}$ will ensure the first q time steps to operate properly.

Direction 2: If $\tilde{\mathcal{I}}$ is $(\tilde{\mathbf{R}}, \tilde{d}_{\tilde{S}}, \tilde{d}_{\tilde{T}})$–feasible then $\mathcal{I}$ is $(\mathbf{R}, d_S, d_T)$-streaming feasible. Consider an $(\tilde{\mathbf{R}}, \tilde{d}_{\tilde{S}}, \tilde{d}_{\tilde{T}})$–feasible code $(\tilde{\mathcal{F}}, \tilde{\mathcal{G}}, \tilde{\mathcal{K}})$ for $\tilde{\mathcal{I}}$, let the memory parameter of this code be q.

We want to show that there exists source realizations such that the broadcast link always sends some constant value. To do this we first show that the broadcast variable $\tilde{X}_{B,\tau}$ is "tight" in a sense that there is no redundancy in the code. We start by expanding $I(\tilde{\mathbf{W}}_{E_{S^c},\tau-3}; \tilde{X}_{B,\tau-2} | [\tilde{\mathbf{W}}_S]_{\tau-q-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3})$ in two different ways using the chain rule:

$$\begin{aligned}
&I(\tilde{\mathbf{W}}_{E_{S^c},\tau-3}; \tilde{X}_{B,\tau-2} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) \\
&\quad = H(\tilde{X}_{B,\tau-2} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) \\
&\quad - H(\tilde{X}_{B,\tau-2} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-3}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) && (10.4) \\
&\quad = H(\tilde{\mathbf{W}}_{E_{S^c},\tau-3} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) \\
&\quad - H(\tilde{\mathbf{W}}_{E_{S^c},\tau-3} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-2}). && (10.5)
\end{aligned}$$

The complete set of side information variables available at all the terminals at time step τ equals $([\tilde{\mathbf{W}}_S, \tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-2})$, by the decoding conditions that these node have to decode $\tilde{\mathbf{w}}_{E_{S^c},\tau-3}$ at time step τ, we have

$$H(\tilde{\mathbf{W}}_{E_{S^c},\tau-3} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-2}) = 0.$$

Next, since $\tilde{\mathbf{W}}_{E_{S^c},\tau-3}$ is independent of $([\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3})$, we have

$$H(\tilde{\mathbf{W}}_{E_{S^c},\tau-3} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) = \tilde{c}_B.$$

Therefore by (10.4), we have

$$\begin{aligned} &H(\tilde{X}_{B,\tau-2} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) \\ &\quad - H(\tilde{X}_{B,\tau-2} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-3}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) = \tilde{c}_B. \end{aligned} \tag{10.6}$$

Since $H(\tilde{X}_{B,\tau-2} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) \leq \tilde{c}_B$, we have

$$H(\tilde{X}_{B,\tau-2} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-3}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) = 0.$$

Namely, the broadcast information $\tilde{X}_{B,\tau-2}$ is a function of

$$([\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-3}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}).$$

From (10.6), we also have

$$H(\tilde{X}_{B,\tau-2} | [\tilde{\mathbf{W}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{W}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{X}_B]_{\tau-q-1}^{\tau-3}) = \tilde{c}_B. \tag{10.7}$$

The above equations have the following implications: for any fixed realization $([\tilde{\mathbf{w}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{w}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{x}_B]_{\tau-q-1}^{\tau-3})$, if we evaluate $\tilde{X}_{B,\tau-1}$ using

$$([\tilde{\mathbf{w}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{w}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{x}_B]_{\tau-q-1}^{\tau-3})$$

and $2^{\tilde{c}_B}$ possible values for $\tilde{\mathbf{w}}_{E_{S^c},\tau-3}$ as inputs, the values for $\tilde{X}_{B,\tau-2}$ cycle through all $2^{\tilde{c}_B}$ possible values in $\mathbb{F}_2^{\tilde{c}_B}$ and eventually hit $\mathbf{0}$. This must be true or we would have $H(\tilde{X}_{B,\tau-2} | ([\tilde{\mathbf{w}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{w}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{x}_B]_{\tau-q-1}^{\tau-3})) < \tilde{c}_B$, which is a contradiction of (10.7). This implies that for any realization

$$([\tilde{\mathbf{w}}_S]_{\tau-q-\tilde{d}_{\max}-3}^{\tau-4}, [\tilde{\mathbf{w}}_{E_{S^c}}]_{\tau-q-3}^{\tau-4}, [\tilde{x}_B]_{\tau-q-1}^{\tau-3}),$$

there exists a corresponding edge message realization for $\tilde{\mathbf{w}}_{E_{S^c},\tau-3}$ such that $\tilde{X}_{B,\tau-2}$ evaluates to $\mathbf{0}$ given these inputs.

Let $\tilde{\mathcal{K}} = \{[\tilde{x}_B]^0_{-\tilde{d}_{\max}-q}, [\tilde{\mathbf{w}}_S]^0_{-\tilde{d}_{\max}-q}, [\tilde{\mathbf{w}}_E]^0_{-\tilde{d}_{\max}-q}\}$ be the initializing constants for $\tilde{\mathcal{I}}$. By induction on the time indices, for any source message realization $(\tilde{\mathbf{w}}_{S,\tau'})^{\tau' \in \mathbb{N}}$, there exists a corresponding edge message realization $(\tilde{\mathbf{w}}_{E_{S^c},\tau'})^{\tau' \in \mathbb{N}}$ such that if these were the realizations for the index code and $\tilde{\mathcal{K}}$ were the initializing constants, the broadcast information $\tilde{X}_{B,\tau}$ would be $\mathbf{0}$ for all $\tau \in \mathbb{N}$. By the feasibility assumption of the index code, all terminals must decode correctly when the message realizations are $(\tilde{\mathbf{w}}_{S,\tau'}, \tilde{\mathbf{w}}_{E_{S^c},\tau'})^{\tau' \in \mathbb{N}}$. Therefore, the following is true when each terminal decodes at time step $\tau > q$:

- For any $e \in E_{S^c}$,

$$\tilde{w}_{e,\tau-3} = \tilde{g}_{\tilde{t}_e}(\mathbf{0}[\tilde{\mathbf{w}}_{E_G(\mathrm{In}(e))}]^{\tau-4}_{\tau-q-3}, [\tilde{\mathbf{w}}_{S_G(\mathrm{In}(e))}]^{\tau-4}_{\tau-q-3}, [\tilde{w}_e]^{\tau-4}_{\tau-q-3}).$$

- For any demand pair (s,t) and delay $d_T(t)$ in $\mathcal{I}$,

$$\tilde{w}_{s,\tau-d_t-3} = \tilde{g}_{\tilde{t}_t}(\mathbf{0}, [\tilde{\mathbf{w}}_{E_G(t)}]^{\tau-4}_{\tau-q-3}, [\tilde{\mathbf{w}}_{S_G(t)}]^{\tau-4}_{\tau-q-3}, [\tilde{w}_s]^{\tau-d_T(t)-4}_{\tau-d_T(t)-q-3}).$$

For a suitable choice of initializing constants, a network code for $\mathcal{I}$ can therefore be obtained as follows:

- For any $e \in E_{S^c}$,

$$\begin{aligned} &f_e([\mathbf{X}_{E_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q}, [\mathbf{W}_{S_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q}, [X_e]^{\tau-1}_{\tau-q}) \\ &\qquad = \tilde{g}_{\tilde{t}_e}(\mathbf{0}, [\mathbf{X}_{E_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q}, [\mathbf{W}_{S_G(\mathrm{In}(e))}]^{\tau-1}_{\tau-q}, [X_e]^{\tau-1}_{\tau-q}). \end{aligned}$$

- For any demand pair (s,t) and delay $d_T(t)$ in $\mathcal{I}$,

$$\begin{aligned} &g_t([\mathbf{X}_{E_G(t)}]^{\tau-1}_{\tau-q}, [\mathbf{W}_{S_G(t)}]^{\tau-1}_{\tau-q}, [W_s]^{\tau-d_T(t)-1}_{\tau-d_T(t)-q}) \\ &\qquad = \tilde{g}_{\tilde{t}_t}(\mathbf{0}, [\mathbf{X}_{E_G(t)}]^{\tau-1}_{\tau-q}, [\mathbf{W}_{S_G(t)}]^{\tau-1}_{\tau-q}, [W_s]^{\tau-d_T(t)-1}_{\tau-d_T(t)-q}). \end{aligned}$$

Finally, we select the initializing constants for $\mathcal{I}$ from suitable realizations of $(\tilde{\mathbf{w}}_{S,\tau}, \tilde{\mathbf{w}}_{E_{S^c},\tau})$ for $\tilde{\mathcal{I}}$. We need realizations $(\tilde{\mathbf{w}}_{S,\tau}, \tilde{\mathbf{w}}_{E_{S^c},\tau})$ that yield $\tilde{x}_{B,\tau} = \mathbf{0}$ for a long enough time period. Start with the constants

$$\tilde{\mathcal{K}} = \{[\tilde{x}_B]^0_{-\tilde{d}_{\max}-q}, [\tilde{\mathbf{w}}_S]^0_{-\tilde{d}_{\max}-q}, [\tilde{\mathbf{w}}_E]^0_{-\tilde{d}_{\max}-q}\}$$

for $\tilde{\mathcal{I}}$, let $[\tilde{\mathbf{w}}_S]^{d_{\max}+2q+1}_1$ be any source realizations and $[\tilde{x}_B]^{d_{\max}+2q+1}_1 = \mathbf{0}$. Let $[\tilde{\mathbf{w}}_E]^{d_{\max}+2q+1}_1$ be the edge message realizations computed as follow:

$$\tilde{w}_{e,\tau-3} = \tilde{g}_{\tilde{t}_e}(\mathbf{0}[\tilde{\mathbf{w}}_{E_G(\mathrm{In}(e))}]^{\tau-4}_{\tau-q-3}, [\tilde{\mathbf{w}}_{S_G(\mathrm{In}(e))}]^{\tau-4}_{\tau-q-3}, [\tilde{w}_e]^{\tau-4}_{\tau-q-3}).$$

The reason for taking variables for $d_{\max}+2q+1$ time steps is that $[\tilde{x}_B]^0_{-\tilde{d}_{\max}-q}$ might not be all $\mathbf{0}$. A suitable assignment for $\mathcal{K}$ is therefore

$$\begin{aligned}[\mathbf{w}_S]^0_{-\tilde{d}_{\max}-q} &= [\tilde{\mathbf{w}}_S]^{d_{\max}+2q+1}_{q+1},\\ [\mathbf{x}_{E_{S^c}}]^0_{-\tilde{d}_{\max}-q} &= [\tilde{\mathbf{w}}_{E_{S^c}}]^{d_{\max}+2q+1}_{q+1}.\end{aligned}$$

□

Chapter 11

SUMMARY

In this work, we explore reductions in various settings.

In the study of reduction with respect to network demands, we show all multiple multicast demands reduce to multiple unicast demands for both linear and general capacity regions. When restricted to linear codes, multiple unicast network coding further reduces to 2-unicast network coding. We also prove that the code reduction of [14] (from multiple multicast to 2-unicast) extends to a capacity reduction if and only if the AERS[1] holds. These results help identify the demand type that is representative in each scenario.

In the study of capacity reduction with respect to network topology, we show that acyclic network coding reduces to index coding under the restriction of linear codes. For general codes, we show that the code reduction in [6], [7] extends to a capacity reduction if and only if the AERS holds. Further, we show that under a streaming model, the code reduction from [6] extends to a code reduction from general network coding to index coding. This result shows that under the streaming model, there is no fundamental difference between acyclic networks and networks containing cycles. A summary of the reduction results is given in Figure 11.1.

We also study characterizations of the network coding capacity region. We show that the Yeung outer bound yields an alternative characterization for the network coding capacity region if and only if the AERS holds. If the Yeung outer bound is tight, it not only gives a potentially simpler characterization of the capacity region but also resolves a series of problems connected to the AERS. Following a similar approach to the one used to prove the Yeung outer bound [15], we derive a characterization of zero-error network coding capacity region using a dense subset of the entropic region. We also show that under the assumption of linear encoders, there is no difference between epsilon-error and zero-error network coding capacity regions.

For future work, it would be useful to understand the implications of the

[1] Asymptotic edge removal statement

streaming model to the network coding capacity region as well as to extend results proven for acyclic networks to networks containing cycles. We have linked several capacity reductions to the edge removal statement, identifying addition problems that are connected to the edge removal statement will help us gain insights to this group of connected problems.

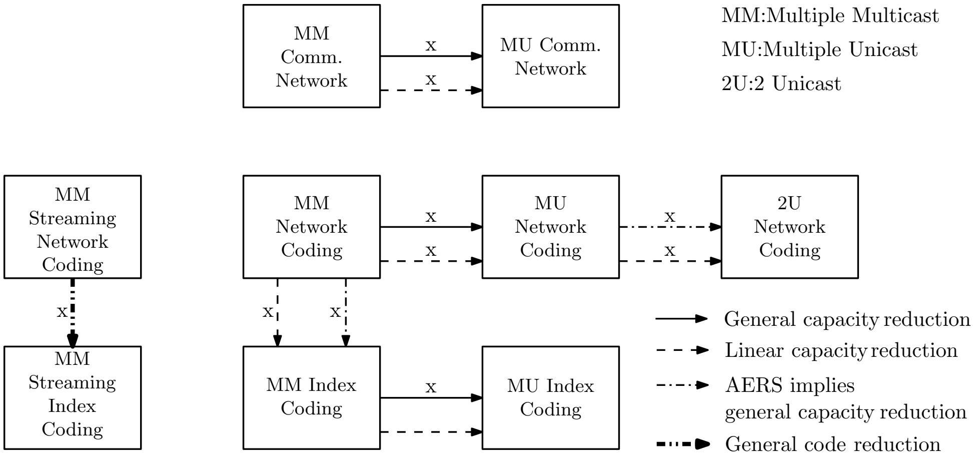


Figure 11.1: A summary of reduction results with respect to demand and network types. Solid arrows represent reductions in the general capacity region. Dashed arrows represent reductions in the linear capacity region. Dash-dotted arrows represent general capacity reductions that hold when the AERS holds. Dash-dot-dotted arrows represent general code reductions. Our contributions are marked with an “x”.

BIBLIOGRAPHY

[1] C. E. Shannon, “A mathematical theory of communication,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[2] R. Ahlswede, “Multi-way communication channels,” in *Second International Symposium on Information Theory: Tsahkadsor, Armenia, USSR, Sept. 2-8, 1971*, 1973.

[3] H. H.-J. Liao, “Multiple access channels.,” DTIC Document, Tech. Rep., 1972.

[4] W. Gu, M. Effros, and M. Bakshi, “A continuity theory for lossless source coding over networks,” in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, IEEE, 2008, pp. 1527–1534.

[5] 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.

[6] S. El Rouayheb, A. Sprintson, and C. Georghiades, “On the index coding problem and its relation to network coding and matroid theory,” *IEEE Transactions on Information Theory*, vol. 56, no. 7, pp. 3187–3195, 2010.

[7] M. Effros, S. El Rouayheb, and M. Langberg, “An equivalence between network coding and index coding,” *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2478–2487, 2015.

[8] S. Jalali, M. Effros, and T. Ho, “On the impact of a single edge on the network coding capacity,” in *Information Theory and Applications Workshop, 2011*, IEEE, 2011, pp. 1–5.

[9] N. J. Harvey, R. Kleinberg, C. Nair, and Y. Wu, “A “chicken & egg” network coding problem,” in *Information Theory, 2007 IEEE International Symposium on*, IEEE, 2007, pp. 131–135.

[10] C.-C. Wang and M. Chen, “Sending perishable information: Coding improves delay-constrained throughput even for single unicast,” *IEEE Transactions on Information Theory*, vol. 63, no. 1, pp. 252–279, 2017.

[11] C. Chekuri, S. Kamath, S. Kannan, and P. Viswanath, “Delay-constrained unicast and the triangle-cast problem,” in *Information Theory, 2015 IEEE International Symposium on*, IEEE, 2015, pp. 804–808.

[12] R. Dougherty and K. Zeger, “Nonreversibility and equivalent constructions of multiple-unicast networks,” *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5067–5077, 2006.

[13] H. Maleki, V. R. Cadambe, and S. A. Jafar, “Index coding-an interference alignment perspective,” *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5402–5432, 2014.

[14] S. Kamath, N. David, and C.-C. Wang, “Two-unicast is hard,” in *Information Theory, 2014 IEEE International Symposium on*, IEEE, 2014, pp. 2147–2151.

[15] R. W. Yeung, *A first course in information theory*. Springer Science & Business Media, 2012.

[16] X. Yan, R. W. Yeung, and Z. Zhang, “An implicit characterization of the achievable rate region for acyclic multisource multisink network coding,” *IEEE Transactions on Information Theory*, vol. 58, no. 9, pp. 5625–5639, 2012.

[17] T. H. Chan and A. Grant, “Network coding capacity regions via entropy functions,” *IEEE Transactions on Information Theory*, vol. 60, no. 9, pp. 5347–5374, 2014.

[18] S. Thakor, A. Grant, and T. Chan, “Network coding capacity: A functional dependence bound,” in *Information Theory, 2009 IEEE International Symposium on*, IEEE, 2009, pp. 263–267.

[19] M. Langberg and M. Effros, “Network coding: Is zero error always possible?” In *Communication, Control, and Computing, 2011 49th Annual Allerton Conference on*, IEEE, 2011, pp. 1478–1485.

[20] L. Song, R. W. Yeung, and N. Cai, “Zero-error network coding for acyclic networks,” *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3129–3139, 2003.

[21] W. Huang, M. Langberg, and J. Kliewer, “Connecting multiple-unicast and network error correction: Reduction and unachievability,” in *Information Theory, 2015 IEEE International Symposium on*, IEEE, 2015, pp. 361–365.

[22] W. Huang, T. Ho, M. Langberg, and J. Kliewer, “Single-source/sink network error correction is as hard as multiple-unicast,” in *Communication, Control, and Computing, 2014 52nd Annual Allerton Conference on*, IEEE, 2014, pp. 423–430.

[23] ——, “On secure network coding with uniform wiretap sets,” in *Network Coding, 2013 International Symposium on*, IEEE, 2013, pp. 1–6.

[24] R. Koetter, M. Effros, and M. Médard, “A theory of network equivalence-part i: Point-to-point channels,” *IEEE Transactions on Information Theory*, vol. 57, no. 2, pp. 972–995, 2011.

[25] M. Langberg and M. Effros, “Source coding for dependent sources,” in *Information Theory Workshop, 2012 IEEE*, IEEE, 2012, pp. 70–74.

[26] T. Chan and A. Grant, "On capacity regions of non-multicast networks," in *Information Theory, 2010 IEEE International Symposium on*, IEEE, 2010, pp. 2378–2382.

[27] M. Langberg and M. Effros, "On the capacity advantage of a single bit," *ArXiv preprint arXiv:1607.07024*, 2016.

[28] O. Kosut and J. Kliewer, "On the relationship between edge removal and strong converses," in *Information Theory, 2016 IEEE International Symposium on*, IEEE, 2016, pp. 1779–1783.

[29] T. Ho, M. Effros, and S. Jalali, "On equivalence between network topologies," in *Communication, Control, and Computing, 2010 48th Annual Allerton Conference on*, IEEE, 2010, pp. 391–398.

[30] P. Noorzad, M. Effros, M. Langberg, and T. Ho, "On the power of cooperation: Can a little help a lot?" In *Information Theory, 2014 IEEE International Symposium on*, IEEE, 2014, pp. 3132–3136.

[31] M. F. Wong, M. Langberg, and M. Effros, "On a capacity equivalence between multiple multicast and multiple unicast," in *Communication, Control, and Computing, 2013 51st Annual Allerton Conference on*, IEEE, 2013, pp. 1537–1544. DOI: `10.1109/Allerton.2013.6736710`,

[32] M. F. Wong, M. Langberg, and M. Effros, "Linear capacity equivalence between multiple multicast and multiple unicast," in *Information Theory, 2014 IEEE International Symposium on*, IEEE, 2014, pp. 2152–2156. DOI: `10.1109/ISIT.2014.6875214`,

[33] E. Van der Meulen, "A survey of multi-way channels in information theory: 1961-1976," *IEEE Transactions on Information Theory*, vol. 23, no. 1, pp. 1–37, 1977.

[34] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 572–584, 1979.

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

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

[37] 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, 2006.

[38] S. Jalali and M. Effros, "Separation of source-network coding and channel coding in wireline networks," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1524–1538, 2015.

[39] P. Elias, "Coding for noisy channels," in *Proceedings of the Institute of Radio Engineers*, vol. 43, 1955, pp. 356–356.

[40] M. F. Wong, M. Effros, and M. Langberg, "On an equivalence of the reduction of k-unicast to 2-unicast capacity and the edge removal property," in *Information Theory, 2015 IEEE International Symposium on*, IEEE, 2015, pp. 371–375. DOI: `10.1109/ISIT.2015.7282479`,

[41] M. Bakshi and M. Effros, "On network coding capacity under on-off scheduling," in *Information Theory, 2012 IEEE International Symposium on*, IEEE, 2012, pp. 1667–1671.

[42] T. Ho, "Networking from a network coding perspective," PhD thesis, Doctoral dissertation, Massachusetts Institute of Technology, 2004.

[43] M. Langberg and M. Effros, "The edge removal problem as a canonical problem in network coding," *Submitted to IEEE Transactions on Information Theory*,

[44] 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, 2005.

[45] 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, 2007.

[46] R. W. Yeung, "A framework for linear information inequalities," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1924–1934, 1997.

[47] T. C. Hu, "Multi-commodity network flows," *Operations research*, vol. 11, no. 3, pp. 344–360, 1963.

[48] I. Csiszár, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Transactions on Information Theory*, vol. 28, no. 4, pp. 585–592, 1982.

[49] M. F. Wong, M. Langberg, and M. Effros, "On a capacity equivalence between network and index coding and the edge removal problem," in *Information Theory Proceedings, 2013 IEEE International Symposium on*, IEEE, 2013, pp. 972–976. DOI: `10.1109/ISIT.2013.6620371`,

[50] M. F. Wong, M. Effros, and M. Langberg, "On tightness of an entropic region outer bound for network coding and the edge removal property," in *Information Theory, 2016 IEEE International Symposium on*, IEEE, 2016, pp. 1769–1773. DOI: `10.1109/ISIT.2016.7541603`,

[51] Z. Zhang and R. W. Yeung, "A non-shannon-type conditional inequality of information quantities," *IEEE Transactions on Information Theory*, vol. 43, no. 6, pp. 1982–1986, 1997.

[52] W. Kim, M. Langberg, and M. Effros, “A characterization of the capacity region for network coding with dependent sources,” in *Information Theory, 2016 IEEE International Symposium on*, IEEE, 2016, pp. 1764–1768.

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

[54] M. F. Wong, M. Effros, and M. Langberg, “A code equivalence between streaming network coding and streaming index coding,” in *To appear in Information Theory, 2017 IEEE International Symposium on*, IEEE, 2017,